

TODO O CONTEÚDO DA NOVA VERSÃO DO CCNA EM
UM LIVRO ESCRITO DE FORMA CLARA E DIRETA.

CCNA - 200 301

O guia definitivo - Por Luiz Silvério





GUIA DEFINITIVO CCNA 200-301

LUIZ SILVÉRIO

(1^a EDIÇÃO)

Conteúdo

Apresentação	16
Estrutura do livro	16
Estrutura da prova	16
Prova.....	16
1.0 - Fundamentos de redes de computadores	17
Objetivos de uma rede de computadores	17
Modelo OSI	18
Modelo TCP/IP	21
Exercícios	23
1.1 Explain the role and function of network components	25
1.1.a Routers (roteadores).....	25
1.1.b L2 and L3 switches	26
Endereço MAC.....	27
1.1.c Next-generation firewalls and IPS.....	28
Tipos de firewall.....	28
Next-generation firewalls– NGFW	29
IPS -Intrusion Prevent System:	30
1.1.d Access Points	31
1.1.e Controllers (Cisco DNA Center and WLC).....	31
Controllers	31
Cisco DNA Center.....	32
WLC (Wireless Lan Controller)	33
1.1.f Endpoints	34
1.1.g Servers	34
1.2 - Describe characteristics of network topology architectures	34
Introdução a arquitetura de redes.....	35
Camada de Acesso.....	39
Camada de Distribuição	39
Camada Central	39
1.2b - 3 tier - Arquitetura em 3 camadas	43
1.2a - 2 tier - Arquitetura em 2 camadas	44
1.2c Spine-leaf	45
1.2d – WAN	46
1.2e - Small office/home office (SOHO)	47
1.2.f On-premises and cloud.....	47
1.3 Compare physical interface and cabling types	48

Introdução a Ethernet	48
Camada física	49
1.3.a Single-modefiber, multimodefiber, copper	49
1.3.a Copper	49
Cooper - Cabos Ethernet – CAT 5e.....	49
Cooper - Cabos Ethernet – CAT 6	49
Cooper - Cabos Ethernet – CAT 6A	50
Cooper - Cabos Ethernet – CAT 7	50
Cooper - Cabos Ethernet – Conector RJ45.....	51
Cooper - Cabos Ethernet – Cabo direto (Straight Through).....	51
Copper- Cabos Ethernet - Cabo cruzado (Crossover)	52
Copper- Cabos Ethernet - Cabo cruzado x Cabo Direto.....	52
Copper- Cabos Ethernet – Seleção do cabo correto	53
Cabo 1000BASE-T (Gigabit Cabling)	53
Single-mode fiber, multi mode fiber, copper.....	54
Multimodefiber ou MM	56
Single-modefiber ou SM	56
1.3.b Connections (Ethernet shared media and point-to-point).....	57
Shared media and point-to-point	57
CSMA/CD – Duplex	58
Ethernet - Data Link Layer.....	60
Ethernet - MAC Address.....	60
Ethernet -Type Field.....	61
Ethernet -Error Detection	61
1.3.c Concepts of PoE	62
Midspan e Endspan	62
Endspan:	62
Midspan:.....	63
Quando usar Midspan ou Endspan:.....	64
Padrões, classes e tipos PoE	64
Dispositivos de alimentação PoE	65
Vantagens da utilização do PoE	65
Desvantagens da utilização do PoE.....	65
Resumo:.....	66
Introdução ao Cisco IOS CLI (Interface de linha de comando)	66
Acesso ao Cisco IOS	67
Conectando a porta console.....	67
Emulador de terminal (terminal emulator)	68

Primeiro boot (inicialização)	69
Modo de usuário e privilegiado (User and Enable mode)	73
Apagando as configurações do dispositivo	73
Comandos Show.....	74
Show version:.....	74
show running-config	76
show mac address-table dynamic	79
Configurações iniciais.....	79
Salvando às configurações	81
Ferramentas de ajuda:	82
Ponto de interrogação.....	82
Abreviações.....	83
Tratamento de comandos errados e incompletos.....	84
Atalhos de teclado:	84
Comando DO.....	85
Modificadores de saída:.....	85
Modo de usuário e modo privilegiado seguros.....	87
Modo de usuário seguro.....	87
Autenticação simples.....	87
Usuário e senha	88
Modo de Segurança no Enable	88
Criptografia de senhas	89
Comando Secret	90
Servidores de autenticação externa	91
1.4 Identify interface and cable issues (collisions, errors, mismatch duplex, and/or speed)	92
Duplex/velocidade.....	92
Violações de segurança nas interfaces	96
Incompatibilidade de protocolos	99
1.5 Compare TCP to UDP	100
Protocolo UDP	102
Protocolo TCP	102
Cabeçalho TCP.....	106
TCP Window Size Scaling	108
1.7 Configure and verify IPv4 addressing and subnetting	110
Subneting.....	115
Design de sub-redes	116
1 ^a Opção: Sub-redes com tamanho único:.....	117
2 ^a opção: Sub-redes de tamanho variado.....	117

Configuração de endereço IPV4 em roteadores Cisco	118
Noções básicas de números binários	120
Decimal para binário:	120
Binário para Decimal	120
Endereço de Network e Broadcast	121
Sub-redes em binário.....	121
Sub-rede Classe C	122
Sub-rede Classe B	124
Sub-rede Classe A	126
Classless InterDomain Routing (CIDR).....	129
Variable Length Subnet Mask (VLSM)	130
1.7 Describe the need for private IPv4 addressing	132
1.8 Configure and verify IPv6 addressing and prefix.....	133
Transformando números decimais e binários em hexadecimal	134
Abreviando endereços IPv6	137
Prefixo IPv6.....	137
Configurando endereços IPv6 em dispositivos Cisco	140
1.9 Compare IPv6 address types	141
1.9.a Global unicast.....	141
1.9.b Unique local	141
1.9.c Link local.....	141
1.9.d Anycast.....	141
1.9.e Multicast.....	142
1.9.f Modified EUI 64	142
1.10 Verify IP parameters for Client OS (Windows, Mac OS, Linux).....	144
Windows	144
Mac OS.....	144
Linux	145
1.11 Describe wireless principles	145
1.11.a Nonoverlapping Wi-Fi channels.....	145
1.11.b SSID	147
IBSS	147
Modo de infraestrutura	148
Basic Service Set (BSS)	148
Distribution System (DS)	149
Extended Service Set (ESS)	150
Mesh Basic Service Set (MBSS).....	150
AP Modes	151

Repetidor	151
Workgroup Bridge.....	152
Resumo:.....	152
1.11.c RF	152
1.11.d Encryption.....	153
Introdução a segurança de rede sem fio	153
Authentication	153
Encryption	155
Integrity	156
Resumo.....	157
Algoritmos de criptografia	157
TKIP	157
CCMP.....	157
GCMP	158
Resumo:.....	158
1.12 Explain virtualization fundamentals (virtual machines).....	159
Virtual Machines	159
Containers	160
Introdução a Cloud Computer	161
Servidores Bare Metal	161
Virtualização de servidores	162
Rede virtual	163
Redes de Data Center	164
TOR (Top of Rack)	164
EoR (End of Row).....	164
Computação em nuvem.....	165
Modelos de serviço.....	165
IaaS (Infraestrutura como serviço)	165
PaaS (plataforma como serviço).....	166
SaaS (software como serviço)	166
Nuvem Pública	166
Nuvem Privada.....	166
Nuvem hibrida.....	167
1.13 Describe switching concepts.....	167
1.13.a MAC learning and aging.....	168
Comandos de verificação	169
Aging.....	171
1.13.b Frame switching.....	173

Método store-and-forward.....	175
Método cut-through.....	175
Método fragment-free	176
1.13.c Frame flooding	176
1.13.d MAC address table	178
Exercícios.....	179
2.0 Network Access	182
2.1 Configure and verify VLANs (normal range) spanning multiple switches	182
Configuração de Vlans em Switches Cisco.....	183
2.1.a Access ports (data and voice).....	186
Vlan de dados.....	186
Vlan de Voz.....	187
Configuração	188
Verificação	188
2.1.b Default VLAN.....	190
2.2 Configure and verify interswitch connectivity	190
2.2.a Trunk ports	190
DTP – Dynamic Trunking Protocol	199
2.2.b 802.1Q.....	199
2.2.c Native VLAN	200
VTP (VLAN Trunking Protocol).....	203
2.3 Configure and verify Layer 2 discovery protocols (Cisco Discovery Protocol and LLDP).....	214
CDP - Cisco Discovery Protocol.....	214
LLDP - Link Layer Discovery Protocol	216
2.4 Configure and verify (Layer 2/Layer 3) EtherChannel (LACP)	218
Configuração do LACP	220
2.5 Describe the need for and basic operations of Rapid PVST+ Spanning Tree Protocol and identify basic operations	224
Introdução ao Spanning Tree.....	224
2.5.a Root port, root bridge (primary/secondary), and other port names	225
Configuração do Spanning-Tree em switches Cisco	228
Per VLAN Spanning Tree (PVST).....	233
Spanning-tree – Configuração do Root Bridge	234
Root Parameter.....	239
Priority Parameter.....	240
Calculo de custo do Spanning Tree	242
Rapid Spanning-Tree (RSTP)	245
Rapid Spanning Tree - Configuração	253

2.5.b Port states (forwarding/blocking).....	264
2.5.c PortFast benefits	267
PortFast desabilitado	267
Portfast habilitado	268
2.6 Compare Cisco Wireless Architectures and AP modes.....	269
Autonomous AP Architecture	269
Split-MAC Architecture	271
CAPWAP	272
Cloud-Based AP Architecture	274
Resumo.....	274
AP Modes.....	275
1. Local.....	275
2. Monitor.....	275
3. FlexConnect.....	275
4. Sniffer.....	276
5. Rogue Detector.....	276
6. Bridge/Mesh	276
7. Flex plus bridge	276
8. SE-Connect.....	276
2.7 Describe physical infrastructure connections of WLAN components (AP, WLC, access/trunk ports, and LAG)... 	276
Switch.....	277
Interfaces	277
WLC	278
LAG.....	280
Interface de gerenciamento	280
2.8 Describe AP and WLC management access connections (Telnet, SSH, HTTP, HTTPS, console, and TACACS+/RADIUS).....	283
2.9 Configure the components of a wireless LAN access for client connectivity using GUI only such as WLAN creation, security settings, QoS profiles, and advanced WLAN settings	286
Criação da Wlan	288
Secutiry Settings.....	288
QoS.....	289
Advanced WLAN settings.....	289
Exercícios	291
3. IP Connectivity	293
Introdução	293
Configuração básica de roteadores Cisco	296
Integrated Services Routers.....	296

Configuração	297
Primeiro Boot	298
Apagando a configuração inicial	300
Interfaces	301
3.1 Interpret the components of routing table.....	303
IP Routing Process	304
3.1.a Routing protocol code.....	309
3.1.b Prefix.....	311
3.1.c Network mask	312
3.1.d Next hop	312
3.1.e Administrative distance	313
3.1.f Metric	315
3.1.g Gateway of last resort.....	315
3.2 Determine how a router makes a forwarding decision by default.....	315
3.2 c Routing Protocol Metric	316
Métrica Internal gateway protocol (IGP).....	317
Métrica Exterior Gateway Protocols	317
Exemplo de melhor seleção de rota usando métricas dos protocolos de roteamento	317
3.2.b Administrative distance	317
Redes conectadas diretamente e rotas estáticas	318
Distância administrativa dos internal gateway protocol (IGP).....	318
Distância administrativa do Exterior Gateway Protocol (EGP)	318
3.2.a Longest match	319
3.3 Configure and verify IPv4 and IPv6 static routing.....	320
3.3.a Default route	320
3.3.b Network route	320
3.3.c Host route	321
3.3.d Floating static.....	321
3.4 Configure and verify single area OSPFv2	323
Introdução ao OSPF	323
Configuração OSPF.....	330
3.4.a Neighbor adjacencies.....	340
Pacotes OSPF e processo de descoberta de vizinho	340
Largura de banda de referência OSPF.....	345
3.4.b Point-to-point	347
3.4.c Broadcast (DR/BDR selection).....	348
3.4.d Router ID.....	350
OSPF Passive Interface	352

3.5 Describe the purpose of first hop redundancy protocol	354
HSRP (Hot Standby Routing Protocol).....	355
Processo de eleição para o Active Gateway	360
Preemption	361
Autenticação.....	362
Temporizadores HSRP	362
HSRP Versão 1 e 2.....	363
Exercícios.....	364
4.0 IP Services	366
 4.1 Configure and verify inside source NAT using static and pools.....	366
Introdução ao NAT.....	366
Configuração do NAT Estático	368
NAT Dinâmico.....	370
Port Address Translation.....	372
 4.2 Configure and verify NTP operating in a client and server mode.....	373
Configuração	374
Configuração do Roteador.....	374
Configuração do switch.....	376
NTP - Multicast e Broadcast	378
NTP Autenticação	379
 4.3 Explain the role of DHCP and DNS within the network	381
Introdução ao DHCP	381
Funcionamento do DHCP	381
DNS	382
 4.4 Explain the function of SNMP in network operations.....	383
Versões SNMP	384
Configuração SNMP	386
 4.5 Describe the use of syslog features including facilities and levels.....	386
Armazenamento de mensagens syslog	387
Armazenamento local.....	387
Servidor Syslog	388
Formato de Mensagem Syslog	389
Níveis de gravidade do syslog.....	389
 4.6 Configure and verify DHCP client and relay.....	391
Configurando Servidor DHCP em roteadores Cisco	392
DHCP Relay Agent	393
Configuração do DHCP relay agent.....	395
Clientes DHCP	397

Configuração	397
4.7 Explain the forwarding per-hop behavior (PHB) for QoS such as classification, marking, queuing, congestion, policing, shaping	398
Introdução	398
Características do tráfego de rede.....	399
Tipos de tráfego.....	400
Aplicação em lote.....	400
Aplicação Interativa	400
Aplicações de voz e vídeo	401
Ferramentas QoS	402
Classification and marking.....	402
Gestão de congestionamento	403
Round Robin	404
Fila de baixa latência - Low Latency Queuing.....	404
Policing and Shaping.....	404
Shaping.....	406
Congestion Avoidance	406
4.8 Configure network devices for remote access using SSH.....	408
Configuração	408
Servidor SSH.....	408
Cliente SSH	409
4.9 Describe the capabilities and function of TFTP/FTP in the network	410
FTP	411
TFTP	411
Exercícios.....	412
5.0 Security Fundamentals	414
5.1 Define key security concepts (threats, vulnerabilities, exploits, and mitigation techniques).....	414
Threats.....	414
Vulnerabilities	420
Exploits	420
Mitigation techniques.....	421
5.2 Describe security program elements (user awareness, training, and physical access control).....	421
Conscientização do usuário:.....	421
Training	422
Physical access control	423
5.3 Configure device access control using local passwords	423

5.4 Describe security password policies elements, such as management, complexity, and password alternatives (multifactor authentication, certificates, and biometrics)	424
5.5 Describe remote access and site-to-site VPNs.....	425
Remote Access	425
Site-to-site VPNs.....	426
Tipos VPN.....	426
Client-to-site VPN	427
Protocolos VPN.....	427
5.6 Configure and verify access control lists.....	428
Wildcard Mask	432
Standard access-list	433
Extended Access-List.....	436
5.7 Configure Layer 2 security features (DHCP snooping, dynamic ARP inspection, and port security)	442
DHCP snooping.....	442
DHCP Snooping configuração	443
Dynamic ARP inspection	445
Configuração	446
Port Security.....	451
5.8 Differentiate authentication, authorization, and accounting concepts.....	454
Authorization.....	454
Accounting	455
Protocolos AAA	455
RADIUS (Remote Authentication Dial-In User Service).....	455
TACACS +	456
5.9 Describe wireless security protocols (WPA, WPA2, and WPA3)	457
WPA	458
WPA2	458
WPA3	458
5.10 Configure WLAN using WPA2 PSK using the GUI.....	459
Exercícios	464
6.0 Automation and Programmability	466
6.1 Explain how automation impacts network management	466
Benefícios da automação de rede	466
6.2 Compare traditional networks with controller-based networking	467
6.3 Describe controller-based and software defined architectures (overlay, underlay, and fabric)	467
6.3.a Separation of control plane and data plane	469
Control plane	470
Data Plane	470

Management plane	470
6.3.b North-bound and south-bound APIs	471
SDN (Software Defined Networking)	471
Southbound Interface	472
Northbound Interface	473
6.4 Compare traditional campus device management with Cisco DNA Center enabled device management	474
Redes Tradicionais	474
SD-Access Fabric	474
Papel do DNA Center na rede	474
Recursos e funções	475
Automation.....	475
Assurance	475
SD-Access	476
Platform.....	476
6.5 Describe characteristics of REST-based APIs (CRUD, HTTP verbs, and data encoding)	477
Web Service	477
CRUD.....	479
HTTP Verbs	480
Protocolo HTTP	480
Métodos HTTP.....	480
6.6 Recognize the capabilities of configuration management mechanisms Puppet, Chef, and Ansible	481
Features (recursos)	481
Agent vs Agentless.....	482
Puppet.....	482
Chef	482
Ansible	482
6.7 Interpret JSON encoded data.....	483
Exercícios.....	485

Sobre o Autor:

Luiz Silvério é um profissional renomado no ramo da Tecnologia da Informação. Detentor de várias certificações das principais empresas do ramo (Cisco, Microsoft e Juniper), Luiz faz parte do seletº grupo de pessoas que compreendem que o conhecimento deve ser compartilhado e JAMAIS retido.

Criador e mantenedor da conta **CiscoBrasil** no Instagram, com aproximadamente 15 mil seguidores, onde oferece de maneira gratuita dicas e aulas sobre Cisco e carreira profissional.

Neste livro, Luiz Silvério se dedicou a escrever um guia completo sobre a nova certificação CCNA, de maneira prática, leve e objetiva para todos aqueles que desejam conquistar a certificação e ter sucesso em qualquer empresa.

A certificação CCNA é, sem sombra de dúvidas, uma das melhores ferramentas para obter um ótimo salário.

Apresentação

Primeiramente gostaria de agradecer a todos vocês que adquiriram esse livro por meios legais.

Esse é o primeiro livro em português sobre a nova versão do CCNA, a versão 200-301 (exceto os livros gigantescos e chatos da Cisco Book). Quem acompanha meu trabalho nas redes sociais, sabe que prezo por uma comunicação clara e direta, sem floreios, mas ao mesmo tempo mantendo a linguagem técnica quando está se faz necessária.

Esse livro é exatamente assim: Claro e direto, como se fosse uma conversa entre dois amigos, porque na verdade é isso que somos, dois amigos conversando sobre um assunto.

Tudo que você precisa para passar na certificação está aqui. Mas, o livro vai além disso, ele não é simplesmente voltado para a certificação, é voltado para que você possa aprimorar-se tecnicamente e ampliar as possibilidades na sua carreira profissional.

Estrutura do livro

O livro foi estruturado da mesma maneira que a Cisco estruturou o blueprint da prova, isso tem vantagens e desvantagens.

A principal vantagem é a facilidade de pesquisar determinado assunto, pois o índice está bem claro. Além disso, seguindo o blueprint, você terá a certeza que estudou todos os pontos cobrados para o exame.

A desvantagem é que pedagogicamente não dá para apresentar determinado assunto apenas quando ele aparece no blueprint, alguns assuntos precisam ser abordados antes, durante e depois da sua aparição ‘oficial’. Isso pode causar uma sensação que o livro é repetitivo, porém, sempre que foi necessário tratar do mesmo assunto procurei fazer de forma diferente.

Alguns assuntos não estão no blueprint, mas julguei necessário inclui-los no livro, afinal, são tópicos fundamentais para compreensão dos assuntos cobrados na prova e para sua evolução como profissional de redes.

Na área de tecnologia grande parte dos termos são em inglês, procurei traduzir os termos, mas não simplesmente ao pé da letra, na verdade, procurei ‘abrasileirar’ as traduções para que façam mais sentido dentro do contexto em que estão inseridos. Porém, depois da primeira vez que os termos aparecem e são traduzidos, eles aparecerão somente em inglês, pois essa é a linguagem que você fará a prova, além de ser a forma que nós, profissionais de redes, conversamos no dia a dia durante o trabalho.

Estrutura da prova

A prova foi dividida em 06 grandes blocos:

- 1.0 **Network fundamentals**, contendo aproximadamente 20% de todo o conteúdo;
- 2.0 **Network Access**, contendo aproximadamente 20% de todo conteúdo;
- 3.0 **IP connectivity**, contendo aproximadamente 25% de todo conteúdo;
- 4.0 **IP Services**, contendo aproximadamente 10% de todo conteúdo;
- 5.0 **Security Fundamentals**, contendo aproximadamente 15% de todo conteúdo;
- 6.0 **Automation and Programmability**, contendo aproximadamente 10% de todo conteúdo.

Note que essa divisão também foi adotada por esse livro.

Prova

A prova é composta por 120 questões em língua inglesa (não existindo versão em português) com o tempo máximo de 120 minutos, porém, como Inglês não é nossa língua nativa, a Cisco oferece um tempo extra de mais 30 minutos.

A prova pode ser agendada no site da Vue: www.vue.com

1.0 - Fundamentos de redes de computadores

Nessa primeira parte do livro vamos entender o funcionamento e o papel de cada componente de uma rede de computadores, aprenderemos também as arquiteturas e tecnologias utilizadas, tipos de cabos e conexões, protocolos de camada 02, endereçamento IP, Wireless e virtualização.

Toda essa parte comprehende a primeira parte da nova prova CCNA (200-301), cobrindo aproximadamente 20% da prova. É um tópico importantíssimo, pois os fundamentos que você utilizará para o resto da sua vida profissional estão aqui.

Mas você sabe o que é uma **rede de computadores**?

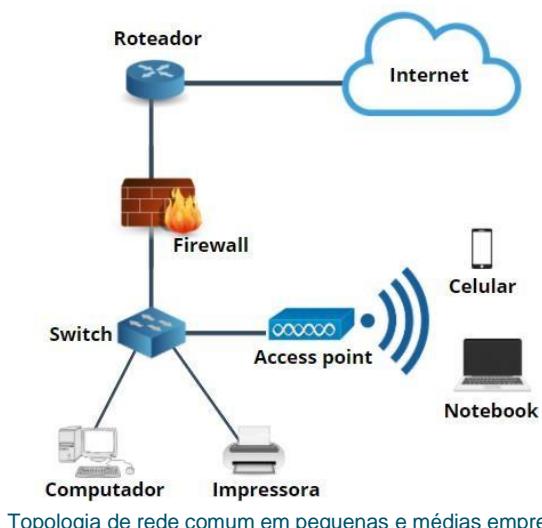
- Uma Rede de Computadores é formada por um conjunto de dispositivos capazes de trocar informações e compartilhar recursos, interligados por um sistema de comunicação. O sistema de comunicação vai se constituir de um arranjo topológico interligando os vários dispositivos através de enlaces físicos (meios de transmissão) e de um conjunto de regras com o fim de organizar a comunicação (protocolos). As redes foram feitas para compartilhamento: Compartilhamento de arquivos, processadores de textos e planilhas, impressoras, conexões com redes de computadores distantes, sistemas de correio eletrônico, etc. Estas são apenas algumas das funções de uma rede.

De maneira resumida, podemos dizer que rede de computador é uma coleção de dispositivos que podem se comunicar utilizando protocolos iguais.

Objetivos de uma rede de computadores

- **COMPARTILHAMENTO DE RECURSOS:** O compartilhamento de recursos permite que programas, dados, periféricos, área de armazenamento, entre outros, estejam disponíveis para qualquer um na rede, independentemente da localização física do recurso e do usuário, por exemplo: Podemos conectar uma impressora à rede e ela ser compartilhada por todo um setor.
- **AUMENTO DA CONFIABILIDADE:** Considerando-se que passa a existir redundância de recursos disponíveis, dependendo da forma como a rede é projetada e implementada, a tolerância a falhas é ampliada consideravelmente, o que amplia a confiança no sistema.
- **REDUÇÃO DE CUSTOS:** Como estamos compartilhando recursos, isso ajuda a diminuir o custo, por exemplo, como o já citado caso da impressora em que basta uma por setor e não uma para cada usuário.

Abaixo o típico diagrama da rede de uma empresa, compartilhando recursos como impressora e possibilitando acesso a Internet.



Topologia de rede comum em pequenas e médias empresas

Aqui temos algumas nomenclaturas importantes:

- **Endpoint Devices (Dispositivos finais)** – São os equipamentos que precisam acessar a rede, tipicamente os equipamentos que os usuários utilizam.
 - Computador
 - Impressora
 - Smartphones
 - Tablets
- **Network Devices (Dispositivos de rede)** - São os equipamentos responsáveis por transmitir dados entre os dispositivos finais.
 - Switches
 - Roteadores
 - Firewall
 - Access Point
- **Network Data (Dados que trafegam na rede)** – São as informações que são enviadas através da rede.
 - Email
 - Navegação na internet
 - Áudio, vídeo
 - Mensagens instantâneas

➤ **Network Protocols (Protocolos de redes)** - Segundo o dicionário: '*Protocolo é o conjunto de informações, decisões, normas e regras definidas a partir de um ato oficial, como audiência, conferência ou negociação, por exemplo*'. Com essa definição, fica mais fácil definir o que seriam os tais protocolos de redes: Eles são um conjunto de regras que os dispositivos utilizam para se comunicar em uma rede.

Abaixo, uma pequena lista com os principais protocolos (existem muitos mais protocolos que esses), organizados conforme sua camada no Modelo OSI, mas, modelo OSI é conversa para daqui a pouco...

Posição	Nome da Camada	Protocolos
7	Aplicação	HTTP, RTP, SMTP, SSH, Telnet, DNS, POP3, RDP
6	Apresentação	XDR, TLS
5	Sessão	Netbios
4	Transporte	TCP, UDP, DCCP, SCTP
3	Rede	IPv4, IPv6, IP, NAT, ICMP, IPsec
2	Enlace	HDLC, FDDI, Frame Relay, ARP, RARP, CDP, STP
1	Física	USB, Bluetooth, Sonet, DSL, 802.11, Ethernet

Modelo OSI

É impossível ensinar redes sem explicar o Modelo OSI. Apesar desse tópico não estar explícito no blueprint da prova, ele é essencial. Preste atenção em cada detalhe, pois aqui está o pilar de toda sua carreira em redes.

No início, as redes foram desenvolvidas de forma caótica e individual. Cada fabricante tinha sua própria solução, o que chamamos de '**solução proprietária**'. Isso trazia um grande problema, pois a solução de um fornecedor não era compatível com a solução de outros fornecedores. Esta é a principal razão da criação do modelo OSI: **Permitir a interoperabilidade entre os diferentes fabricantes**.

O modelo OSI foi lançado em 1984, pela Organização Internacional de Padronização (ISO). Foi escolhido como modelo ‘oficial’ após anos de estudos e pesquisas com diferentes tipos de modelos. Ele é um modelo aberto (todos os fabricantes sabem exatamente como funciona), com uma abordagem em camadas (veremos mais sobre camadas a seguir). Isso significa que, seguindo esse modelo, é possível construir redes compatíveis entre si. Atualmente, a grande maioria dos fabricantes constrói equipamentos baseadas no modelo OSI.

O modelo OSI não é apenas um modelo para tornar as redes compatíveis; também é uma das MELHORES maneiras de ensinar sobre redes. Tenha isso em mente, pois quando você estiver estudando redes verá que o modelo OSI é frequentemente citado.

Abaixo, um esquema do modelo OSI conforme sua divisão em 07 camadas:



As sete camadas do modelo OSI

Em inglês às 07 camadas são conhecidas como:

- 7 - Application Layer
- 6 - Presentation Layer
- 5 - Session Layer
- 4 - Transport Layer
- 3 - Network Layer
- 2 - Data Link Layer
- 1 – Physical Layer

Como você pode notar, o modelo OSI possui sete camadas; vamos fazer uma abordagem *bottom to the top* (de baixo para cima), para conhecer a função de cada uma dessas camadas. Vamos começar na camada física:

- **Camada física:** Esta camada descreve os elementos no nível físico, como o próprio nome indica: Níveis de tensão, tempo, taxas de transmissão, conectores físicos e assim por diante. Aqui está tudo que você pode “tocar”.

- **Enlace:** Esta camada garante que os dados sejam formatados da maneira correta, ela cuida da detecção de erros e garante que os dados sejam entregues de forma confiável. Isso pode parecer um pouco vago, mas por enquanto, tente se lembrar que é aqui que mora a “Ethernet”. **Grave:** Os endereços MAC e os quadros Ethernet estão na camada de enlace de dados.

- **Rede:** Esta camada cuida da conectividade e da seleção do caminho (roteamento). É aqui que habitam o IPv4 e o IPv6. Cada dispositivo precisa de um endereço exclusivo na rede, e é esta camada que cuida disso.

- **Transporte:** A camada de transporte cuida do transporte (como o nome indica). Nesse momento, você precisa saber da existência de 02 protocolos que trabalham na camada de transporte:

- **TCP:** Protocolo que envia dados de forma confiável.
- **UDP:** Protocolo que envia dados de uma forma não confiável, porém com mais velocidade.

Hora de fazer uma pequena pausa. Essas quatro camadas que acabei de descrever são importantes para a parte de rede - **networking** (a parte física da conexão); as três camadas superiores que veremos a seguir são mais importantes para as aplicações – **applications**.

- **Sessão:** A camada de sessão se encarrega de estabelecer, gerenciar e encerrar as sessões entre dois hosts.
- **Apresentação:** Essa camada garante que as informações sejam legíveis para a camada de aplicação, formatando e estruturando os dados. A maioria dos computadores usa a tabela *ASCII* para caracteres. Se outro computador usar outra tabela de caracteres como *EBCDIC*, a camada de apresentação traduzirá esses caracteres para que os dois computadores possam se comunicar corretamente.
- **Aplicação:** Aqui ‘moram’ os aplicativos: E-mail, navegação na web (HTTP), FTP e muito mais.

Lembre-se: Não é possível pular nenhuma camada no modelo OSI, é impossível pular da camada de aplicação diretamente para a camada de rede, por exemplo. Sempre teremos que passar por todas as camadas para enviar dados pela rede.

Vamos dar uma olhada passo a passo em um exemplo de transmissão de dados na vida real:

1. Você deseja baixar algumas fotos suas que estão hospedadas no Google Drive. Então, você abre o Firefox, e digita a URL do Google drive. O computador enviará uma mensagem ao servidor solicitando aquela determinada página. Ao realizar isso, estamos usando o protocolo HTTP, que como dito anteriormente, está na camada de aplicação.
2. A camada de apresentação irá estruturar as informações da aplicação em um formato inteligível para ambas as partes.
3. A camada de sessão, se certificará de separar todas as diferentes sessões existentes no servidor, afinal, não terá só você acessando o gdrive naquele momento.
4. Dependendo da aplicação, será desejável utilizar um protocolo confiável (**TCP**) ou não confiável (**UDP**) para transferir dados do servidor web. Nesse caso, o escolhido será TCP, pois a aplicação necessita de garantias que a página da web chegue ao seu computador. Discutiremos mais tarde os conceitos de TCP e UDP.
5. Seu computador possui um endereço IP exclusivo (por exemplo: 200.123.2.67), e criará um pacote IP com este endereço como remetente. Este pacote IP, conterá além do seu próprio endereço IP, todos os dados da camada de aplicação, apresentação e sessão. Também especificará qual protocolo de transporte está sendo usado (TCP, neste caso) e o endereço IP do destino (o endereço IP da página).
6. O pacote IP será colocado em um quadro Ethernet. O quadro Ethernet possui o endereço MAC de origem (seu computador) e o endereço MAC de destino (servidor da página). Aprenderemos mais sobre endereços Ethernet e MAC posteriormente.
7. Finalmente, tudo é convertido em bits e enviado pelo cabo usando sinais elétricos.

Mais uma vez, **não é possível “pular” nenhuma camada do modelo OSI**. Sempre teremos que trabalhar com **todas** as camadas.

Vamos fazer uma analogia com a “vida real” para melhor entendimento. Na nossa analogia enviaremos uma carta:

1. Primeiro passo é escrevemos a carta.
2. Colocamos a carta em um envelope.
3. Escrever o nosso nome e o nome do destinatário no envelope.
4. Colocar o envelope na caixa de correio.
5. O conteúdo da caixa de correio irá para a central de processamento dos Correios.
6. O envelope será entregue ao destinatário.
7. O destinatário irá abrir o envelope e ler o conteúdo.

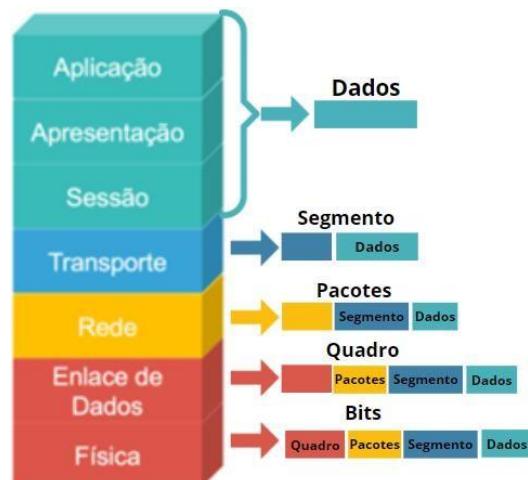
Se colocarmos a carta diretamente na caixa de correio, sem escrever o endereço do destinatário na carta, ela não será entregue, é preciso passar por todo o processo.

Chamamos o processo dos dados saírem da camada de aplicação até a camada física de **encapsulamento**. Quando está sendo realizado o caminho contrário, ou seja, subindo da camada física até a camada de aplicação, está ocorrendo o processo de **desencapsulamento**.

Agora, já conhecemos como opera o modelo OSI, as diferentes camadas e a função de cada uma delas.

Durante a comunicação ponto a ponto (peer-to-peer), cada camada tem pacotes de informações adicionais que são incorporadas aos dados, esses pacotes de informações são chamados de **protocol data units (PDU)**. Cada unidade tem um nome diferente nas diferentes camadas:

- **Camada de transporte:** Nessa camada os PDUs são chamados de **Segmentos**; por exemplo, segmentos TCP (*TCP segments*).
- **Camada de rede:** Nessa camada os PDUs são chamados de **Pacotes**; por exemplo, pacotes IP (*IP packets*).
- **Camada de enlace de dados:** Nessa camada os PDUs são chamados de **Quadros**; por exemplo, quadros Ethernet (Ethernet frames).



Modelo OSI com as respectivas PDU's

Agora que você conhece o modelo OSI é hora de apresentar o outro modelo de rede que temos, o modelo TCP/IP.

Modelo TCP/IP

O Modelo TCP/IP é mais simples que o Modelo OSI, possuindo apenas 04 camadas. Sendo que essas camadas são equivalentes em ambos os modelos. Na verdade, as camadas possuem o mesmo papel, apenas foram agrupadas e receberam outros nomes:

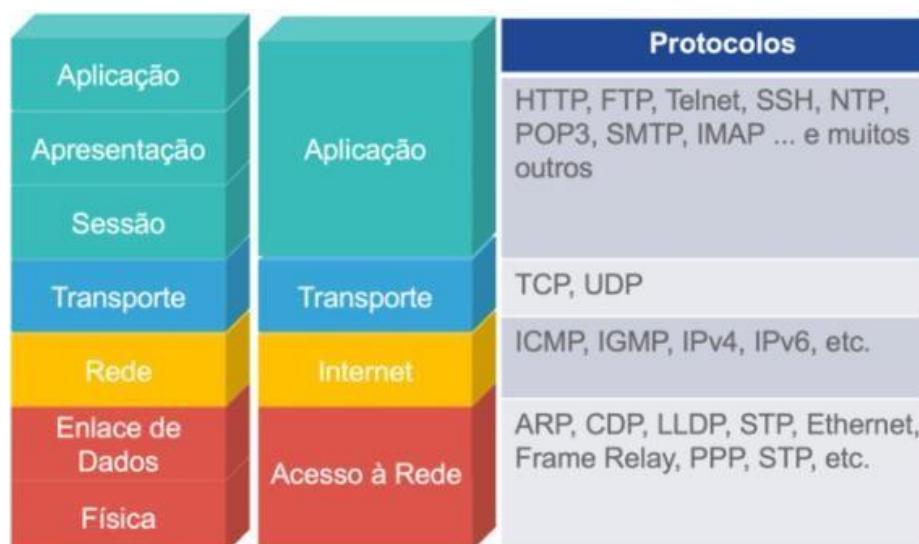


Modelo TCP/IP com suas quatro camadas

Nomenclatura em inglês:

- 4 –Application
- 3 –Transport
- 2 – Network
- 1 – Network Interface

Abaixo uma comparação entre os dois modelos e os principais protocolos que residem em cada camada:



Modelo OSI, TCP/IP e seus respectivos protocolos

Note que não temos muita diferença:

- A Camada de Aplicação no Modelo TCP/IP engloba as três camadas superiores do Modelo OSI;
- O nome da Camada de Rede muda no modelo TCP/IP para Camada de Internet, mas a função é a mesma;
- As Camadas de Enlace de Dados e Física se tornam uma só com o nome de Camada de Acesso à Rede;

Acostume-se com essas terminologias, leia e releia, treine utilizá-las no dia a dia. Conhecer as terminologias em inglês e português fará toda a diferença na hora da prova. Por enquanto não se preocupe em entender o que cada protocolo faz, foque apenas nos mais importantes que detalharei ao longo do livro.

No próximo capítulo vamos entrar no blueprint da prova, a ideia é dissecar o conteúdo na mesma ordem, porém, como explicado anteriormente, algumas vezes por questões pedagógicas não será possível.

Exercícios

- 1) Qual o nome do PDU gerado pela camada 02 do modelo OSI?
 - a) Segmentos
 - b) Pacotes
 - c) Quadros
 - d) Bits
 - e) Datagramas

- 2) Qual o nome da 6^a camada do modelo OSI?
 - a) Redes
 - b) Internet
 - c) Transporte
 - d) Sessão
 - e) Apresentação

- 3) Qual dos protocolos abaixo atua na camada de transporte?
 - a) HTML
 - b) CDP
 - c) TCP
 - d) IPv4
 - e) Telnet

- 4) Qual dos dispositivos abaixo não é um endpoint?
 - a) Laptop
 - b) Access Point
 - c) Impressora
 - d) Computador
 - e) Telefone Celular

- 5) Qual a primeira camada do modelo TCP\IP?
 - a) Acesso à rede
 - b) Internet
 - c) Transporte
 - d) Física
 - e) Enlace de dados

- 6) Um cliente web recebe uma resposta de uma página alocada em um servidor WEB. Do ponto de vista do cliente, qual é a ordem correta da pilha de protocolos usada para decodificar a transmissão recebida?
 - a) HTTP, Ethernet, IP, TCP
 - b) Ethernet, IP, TP, HTTP
 - c) HTTP, TCP, IP, Ethernet
 - d) Ethernet, TCP IP, HTTP

- 7) Qual processo envolve a colocação de uma PDU dentro de outra PDU?
 - a) Segmentação
 - b) Encapsulamento
 - c) Codificação
 - d) Controle de fluxo

- 8) Qual dispositivo desempenha a função de determinar o caminho que as mensagens devem seguir em diferentes redes?
- a) Roteador
 - b) Firewall
 - c) Modem DSL
 - d) Servidor WEB
- 9) Em qual camada OSI um endereço MAC de origem é adicionado a um PDU durante o processo de encapsulamento?
- a) Camada de link de dados
 - b) Camada de transporte
 - c) Camada de aplicação
 - d) Camada de apresentação
- 10) Qual o nome da PDU da camada de transporte?
- a) Dados
 - b) Quadro
 - c) Pacote
 - d) Segmento
 - e) Bits
- 11) Qual afirmação está correta sobre os protocolos de rede?
- a) Os protocolos de rede definem o tipo de hardware usado e como ele é montado nos racks.
 - b) Todas elas funcionam na camada de acesso à rede do TCP\IP
 - c) Eles são necessários apenas para troca de mensagens entre dispositivos em redes remotas
 - d) Eles definem como as mensagens são trocadas entre a origem e o destino
- 12) Em qual camada OSI um endereço IP de origem é adicionado a uma PDU durante o processo de encapsulamento?
- a) Camada de aplicação
 - b) Camada de rede
 - c) Camada de link de dados
 - d) Camada de aplicação
- 13) Quais as duas camadas do modelo OSI têm a mesma funcionalidade que uma única camada do modelo TCP\IP? (Escolha duas)
- a) Sessão
 - b) Física
 - c) Rede
 - d) Transporte
 - e) Link de dados
- 14) Em qual camada OSI um número de porta de destino é adicionado a uma PDU durante o processo de encapsulamento?
- a) Camada de link de dados
 - b) Camada de rede
 - c) Camada de transporte
 - d) Camada de aplicação
- 15) Em qual camada OSI um endereço MAC destino é adicionado a uma PDU durante o processo de encapsulamento?
- a) Camada de link de dados
 - b) Camada de rede
 - c) Camada de transporte
 - d) Camada de aplicação

Respostas: 1- c, 2- e, 3- c, 4- b, 5- a, 6 – b, 7 – b, 8 – a, 9 – a, 10 – d, 11 – e, 12 – b, 13 – b, e; 14 – e, 15 - a

1.1 Explain the role and function of network components

Hora de explicar o papel e a função dos componentes de uma rede. É hora de entendermos o papel que cada dispositivo desempenha, e você, como futuro Analista de Redes, irá trabalhar diretamente com esses dispositivos, seja, projetando redes, administrando ou prestando suporte.

1.1.a Routers (roteadores)

Os roteadores são usados por vários motivos, sendo a finalidade principal descobrir e decidir qual caminho usar para chegar a um determinado destino.

Por exemplo, você mora no Rio de Janeiro e o servidor de e-mail que você utiliza está localizado em São Paulo. Assim como há muitas estradas diferentes se você desejar ir de carro do Rio de Janeiro a São Paulo, há muitos caminhos de redes diferentes a serem seguidos, para enviar o tráfego do seu computador no Rio de Janeiro até o servidor de e-mail em São Paulo.

Fisicamente os roteadores podem ser tão pequenos quanto os que usamos em casa ou em uma pequena empresa:



Roteador geralmente utilizado em residências e pequenas empresas

Ou, podem ser esses modelos de entrada da Cisco, utilizados em pequenas e médias empresas:



Roteadores Cisco utilizados em médias empresas, geralmente da linha 1900/2800

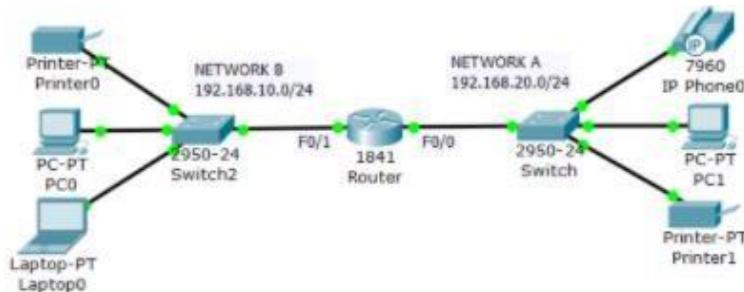
Podem ser enormes, como os roteadores usados por provedores de Internet como a AT&T, Embratel. Esses roteadores lidam com uma quantidade gigantesca de tráfego:



Roteadores utilizados em Data Centers e operadoras

Para entender melhor a função de um roteador, observe a imagem abaixo. Nela temos duas redes diferentes: **Rede A (Network A)** e **Rede B (Network B)** e um roteador (**router**) tornando possível a comunicação entre essas redes.

Essa imagem foi retirada de um programa emulador criado pela Cisco, chamado Packet Tracer. Com ele, é possível emular redes e equipamentos. Falaremos mais sobre emuladores|simuladores e como eles podem auxiliar na preparação para o CCNA nos próximos capítulos.



Topologia retirada do programa Cisco Packet Tracer da Cisco.

Importante: Roteador é um dispositivo que opera na camada **03** (layer 03, **network layer**) do modelo OSI. É nessa camada que está o endereçamento IP, os roteadores encaminham tráfego com base nesses endereços. Mais pra frente, entenderemos melhor como funciona todo esse processo decisório.

Os principais protocolos que os roteadores utilizam para fazer o roteamento do tráfego são o EIGRP, OSPF, RIP, BGP, mas uma vez, não se preocupe agora com essas ‘siglas’, no momento certo você entenderá o papel de cada um deles.

Os modelos de roteadores da linha ISR 4000 e o ASR 1000 Series, estão ganhando muito espaço, vale a pena pesquisar no site da Cisco as funcionalidades e como eles se diferenciam entre si das outras linhas de roteadores.

1.1.b L2 and L3 switches

Em redes locais, normalmente usamos switches para conectar computadores e servidores à rede, permitindo assim, que esses dispositivos se comuniquem entre si. Esses switches podem ser pequenos como os switches que temos em casa ou gigantescos como os switches de data centers.

A seguir veremos alguns exemplos de switches:



Switch com 05 portas, geralmente utilizado em residências.

Switch mais popular da Cisco utilizado para ligar dispositivos finais de um mesmo setor, andar.



Switch Catalyst2900.

Os switches abaixo são utilizados em grandes empresas, eles possuem dezenas, as vezes centenas de interfaces para conexão de computadores, servidores, etc.



Switches modulares da linha Nexus 9500

Importante: Switch é um dispositivo que opera na **camada 02** (layer 02, **data link layer**) do modelo OSI, embora alguns modelos também operem na **camada 03**. Switches operando na camada 02, significa que podem apenas comutar tráfego dentro da mesma rede, já os que operam também na camada 03, podem comutar tanto o tráfego interno quanto rotear este tráfego para outras redes.

Entre os principais protocolos utilizados pelos switches podemos destacar o Rapid PVST, Spanning Tree e LACP. As interfaces dos switches podem ser basicamente de dois tipos:

- 1- **Access Mode** (De acesso): Nesse tipo de interface são ligados os dispositivos finais, como computadores, telefones e impressoras
- 2- **Trunk**: Esse tipo de interface é utilizado para conectar um switch a outro switch.

Os switches não utilizam endereço IP para comunicarem com os dispositivos, eles utilizam outro endereço, o chamado endereço MAC.

Endereço MAC:

O endereço **MAC** (Media Access Control ou Controle de Acesso ao meio) é um endereço físico único, que é associado à interfaces de comunicação utilizadas em dispositivos de rede. A identificação é gravada em hardware por fabricantes de placas de rede, tornando-se posteriormente, parte de equipamentos como computadores, roteadores, smartphones, tablets, impressoras de rede e diversos outros equipamentos.

Como a identificação é única, ela também pode ser usada para fazer “controle de acesso”, porém, apesar de ser único e gravado em hardware, é possível alterar o endereço MAC com técnicas específicas.

É importante destacar que, embora não seja algo visível, sempre que a rede utiliza uma identificação baseada em software como o protocolo TCP/IP, o endereço MAC é utilizado.

Os endereços MAC possuem uma padronização que é administrada pela **IEEE** (Institute of Electrical and Electronics Engineers). Basicamente, ele é formado por um conjunto de seis bytes separados por dois pontos ou hífen, e cada byte é representado por dois algarismos na forma hexadecimal, como por exemplo: “00:1B:C9:4B:E3:57”

Switches são usados na maioria das vezes para conectar dispositivos finais como: Telefones, impressoras, computadores, servidores, etc, como demonstrado na imagem a seguir:

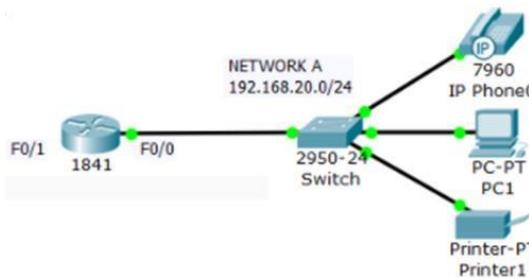


Imagen de uma topologia construída no Cisco Packet Tracer da Cisco

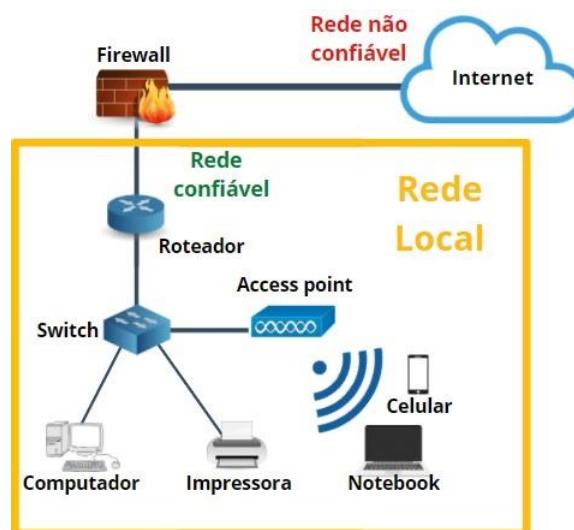
Observe que o switch está conectado a um roteador. Caso algum desses computadores necessite acessar a Internet, o switch encaminhará o tráfego para o roteador, e este, fará o direcionamento da rede local (LAN) para a Internet.

Os modelos que estão ganhando mais destaque no momento são os da linha Catalyst 9200, Catalyst 9400, Catalyst 9300, Catalyst 9500, porém, o mais popular ainda é o Catalyst 2960.

1.1.c Next-generation firewalls and IPS

Firewall é uma barreira entre uma rede segura (confiável) e uma rede não confiável. Geralmente, é utilizado para separar a rede interna (Lan) de uma rede externa (Internet).

Um firewall pode ser um hardware, software ou ambos, ele é responsável por monitorar o tráfego de entrada e saída da rede, decidindo se aquele tráfego será permitido ou bloqueado de acordo com um conjunto definido de regras de segurança.



Topologia com firewall separando Lan da Internet

Tipos de firewall

Vamos ver os diferentes tipos de firewall existentes e a forma que cada um deles é cobrado na prova.

- Firewall de proxy:** O firewall de proxy foi um dos primeiros tipos de firewall, ele funciona como passagem de uma aplicação específica de uma rede para outra. Servidores proxy podem oferecer recursos adicionais, como armazenamento em cache e segurança de conteúdo, ao evitar conexões diretas de fora da rede. No entanto, isso também pode afetar a capacidade de taxa de transferência e nem todas as aplicações comportam esse tipo de funcionamento.
- Firewall com inspeção de estado:** Conhecido como o firewall tradicional. Um firewall com inspeção de estado permite ou bloqueia tráfego de acordo com o estado, a porta e o protocolo. Ele monitora toda atividade desde o momento em que uma conexão é aberta até ela ser fechada. As decisões de filtragem são tomadas de acordo com as regras definidas

pelo administrador e com contexto, o que significa o uso de informações de conexões e pacotes anteriores que pertencem à mesma conexão.

- **Firewall de gerenciamento unificado de ameaças (UTM):** Normalmente, um dispositivo UTM combina de maneira flexível as funções de um firewall com inspeção de estado e prevenção contra intrusões e antivírus. Ele também pode incluir serviços adicionais e, às vezes, gerenciamento em nuvem.
- **Firewall de próxima geração:** Esse tipo de firewall é uma combinação dos firewalls tradicionais com funcionalidades típicas de outros equipamentos de rede, entre essas funcionalidades é possível destacar Deep packet inspection (**DPI**) e Intrusion Prevent System (**IPS**).

Para o CCNA é fundamental conhecer as definições dos tipos de firewall, principalmente a nova geração de firewall que falaremos a seguir:

Next-generation firewalls– NGFW

O firewall de próxima geração (Next-generation firewall) também é conhecido como firewall de terceira geração. De acordo com a definição do Gartner, Inc., um firewall de próxima geração deve incluir:

- Recursos padrão de firewall, como inspeção stateful;
- Prevenção integrada de invasão;
- Reconhecimento e controle das aplicações para detectar e bloquear aplicativos nocivos;
- Formas de atualização para incluir feeds futuros de informação;
- Técnicas para lidar com ameaças à segurança ainda não exploradas.

O exame CCNA 200-301 menciona os termos firewall e IPS sempre precedido do termo *next generation*, aqui cabe uma pequena volta ao passado para entender esse termo.

Nos primeiros anos da década de 2010, a Cisco e alguns de seus concorrentes começaram a usar o termo *próxima geração* para falar dos seus produtos de segurança, o objetivo era enfatizar alguns dos recursos mais recentes, diferenciando esses dispositivos da nova geração da geração antiga de dispositivos de segurança.

Resumindo, um firewall de próxima geração (**NGFW**) e um IPS de próxima geração (**NGIPS**) são os firewalls e IPS atuais da Cisco. Entretanto, o uso do termo próxima geração vai muito além de apenas um rótulo de marketing, os dispositivos realmente sofreram transformações e atualizações significativas.

Segurança digital é um ciclo interminável de novos tipos de ataques seguidos por novas soluções, sendo que algumas dessas soluções exigem novos recursos dos dispositivos já existentes, ou até mesmo, novos produtos.

Um exemplo dos novos desafios são os dispositivos móveis BYOD. BYOD é uma sigla para Bring Your Own Device, em português “*traga seu próprio dispositivo*”. É um conceito de que consiste na utilização dos aparelhos dos próprios funcionários para desempenhar as atividades empresariais. Eles são um desafio tremendo em questão de segurança, pois saem da empresa com o funcionário (o funcionário pode navegar na Internet, baixar arquivos, etc) e depois retornam à empresa, criando diversos tipos de riscos.

Retornando ao ano de 2013, a Cisco adquiriu uma empresa de segurança chamada Sourcefire, e com ela o modelo e tecnologia inicial dos firewalls de última geração e IPS. Os firewalls vendidos atualmente pela Cisco, ainda têm nomes atrelados a Sourcefire, como a linha de produtos de firewall chamados de **Firewalls Cisco Firepower**. Inclusive, essa linha Firepower é a substituta dos populares firewalls Cisco ASA (Adaptive Security Appliance).

Diferente de um modelo tradicional de firewall, que faz controle apenas dos IPs de origem e destino, portas de origem e destino e flags, um Next Generation Firewall vai além, com análises mais profundas no pacote que está trafegando através dele. Vamos a alguns exemplos práticos:

- Em um NGFW, é possível analisar se um download que está sendo realizado contém algum tipo de ameaça, como ransomware ou outro malware qualquer, seja ele conhecido (que já tenha uma assinatura) ou desconhecido (zero day).

Nesse último caso, a análise é realizada em uma ‘sandbox’ local ou na nuvem. *Sandbox* é um ambiente isolado onde arquivos ou aplicativos suspeitos podem ser executados, examinados e sondados antes de passarem pelo firewall e terem acesso à rede.

- O NGFW pode agregar funções de IPS, ou seja, analisar dentro cada pacote para verificar a existência de ferramentas voltadas a exploração de vulnerabilidades em algum serviço, por exemplo: Apache, RDP, Oracle, Tomcat, JBoss, SSH, Nginx, SQL Server.
- Outra funcionalidade extremamente importante é a de URL Filtering, onde é possível controlar o acesso a sites não desejados com base nas políticas da empresa, assim, é possível evitar incidentes de segurança ou o uso indevido dos recursos da rede (uso de torrents e streaming) além de outras situações não desejadas.
- Oferece também prevenção contra vazamento de dados (**DLP**) sensíveis para o negócio.



ASA 5500-X Series Next-Generation Firewalls.

IPS -Intrusion Prevent System:

O **IPS** geralmente fica conectado diretamente atrás do firewall, fornecendo uma camada complementar de análise que seleciona negativamente conteúdo perigoso. Ao contrário de seu antecessor, o Sistema de Detecção de Intrusão (**IDS**) — que é um sistema passivo que verifica o tráfego e informa sobre ameaças — o **IPS** é colocado *inline* (no caminho de comunicação direta entre fonte e destino), analisando ativamente e tomando ações automatizadas em todos os fluxos de tráfego que entram na rede. Especificamente, essas ações incluem:

- Enviar um alarme ao administrador;
- Bloquear os pacotes maliciosos;
- Bloquear o tráfego enviado por determinado endereço de origem.

Como é um componente de segurança inline (fica no meio do caminho), o Intrusion Detection System deve ser configurado com extrema atenção, pois qualquer erro causará degradação no desempenho da rede. Falhas de configuração também podem aumentar o número de falsos positivos (pacotes legítimos interpretados como ameaças), o que constitui um grande desafio, pois o IPS trabalha com uma grande quantidade de dados em tempo real.

O **IPS** tem uma série de métodos de detecção para encontrar vulnerabilidades, as três principais são: Detecção baseada em assinatura, detecção baseada em anomalias estatísticas e detecção baseada em diretivas. Vamos analisar cada um desses modelos de detecção!

1. **Detecção baseada em assinatura** - A detecção baseada em assinatura faz uso de um dicionário de padrões (ou assinaturas), exclusivamente identificáveis no código de cada ameaça. À medida que uma atividade é descoberta, sua assinatura é gravada e armazenada em um dicionário de assinaturas que segue crescendo continuamente.
2. **Detecção baseada em anomalias** - A detecção de anomalias estatísticas analisa amostras de tráfego e as compara a um nível de desempenho pré-calculado. Quando a amostra da atividade de tráfego está fora dos parâmetros do desempenho dessa baseline (que foi calculada anteriormente), o **IPS** entra em ação, tomando medidas para lidar com a situação.

3. **Detecção baseada em diretivas** - O dispositivo é programado com um conjunto de normas e regras de operação. Sempre que alguma atividade violar uma das diretivas previamente programadas, o IPS interrompe a atividade em questão e notifica o administrador da rede sobre esta ocorrência.



Sensor Cisco IPS 4240

1.1.d Access Points

Access point ou simplesmente “pontos de acesso”, são dispositivos que permitem as pessoas conectarem dispositivos sem fio como laptops, tablets e celulares a uma rede cabeadas. O Access point geralmente está conectado a um roteador wireless (*roteador sem fio*).



Access point Cisco Aironet 702i



Representação de um Access Point

Basicamente, o access point pode ser compreendido como um tipo de repetidor Wi-Fi, que usa cabos e não pode ser usado como um substituto a um roteador, é disponível apenas como 'appliance' (Appliance é um dispositivo de hardware separado e dedicado com software integrado, especificamente projetado para fornecer um recurso de computação específico).

Access Points têm algumas características básicas associadas, sendo as principais: **Velocidade e capacidade de gerenciamento de rede**.

1.1.e Controllers (Cisco DNA Center and WLC)

Abaixo estudaremos o que são controladoras e os principais equipamentos na Cisco nessa área:

Controllers

Também traduzida com “controlador” ou “controladora”, podemos defini-la como um dispositivo (pode ser um appliance ou embarcado em outras caixas) que fornece um único ponto de gerenciamento para dispositivos de uma rede. Temos vários tipos de *controllers*, porém no momento vamos focar nas duas exigidas nesse tópico do exame.

Cisco DNA Center

Cisco DNA Center (Digital Network Architecture) é um ‘Appliance’ que traz uma plataforma onde é possível gerenciar, verificar, monitorar, configurar e automatizar todos os dispositivos de rede da empresa (roteadores, switches, etc) em um só portal em formato WEB.

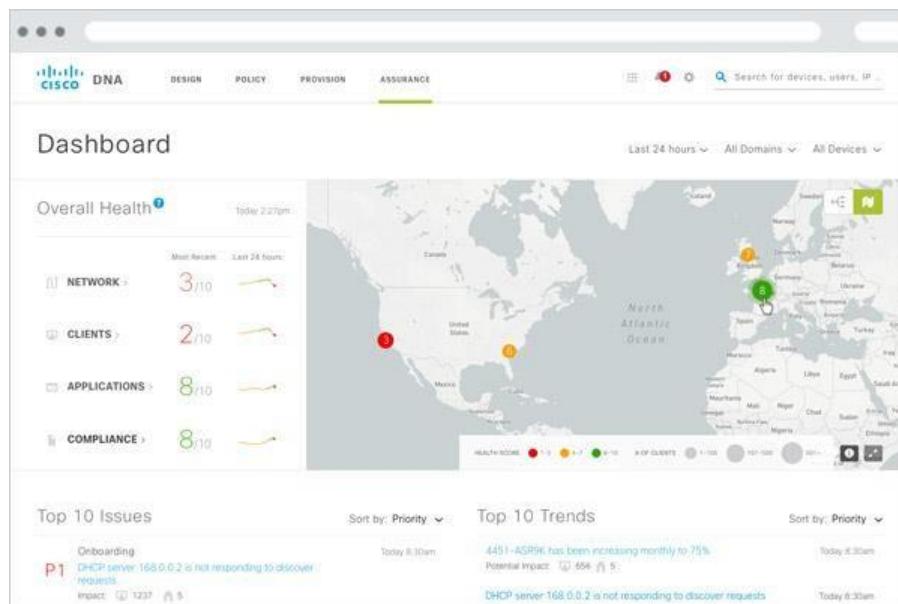
Aqui entramos no conceito de SDN (Software Defined Network), ou seja, software definindo as redes, ou simplesmente automatização da rede. O funcionamento do Cisco DNA Center é bem simples, basta instalar a ‘caixa’ na rede (literalmente plug-and-play) e ele fará o auto-discovery (descoberta automática) da topologia e mostrará em tempo real (real-time) como estão todos os equipamentos na rede.



Cisco DNA Center UCSC-C220-M5SX



Cisco DNA Center UCSC-C480-M5



Dashboard do Cisco DNA Center

Em suma, Cisco DNA Center automatiza a rede substituindo as tarefas manuais por um processo mais ágil e eficaz, com base em quatro conjuntos de recursos: visibilidade, intenção, implantação e gerenciamento. Vejamos cada uma delas:

- **Visibilidade:** Descobre os dispositivos automaticamente obtendo visibilidade ponta a ponta da rede e montando um inventário atualizado.
- **Intenção:** Interpreta o desejo de como você pretende que sua rede funcione e transforma essa intenção em automação.
- **Implantação:** Facilita a implantação de um novo dispositivo. Basta conectar esse dispositivo de rede para integrá-lo automaticamente.
- **Gerenciamento:** Ajuda a garantir a consistência e o desempenho da rede de ponta a ponta realizando o gerenciamento de software adequado e atualizado para cada dispositivo.

WLC (Wireless Lan Controller)

Os dispositivos da série Cisco Wireless Controller (WLC), fornecem uma solução única para configurar, gerenciar e oferecer suporte a redes sem fio corporativas, independentemente de seu tamanho e localização. Esses dispositivos tornaram-se muito populares durante a última década, conforme as empresas mudam os **designs de implantação de pontos de acesso (AP) independentes** para o **design baseado em controladoras centralizadas**, colhendo os benefícios de **funcionalidade aprimorada** e redundância que este tipo de design oferece.

Embora todos os modelos de **WLC** possuam configuração baseada em **GUI** (interface gráfica) e **CLI** (interface de linha de comando), os WLCs são frequentemente configurados por meio de sua GUI, uma interface web muito bem projetada (Cabe o elogio, pois aqui, a Cisco acertou exatamente onde ela sempre erra: Interface gráfica). O CLI é obrigatório apenas durante a configuração inicial, onde o administrador deve atribuir um endereço IP ao dispositivo WLC, junto com alguns outros parâmetros importantes.

Uma das vantagens da interface gráfica do WLC é que a GUI é idêntica em todos os modelos.



WLC 8500



WLC 7500

Como afirmado anteriormente, o contrário de outros produtos Cisco, a interface GUI do WLC é extremamente bem projetada e com um layout preciso e bem dividido. Ao efetuar login no WLC GUI, o administrador recebe informações essenciais, incluindo uma vista frontal da controladora, onde é possível ver o status de cada porta física e uma infinidade de detalhes que facilita muito a resolução de troubleshootings.

A screenshot of the Cisco WLC2504 web interface. The top navigation bar includes links for MONITOR, WLANs, CONTROLLER, WIRELESS, SECURITY, MANAGEMENT, COMMANDS, HELP, and FEEDBACK. The main content area is titled 'Summary' and features a large graphic showing '46 Access Points Supported' and a 'Cisco 2500 Series Wireless Controller'. Below the graphic, the 'Controller Summary' section displays various system details like Management IP Address (192.168.4.10), Software Version (7.4.121.0), and Up Time (1 days, 12 hours, 16 minutes). The 'Rogue Summary' section shows Active Rogue APs (22), Active Rogue Clients (0), and Adhoc Rogues (0). The 'Top WLANs' section lists profiles with their respective client counts. The 'Access Point Summary' section shows the status of 42 radios across different interfaces. The 'Client Summary' section provides a breakdown of current, excluded, and disabled clients. At the bottom, there are sections for 'Most Recent Traps' and 'Top Applications'.

Interface web da página inicial WLC2504

Observe a praticidade, é mostrado na página inicial todas as informações necessárias para uma verificação de rotina:

- Estado visual das portas físicas da controladora;
- Status do hardware da controladora (tempo de atividade, endereço IP, uso de CPU/memória, temperatura, versão do firmware, etc.);
- Resumo dos pontos de acesso conectados e interfaces up/down;

- Clientes atualmente conectados (em todas as redes sem fio);
- Principais redes sem fio e número de clientes conectados a cada uma;
- Pontos de acesso e clientes invasores que foram detectados.

1.1.f Endpoints

Endpoint Devices (Dispositivos finais) – São os equipamentos que precisam acessar a rede, tipicamente os equipamentos que os usuários utilizam.

- Computador;
- Impressora;
- Smartphones;
- Tablets;
- Telefones.

Não há distinção entre serem parte de uma rede cabeada (Wired) ou sem fio (wireless)

1.1.g Servers

Em geral Servidores podem ser definidos como um computador\software que fornece serviços específicos para um ou vários clientes. Esses servidores podem armazenar\disponibilizar acesso a inúmeras aplicações:

- Servidor de dados;
- Email;
- Aplicações;
- Hypervisor.

São infinitas as possibilidades e utilizações de servidores. Geralmente, encontraremos esses dispositivos conectados a switches de Data Centers, que possuem maior largura de banda e poder de processamento, porém em pequenas empresas eles estarão dentro dos CPDs.

Um erro grande que muitos iniciantes cometem, é achar que servidores tem de ser parecidos com os computadores de mesa que utilizamos (desktop), alguns parecem mais com um switch.



Servidor rack Cisco UCS C240 M5

1.2 - Describe characteristics of network topology architectures

Entramos agora na segunda parte do primeiro tópico do exame, nessa parte vamos estudar e entender as principais características de uma rede e as diversas arquiteturas possíveis.

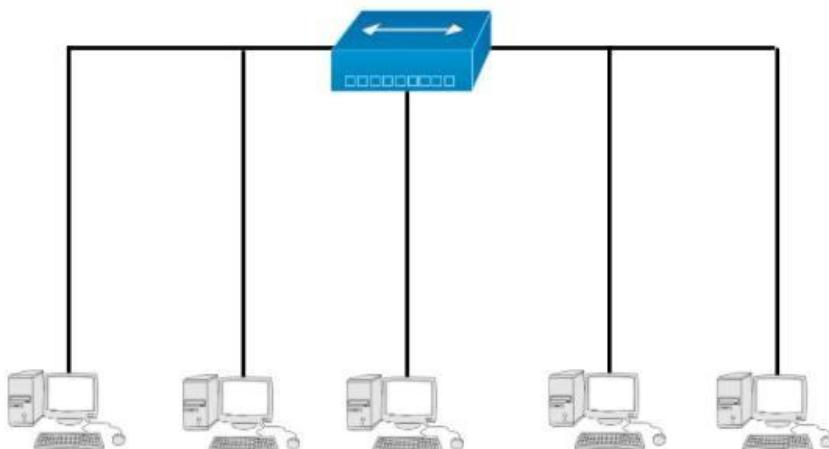
Antes de entrarmos nos tópicos do exame propriamente dito, é necessário entendermos como é formada uma Lan e porque arquiteturas diferentes são necessárias.

Introdução a arquitetura de redes

Imagine uma rede corporativa de uma grande empresa com dezenas, centenas ou até milhares de usuários. Esses usuários estão divididos em andares de vários edifícios com seus computadores e telefones conectados a diversos switches diferentes. Esses switches estão conectados entre si e a dispositivos de saída para Internet como os roteadores.

Para que toda essa estrutura funcione a contento, é necessário que tenhamos um bom design físico para conectar esses switches fisicamente entre si, e é fundamental um bom design lógico para não sobrecarregar a rede com tráfego desnecessário.

Vamos simular alguns cenários com equipamento diferentes, analisar como as redes "crescem" e alguns problemas de design que podem advir desse crescimento caso não seja bem planejado.



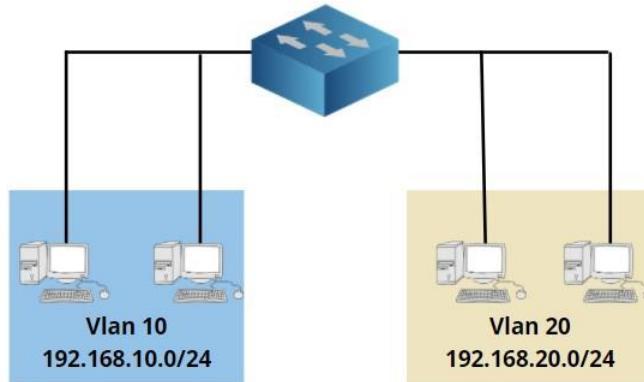
5 computadores ligados em um hub

Quase não vemos mais **hubs** atualmente, mas antigamente eram bem populares. Os **hubs** são dispositivos 'burros' que operam na camada 01 do **Modelo OSI**. Explico, todo sinal recebido em uma interface do hub é transmitido para todas as outras interfaces e não apenas para aquela que o dispositivo de destino está conectado, formando assim um único **domínio de colisão** e um único **domínio de broadcast**. Devido às limitações dos hubs as redes operavam em **half-duplex**, ou seja, quando um host transmitia algo, os outros tinham que esperar. Quando dois hosts enviam tráfego ao mesmo tempo, ocorria uma colisão e os pacotes se perdiam. Para contornar esse problema de colisões constantes, foi criado o algoritmo **CSMA/CD**.

Neste exemplo existem apenas 5 hosts, então não há muito problema com colisões, mas se tivéssemos centenas de hosts, as transmissões sofreriam um sério impacto devido as colisões frequentes que ocorreriam mesmo utilizando o protocolo **CSMA/CD**.

A solução ideal em cenários assim, é reduzir o tamanho do domínio de colisão, por isso começamos a utilizar bridges (pontes) ou switches (que possuem mais portas). E para diminuir os domínios de broadcast a solução é a implementação de **VLANs**.

Veja o exemplo a seguir:

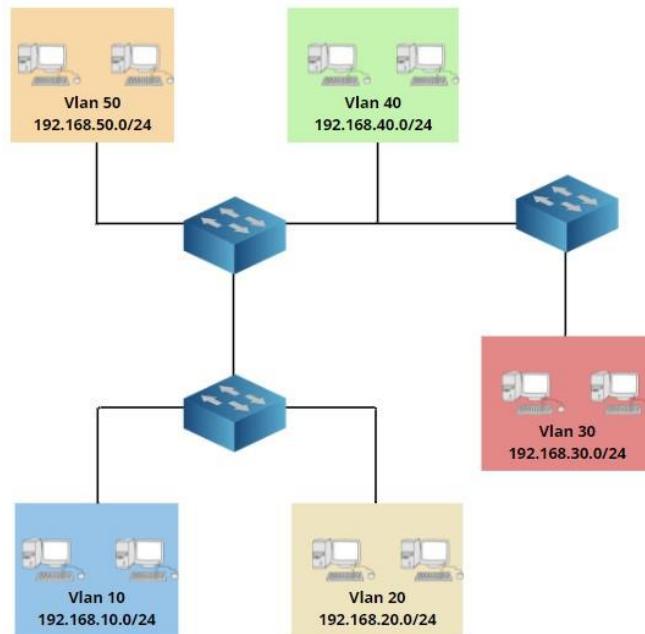


Switch conectando 04 computadores a 02 VLANs diferentes.

Observe que agora temos um switch com quatro hosts que estão divididos em duas VLANs diferentes. Cada porta do switch é um domínio de colisão e cada VLAN é um domínio de broadcast. Se usarmos um switch **layer 03**, as VLANs poderão se comunicar umas com as outras sem a necessidade de um roteador.

À medida que essa rede for crescendo as portas desse switch não serão mais suficientes para conectar todos os dispositivos a rede. A solução, é adicionar um segundo switch, conectando este switch ao primeiro. Mas, e se a rede continuar crescendo e precisarmos de mais portas vagas para conectarmos mais dispositivos na rede? Neste caso, a solução seria adicionarmos um terceiro switch ou quarto switch. Mas qual seria a maneira certa de conectar esses switches?

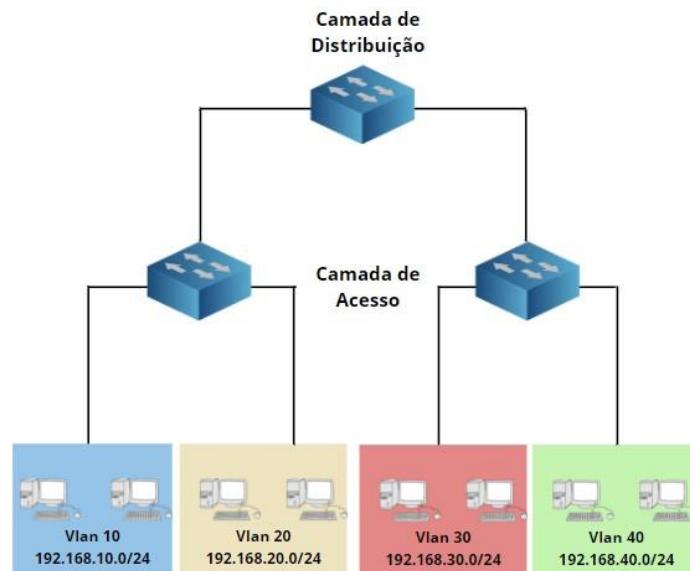
Se não houver um design bem planejado corremos o risco de produzirmos algo assim:



Switches conectados sem hierarquia definida.

Observe que não há nenhum sentido nesse tipo de ligação: Switches, computadores, cabos e VLANs estão plugados de qualquer jeito, sem a mínima lógica. A rede se tornou uma imensa bagunça, dificultando e muito um possível troubleshooting.

Uma rede bem projetada precisa ser de fácil manutenção, oferecer alta disponibilidade, escalabilidade e ser capaz de responder rapidamente às mudanças na topologia. Para conseguir atingir esses objetivos, a Cisco criou uma abordagem hierárquica com várias camadas. Observe o exemplo abaixo:



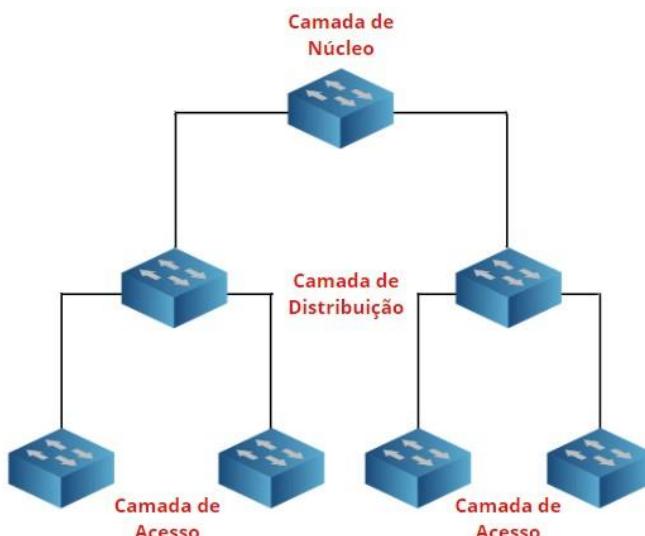
Switches conectados seguindo hierarquia de duas camadas.

Neste exemplo, utilizamos 02 camadas para racionalizar a rede: Camada de Acesso (**Access Layer**) e Camada de Distribuição (**Distribution Layer**).

A **camada de Acesso** fica mais próxima aos usuários finais, é formada pelos switches que usamos para conectar computadores, laptops, Access Point e todos os chamados *endpoints*.

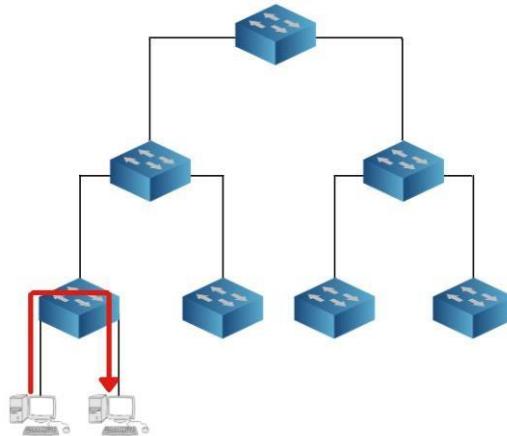
A **Camada de Distribuição** é usada para agregar todos os switches da camada de acesso.

A principal vantagem desse design de rede hierárquico é que ele é escalonável. Quando a rede crescer e houver mais usuários, prédios e andares, poderemos adicionar várias **camadas de distribuição**, nesse momento, será hora de adicionarmos outra camada, a **Camada Core (camada de núcleo)**.



Switches conectados seguindo hierarquia de três camadas.

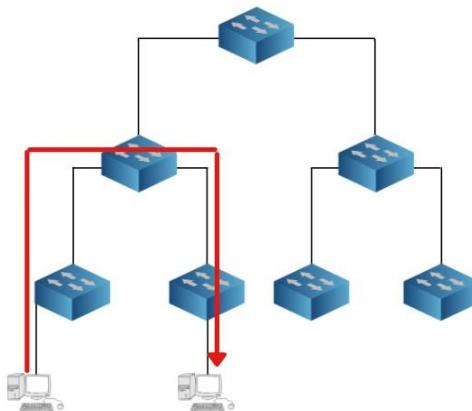
A Camada de Núcleo (**Core Layer**) agrupa todos os switches da **Camada de Distribuição**. Esse design também torna previsível e fácil de visualizar o tráfego da rede. Basicamente, existem três fluxos de tráfego diferentes:



Exemplo de tráfego local, ou seja, o tráfego não sai de dentro do mesmo switch.

Todo o tráfego começa na camada de acesso e, se necessário, sobe para a camada de distribuição e núcleo. Neste exemplo, o tráfego é local, ou seja, ele não sai do switch da camada de acesso. O tráfego entre dois hosts na mesma VLAN em uma rede bem projetada, ficará sempre no mesmo switch.

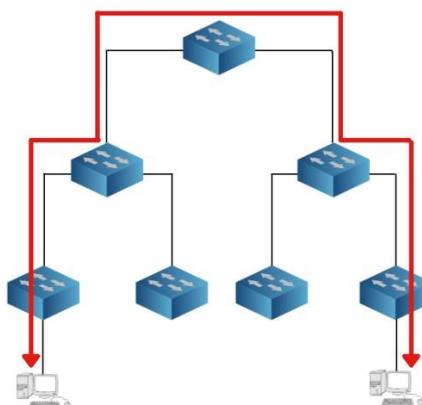
Outro exemplo:



Exemplo de tráfego que utiliza 02 switches de acesso.

O tráfego entre hosts que atravessam diferentes switches da camada de acesso deve, obrigatoriamente, cruzar um switch da camada de distribuição.

Finalmente, às vezes, é necessário atravessar a camada central:



Exemplo de tráfego que passa pela camada central.

Camada de Acesso

A principal função da camada de acesso é conectar os dispositivos finais como computadores, laptops, pontos de acesso, telefones IP, impressoras, etc.

Principais características da Camada de Acesso:

- Para conectar todos esses dispositivos finais são necessários centenas, dependendo do tamanho da empresa, até milhares de interfaces, isso faz com que normalmente esses switches de acesso sejam de menor custo, por esse motivo, normalmente usamos **switches da camada 2** na camada de acesso.
- Dependendo do fluxo do tráfego, o tráfego da camada de acesso deve seguir em direção à camada de distribuição, portanto, deverá haver vários uplinks (interfaces trunk que ligam um switch a outro).
- POE (*Power over Ethernet*): Se houver telefones IP ou pontos de acesso sem fio, será necessário switches com PoE na camada de acesso.
- Recursos de QoS: Caso seja utilizado VoIP pode ser necessário a utilização de switches que ofereçam suporte a QoS (Quality of service), para dar precedência ao tráfego do VoIP.
- Segurança: A camada de acesso é porta de “entrada” da rede, por isso, é necessário protegê-la de ataques advindos de camada 01 e 02.

Camada de Distribuição

A camada de distribuição conecta a camada de acesso a camada central. Como estudado, é aqui que agregamos todos os switches da camada de acesso, por isso, é necessário que haja uplinks com grande largura de banda até a camada central. Normalmente, a camada de distribuição é onde usamos roteamento e é nela que ‘morrem’ as VLANs da camada de acesso.

Principais características da Camada de distribuição:

- Usamos roteamento na camada de distribuição, portanto, precisamos de switches que sejam capazes de desempenhar a função de roteamento e em alto rendimento.
- Deve haver múltiplos uplinks redundantes para a camada de acesso e camada de núcleo. Se um switch da camada de distribuição falhar, os switches da camada de acesso podem perder a conectividade, por isso é importante a redundância nos uplinks.
- Recursos de QoS: Tal como na camada de acesso, podemos precisar de QoS para dar preferência a determinado tráfego como VoIP.
- Segurança: Usamos listas de acesso na camada de distribuição para filtrar tráfego entre VLANs.

Camada Central

O núcleo agrupa todos os switches da camada de distribuição, este é o backbone da rede. Isso significa que, os switches na camada central devem ser capazes de lidar com todo o tráfego dos switches da camada de distribuição. Além disso, se o núcleo falhar, toda a conectividade entre as camadas de distribuição será interrompida.

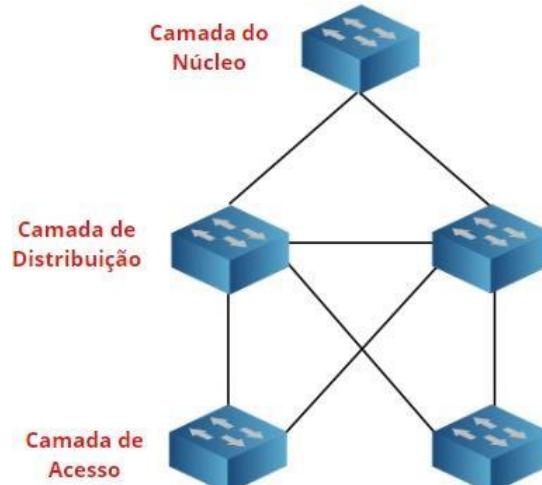
Principais características da Camada Central:

- Grandes taxas de largura de banda são exigidas nessa camada, pois, a quantidade de dados que trafegam por ela é geralmente alta.
- Alta disponibilidade, ou seja, necessita de redundância. Essa redundância inclui vários links, fontes de alimentação, e supervisores redundantes (CPU).
- Recursos de QoS: QoS é implementado de ponta a ponta na rede, portanto, também precisamos dela na camada do núcleo.
- Sem manipulação de pacotes: Não configuramos listas de acesso ou fazemos quaisquer alterações nos pacotes nessa camada.

Às vezes, o tamanho da rede é muito pequeno para justificar uma camada central separada. Neste caso, a função do núcleo e da camada de distribuição é combinada em uma única camada. Essa camada recebe o nome de **núcleo colapsado**.

O modelo de três camadas é bastante simples, mas ainda não falamos sobre redundância entre os dispositivos. Em todos os modelos que vimos até agora, tínhamos apenas um link entre os switches.

Primeiro, vamos ver a redundância entre a camada de acesso e distribuição:

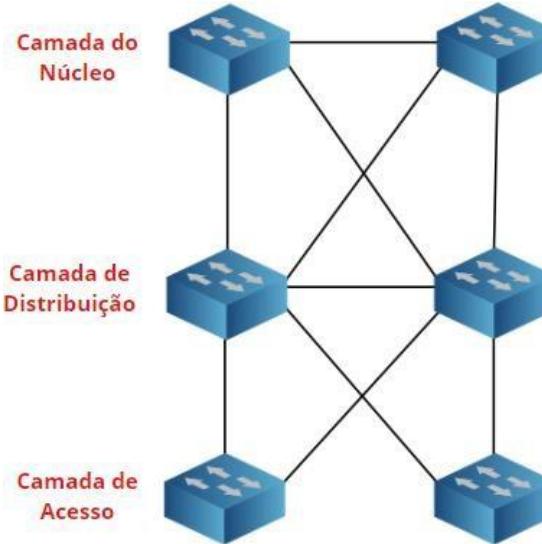


Modelo de designer com três camadas

Observe o exemplo acima, adicionamos redundância entre os switches. Além disso, em vez de um único switch na camada de distribuição, agora temos dois, e cada switch da camada de acesso está conectado a ambos os switches da camada de distribuição.

Observe também o link conectando os switches da camada de distribuição, ele é necessário para que eles possam alcançar um ao outro diretamente e também será necessário quando utilizamos protocolos de roteamento.

Porém, ainda temos um problema: Não há redundância no núcleo. Precisamos adicionar mais um switch a rede:



Modelo de designer com três camadas e redundância no núcleo.

O núcleo é o backbone da rede, é a parte mais importante, portanto, precisamos de redundância nele.

Com esse switch a rede ficou mais completa, e com isso mais complexa. Observe que também foram adicionados mais alguns links redundantes entre os switches da camada de distribuição.

O design acima ainda é bastante simples, com apenas algumas opções. E se adicionarmos mais camadas de distribuição e múltiplas camadas de acesso? Devemos conectar todos os switches da camada de distribuição aos outros? E quanto aos switches da camada de acesso? Devemos conectá-los a todos os switches da camada de distribuição?

Chega um determinado momento que não é mais viável ligar todos os switches de acesso em todos os switches da camada de distribuição, então, criamos os chamados **blocos de switches (switch blocks)**. Um bloco de switches é formado por dois switches da camada de distribuição conectados com os switches da camada de acesso abaixo deles. Cada bloco de switch é então conectado à camada central:

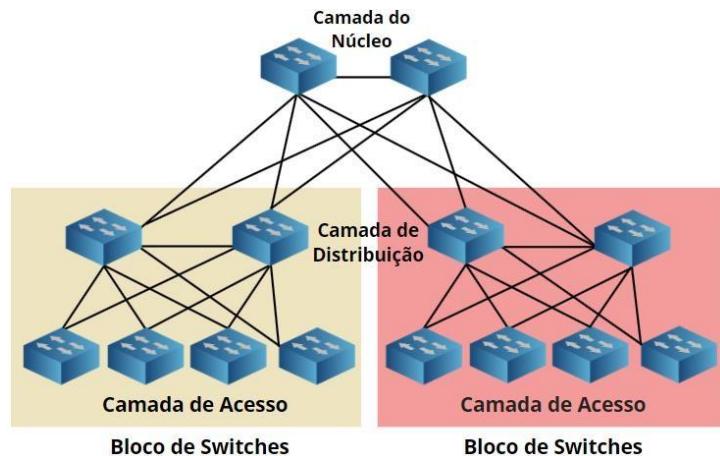


Diagrama com blocos de switches.

Este é um diagrama de rede hierárquico bonito e organizado (dificilmente você verá isso na vida real). Note que há dois blocos de switch, em cada bloco de switch temos dois switches na camada de distribuição e há vários switches da camada de acesso. Não há conectividade entre os diferentes blocos de switch diretamente, eles se conectar apenas entre si e com a camada de núcleo. Os blocos de switch também são usados para conectar datacenters à camada central.

O tamanho de um bloco de switch depende diretamente do número de usuários, das aplicações e tipo de tráfego que passarão por ele. A análise de rede é necessária para ver quais padrões de tráfego existem na rede, e que tipo de requisitos eles exigem.

Até agora, falamos sobre topologias físicas, chegou a hora de falarmos sobre topologias lógicas! Comentei anteriormente que normalmente usamos a camada 2 na camada de acesso, e a camada 3 (roteamento) na camada de distribuição, mas há mais a ser contado nesta história. Vejamos alguns exemplos:

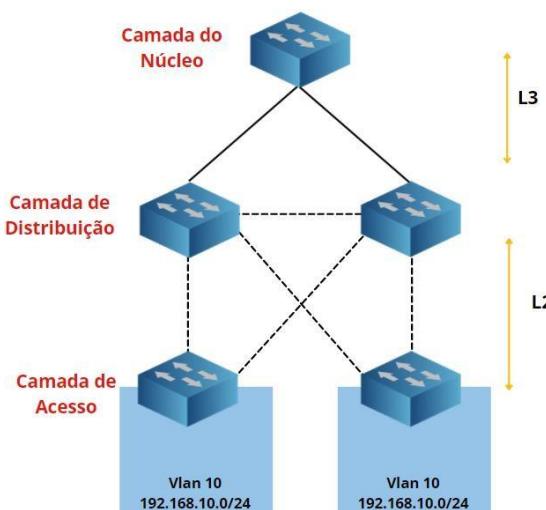


Diagrama com três camadas e apenas uma vlan.

Neste exemplo, temos um bloco de switch com dois switches na camada de distribuição e dois switches na camada de acesso. Temos uma VLAN configurada nos dois switches da camada de acesso.

Na conexão entre os switches da camada de distribuição e acesso, usamos a **camada 2 (layer 2, L2)**, já na conexão entre os switches da camada de distribuição e central usamos a **camada 3 (layer 3, L3)**. Os switches da camada de distribuição são usados como roteadores para os dispositivos finais que estão conectados nos switches da camada de acesso.

Como a VLAN 10 é usada em ambos os switches da camada de acesso, o link entre os dois switches da camada de distribuição **DEVE** ser um link da camada 2. Há duas razões para isso:

- Se um dos uplinks da camada de acesso para a camada de distribuição falhar, o tráfego da VLAN 10 poderá passar pela camada de acesso.
- Os switches na camada de distribuição usarão um protocolo específico para criar um endereço IP de gateway virtual (falaremos mais sobre esse assunto nos próximos capítulos), por isso, precisamos de conectividade na camada dois, e não há porque esse tráfego chegar até camada de acesso.

As linhas tracejadas indicam onde precisamos que a VLAN 10 esteja configurada. A configuração desse jeito irá funcionar, mas essa não é a forma ideal, é possível otimizar ainda mais essa configuração. Do jeito que está necessitaremos do spanning-tree (Spanning-tree é um protocolo utilizado para criar uma topologia livre de loops) para manter a rede sem loops, só que dessa forma todo esse bloco de switch se tornará um único ponto de falha.

Observe, com essa configuração, se um host começar a enviar muitos frames de broadcast, todo o bloco de switch será afetado. A solução ideal seria:

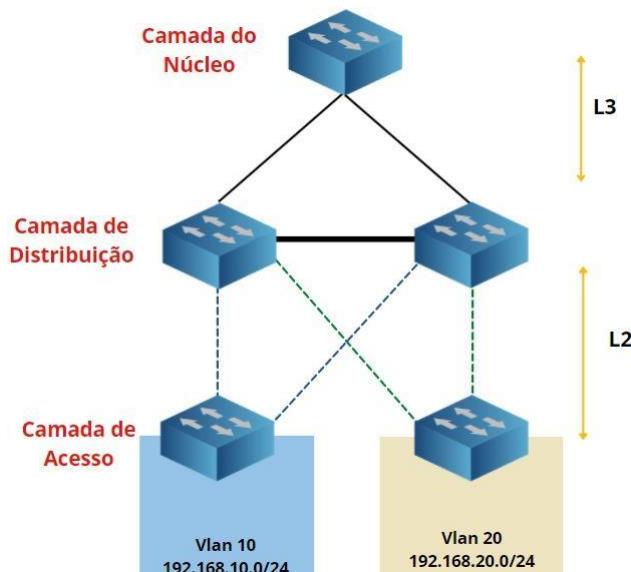


Diagrama com três camadas e apenas uma vlan.

Cada switch da camada de acesso agora pertence a uma única VLAN (vlan 10 e vlan 20), isso traz uma série de vantagens. Em primeiro lugar, nessa topologia não há redundância seja na VLAN 10 ou 20, então não dependemos mais do spanning-tree para criar uma topologia sem loop, o que torna todo o bloco mais estável.

Além disso, cada VLAN pode usar qualquer um dos uplinks, isso permite balanceamento de carga (load balance). O link entre os dois switches da camada de distribuição agora é um link da camada 3.

A terceira opção seria usar roteamento em todos os lugares:

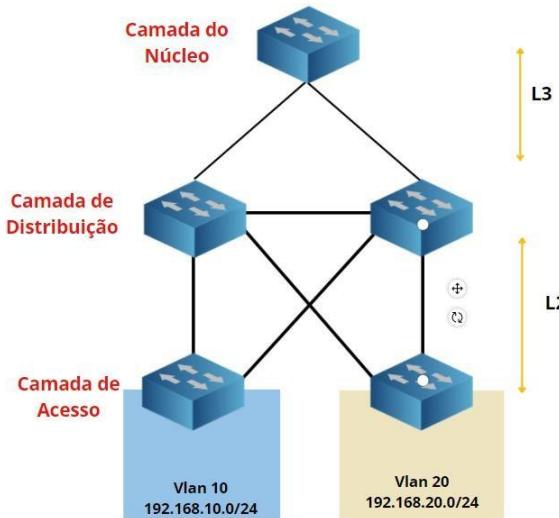


Diagrama com três camadas e duas VLANs na camada de acesso.

Podemos ‘descer’ com a camada 3 até a camada de acesso se os switches suportarem layer 03. A vantagem é que os protocolos de roteamento têm tempo de convergência menor, e com eles, podemos usar todos os links disponíveis, sendo que OSPF e EIGRP permitem balanceamento de carga.

Agora que vimos às diferentes camadas e opções que temos de designer para uma rede LAN, é hora de uma pequena revisão em alguns dos pontos-chaves:

- Cada camada deve ter pelos menos um par de switches para redundância.
- Cada switch deve ser conectado à camada superior com dois links para redundância.
- Não se conecta os switches da camada de acesso entre si.
- Não devemos estender VLANs acima da camada de distribuição. Da camada de distribuição até a camada central, usamos apenas roteamento.
- Sempre que possível, devemos evitar estender a mesma VLAN em mais de um switch na camada de acesso.
- A camada central é a mais simples, nela não fazemos nenhuma manipulação de pacotes, porém, ela precisa suportar um grande volume de tráfego, possuir disponibilidade e escalabilidade, afinal, ela agrupa todos os switches da camada de distribuição.

Hoje, os principais modelos utilizados para a camada de acesso são: **2960-X, 3650, 3850 e 4500-E**. Já os modelos mais utilizados nas camadas de distribuição e central são: **4500-X, 4500-E, 6807-XL**.

Após essa pequena introdução, podemos entrar sem medo nos assuntos cobrados na prova, perceba, que para entender os próximos conceitos essa explicação inicial é fundamental.

1.2 b - 3 tier - Arquitetura em 3 camadas

É chamado de **3 tier** pois possui **03 camadas**:

1. Camada de **acesso**: Onde estão conectados os usuários, impressoras, etc. Sempre usamos switches **layer 02** para essas conexões;
2. Camada de **distribuição**: Aqui é onde o roteamento acontece, nessa camada usamos switches **layer 03**
3. Camada **Core** – Nesta camada também temos roteamento, tanto em switches layer 3 quanto em roteadores, firewall, etc. A camada core geralmente é a camada de saída para Wan (Internet).

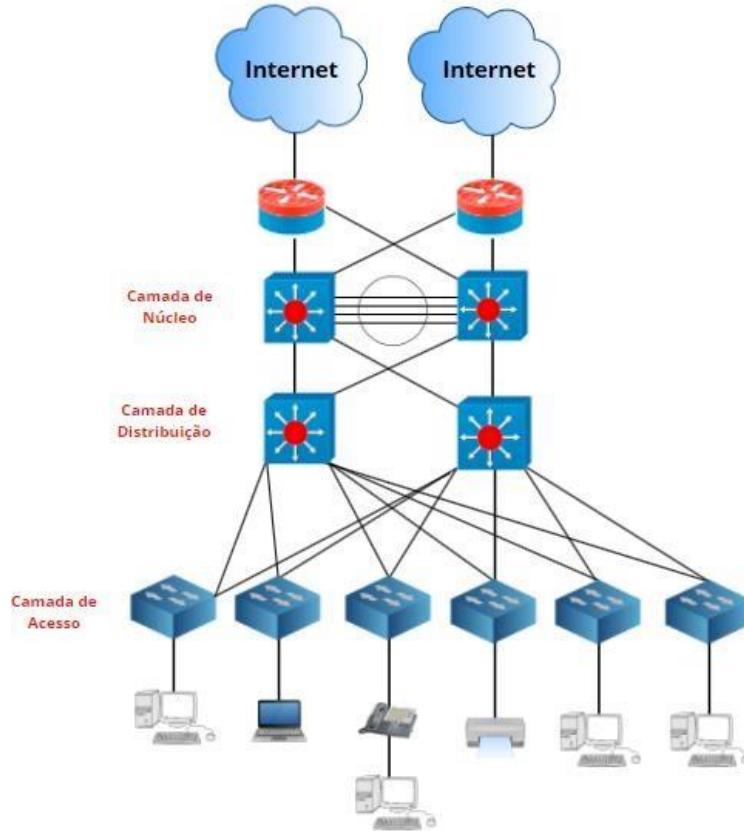


Diagrama 3 Tier.

1.2a - 2 tier - Arquitetura em 2 camadas

É chamado de **2 tier** pois possui **02** camadas:

A grande diferença entre a arquitetura com 02 e 03 camadas é que na arquitetura com 02 camadas, as camadas de **distribuição** e **core** da rede se tornam **uma só**, como no desenho abaixo.

Observe que qualquer switch layer 3 consegue executar o papel de **core\distribuição**, isso torna esse modelo bem mais barato, porém, perde-se um pouco de **desempenho e flexibilidade** na rede.

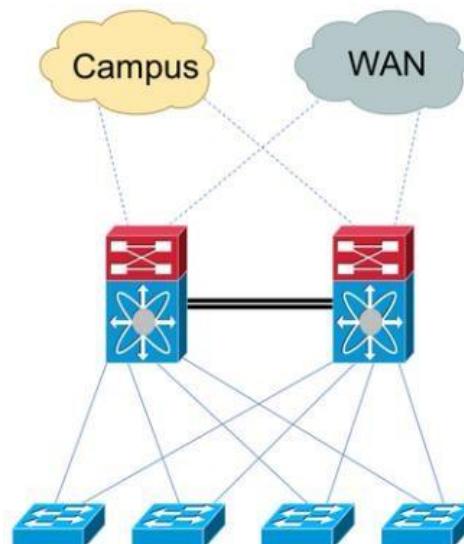
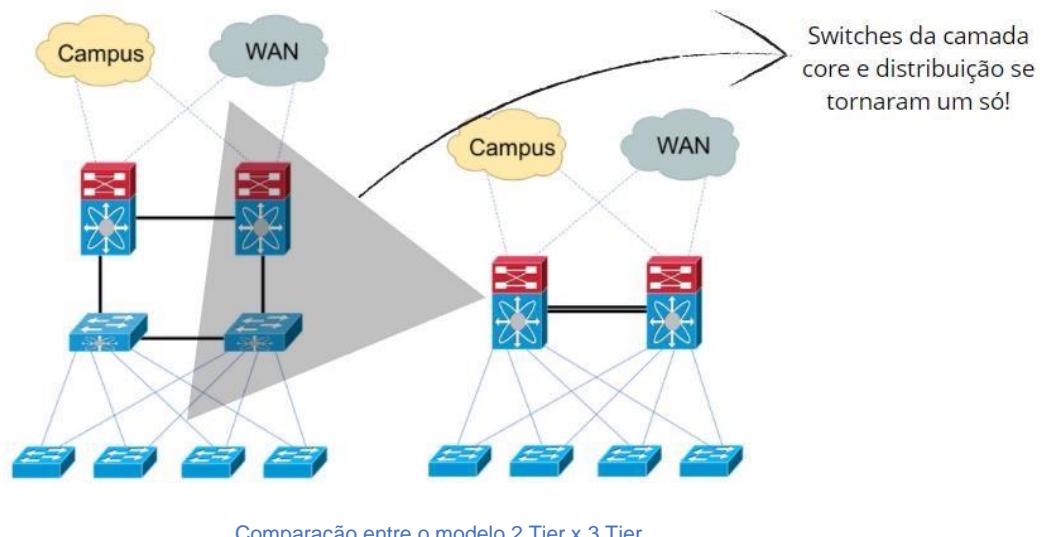


Diagrama 2 Tier.

2 tier x 3 tier - Comparação



1.2 c Spine-leaf

É uma forma de distribuir a topologia como se fosse uma **árvore**, afinal a tradução de **leaf** significa '**folha**'. Os switches **spines** são de **camada 03** e os switches **leaf** são de **camada 02**.

Os switches **leaves** são conectados aos equipamentos de borda, tais como servidores, máquinas virtuais. Essa é uma topologia simples, com apenas 02 camadas, utilizada principalmente em Data Centers

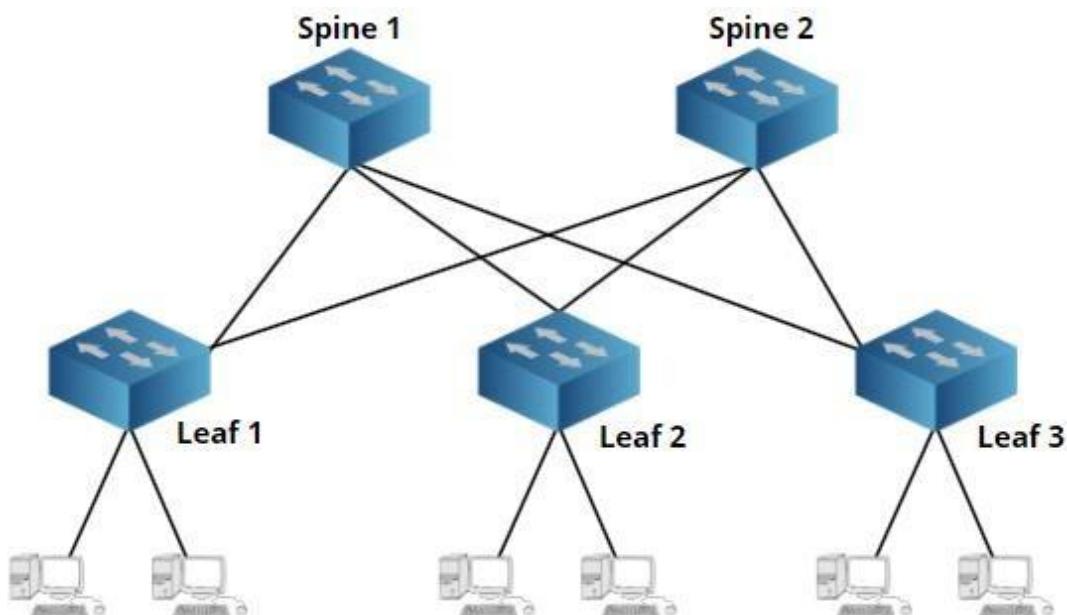


Diagrama Spine-leaf.

Observe que os switches da mesma camada não se conectam entre si:

- Cada switch Leaf conecta-se a todos os switches Spine;
- Cada switch Spine conecta-se a todos os switches Leaf;

As principais vantagens da arquitetura Spine-Leaf são:

- Projeto de rede tão simples quanto possível, ao mesmo tempo que atende aos requisitos de negócios;
- Balanceamento de carga determinístico e mais simples entre os dispositivos principais;
- Baixa latência;
- Escalabilidade simples por meio da adição de dispositivos ‘spine’



Diagrama spine-leaf em um data center.

1.2d – WAN

Wide Area Network conecta computadores dentro de uma grande área geográfica, abrangendo uma região específica, um país, um continente ou mesmo o mundo inteiro.

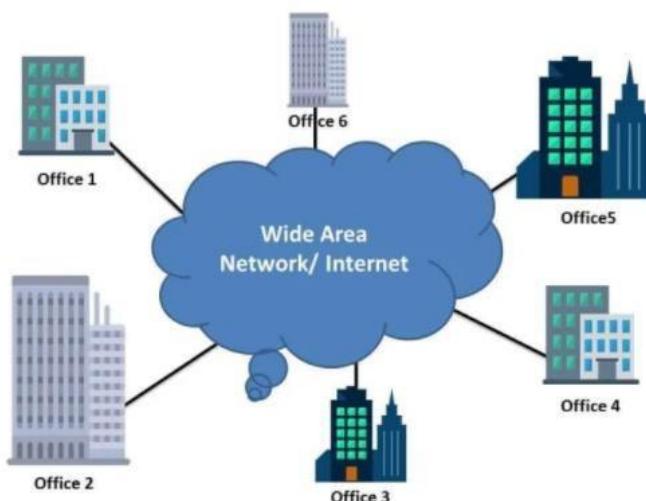
São geralmente redes pertencentes a ISPs (Operadoras de telefonia) que fornecem uma maneira útil de compartilhar recursos entre filias de empresas, seja por transmissão de longa distância de dados, voz e imagem.

O melhor exemplo de rede de longa distância é a Internet, que conecta muitas LANs e MANs menores por meio de provedores de serviços.

As operadoras podem oferecer conectividade a rede WAN através de diversos tipos de conexão como:

-MPLS, SD-WAN, Frame Relay, VPN, Link Dedicado.

Todas essas tecnologias estão dentro da WAN, e essas conexões estão ativas 24/7 durante 365 dias.



Representação de uma rede WAN\Internet.

1.2.e - Small office/home office (SOHO)

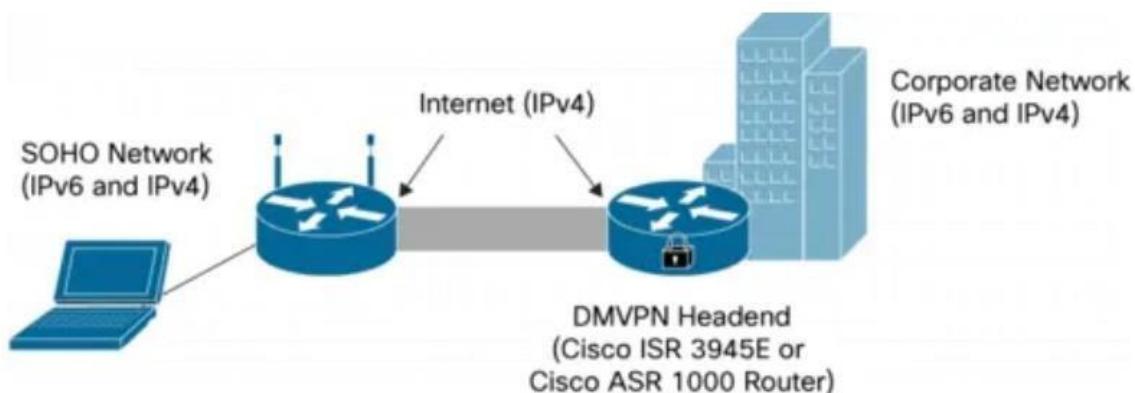
Não há muito o que falar nesse tópico, pois são utilizados basicamente os mesmos equipamentos de uma LAN normal, porém aqui estamos nos referindo a pequenos escritórios e home office.

Os roteadores Cisco para SOHO são equipamentos com preços acessíveis que por muitas vezes vem com firewall, VPN e Wifi integrados como este modelo abaixo:



Roteador Cisco C88EA-K9

Eis uma pequena topologia representando uma SOHO com conexão a um escritório central:



Representação de uma rede SOHO.

1.2.f On-premises and cloud

Aqui teremos apenas uma visão geral do que é Cloud, nos capítulos ulteriores teremos uma visão mais completa.

On-premises- Quando temos usuários que acessam aplicativos e servidores que estão **instalados/armazenados localmente** na nossa infraestrutura, ficando sob nossa responsabilidade o gerenciamento, manutenção, recuperação de desastres, etc.

Cloud - Quando os aplicativos e servidores estão **instalados/armazenados na nuvem**, ficando sob responsabilidade da empresa contratada o gerenciamento e a manutenção.

Exemplo: Podemos contratar Amazon AWS, Microsoft Azure ou Google Cloud.

Dentro da Cloud é possível segmentar as redes, criar firewalls, desktop remotos, servidores, absolutamente tudo é customizável. As principais vantagens da cloud são:

- Escalabilidade;
- Segurança;
- Preço.

1.3 Compare physical interface and cabling types

É hora de aprendermos o que é ethernet e como ela funciona.

Introdução a Ethernet

Entraremos agora na parte de conexões físicas: Conectores, os diversos tipos de cabos, as diferenças entre cada um deles e o funcionamento. Antes de entrarmos nos tópicos da prova é necessário que entendamos o conceito de '**Ethernet**':

Ethernet não é um protocolo único, mas uma coleção inteira de padrões diferentes. Esses padrões vêm do **IEEE** e todos eles começam com 802.3 no nome. O padrão Ethernet é muito antigo, para você ter uma ideia, o primeiro memorando sobre Ethernet foi escrito por *Bob Metcalfe* no hoje longínquo ano de 1973.

Apesar de sua idade, a Ethernet é ainda hoje a tecnologia dominante nas LANs, englobando uma gama gigantesca de padrões diferentes, indo de velocidades de 10 Mbps (megabits por segundo) até 100 Gbps (gigabits por segundo).

Abaixo, um quadro com uma visão geral de alguns dos padrões Ethernets mais populares:

Largura de banda	Nome	Nome informal	IEEE	Tipo de cabo
10 Mbps	Ethernet	10BASE-T	802.3	UTP 100m
100 Mbps	Fast Ethernet	100BASE-T	802.3u	UTP 100m
1000 Mbps	Gigabit Ethernet	1000BASE-LX	802.3z	Fiber 5000m
1000 Mbps	Gigabit Ethernet	1000BASE-T	802.3ab	UTP 100m
10 Gbps	10 Gigabit Ethernet	10GBASE-T	802.3an	UTP 100m

Na camada física, existem diferentes opções de cabos com velocidades distintas. Uma das vantagens do Ethernet, entretanto, é que ela usa o mesmo padrão na camada de enlace de dados. Com isso, podemos misturar diferentes padrões Ethernet em uma rede. Eis um exemplo:

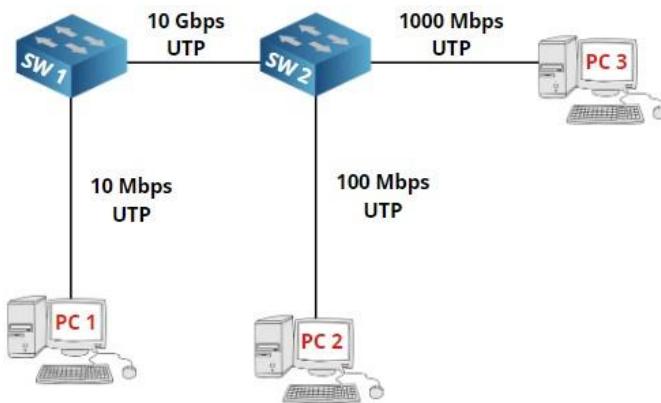


Diagrama de uma LAN com diferentes padrões Ethernet

Acima, vemos três hosts conectados a dois switches usando padrões Ethernet diferentes. A conexão entre os switches é realizada através de uma conexão a fibra de 10 Gbps. Essa rede funcionará normalmente encaminhando frames Ethernet, mesmo havendo uma mistura de vários padrões diferentes.

Data Link

Physical

Duas primeiras camadas do Modelo OSI

Ethernet compreende a **camada física** e a **camada de enlace de dados**. Vamos analisá-las no contexto exigido na prova:

Camada física:

Existem dois cabos que podemos usar para Ethernet:

- **UTP**, que é o par trançado não blindado ou **Unshielded Twisted Pair**;
- **Fibra ótica**

Os próximos tópicos serão voltados para esses dois tipos de cabo.

1.3.a Single-modefiber, multimodefiber, copper

Como dito anteriormente, Ethernet é uma arquitetura de interconexão para redes locais, baseada no envio de pacotes. Através dela são definidos o cabeamento e os sinais elétricos que são enviados para a subcamada de controle de acesso ao meio (**Media Access Control - MAC**).

Mas, para que o sinal seja enviado de forma eficiente, com estabilidade e boa capacidade de transferência, é necessário utilização de um cabo adequado. Os cabos utilizados para conectar os dispositivos a rede em nossas casas ou em empresas são chamados de **cabos ethernet** (os populares cabos de rede), e há diferentes categorias de cabos, embora isso não seja visível para o leigo.

A cada geração que surge, há um aumento na velocidade de transmissão de dados e no cancelamento de ruído eletromagnético. Embora exista categorias anteriores ao cabo ethernet CAT5E, só descreverei as categorias a partir deste, pois as outras classes já não são mais utilizadas e não serão cobradas no CCNA.

1.3.a Copper

Cooper - Cabos Ethernet – CAT 5e

O cabo de categoria CAT5E surgiu no ano de 2001, ele é um aprimoramento da categoria CAT5. Com ele conseguiu-se uma interferência menor entre os fios, obtendo um sinal de melhor qualidade.

Essa categoria de cabo é a mais comum e a mais utilizada nas instalações, devido ao seu baixo custo e maleabilidade.

Embora, ele possua a teórica capacidade de chegar a uma velocidade de até **1000Mbps** (cat 5 chega a 100Mbps), é algo muito difícil de acontecer devido à sua estrutura vulnerável a interferências eletromagnéticas. Ele opera na frequência de 100MHz em uma distância máxima de **100 metros**,

Ele possui **4 pares de fios trançados sem blindagem** alguma e nenhum distanciamento entre eles.



CAT5e.

Cooper - Cabos Ethernet – CAT 6

O cabo de ethernet CAT6 consegue suportar até **1Gbps** a uma frequência de 250Mhz a uma distância máxima de **100m**.

Uma diferença entre esses cabos é que os cabos CAT5E são trançados na razão de 1 volta e meia a 2 voltas por cm, já os de categoria 6 são enrolados com mais força, algo em torno de 2 ou mais voltas por cm (a quantidade de torções varia de acordo com o fabricante do cabo).

Os cabos CAT6, possuem uma grossura maior em comparação aos CAT5E, pois além da presença dos **4 pares de fios trançados**, há uma estrutura de plástico em forma de "X" que divide os pares de fios.



CAT6

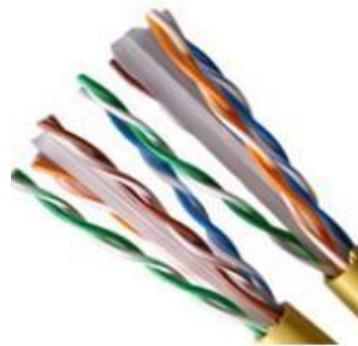
Esta estrutura divisora dos cabos é responsável por dar uma maior durabilidade, pois evita que os fios se dobrarem muito, além de diminuir o nível de interferência entre eles.

Porém, isso nem tudo são flores, devido a essas mesmas características o cabo tem uma certa perda de flexibilidade quando comparado aos cabos CAT5E. A estrutura diferenciada do cabo CAT6 colabora para uma transmissão de dados mais estável, principalmente durante a utilização de serviços de streaming como o Netflix, jogos online e chamadas de áudio e vídeo como as utilizadas no Skype.

Cooper - Cabos Ethernet – CAT 6A

Os cabos de categoria 6A são uma atualização da CAT6, eles são capazes de atingir uma taxa de transmissão de 10Gbps a 500Mhz.

Os cabos dessa categoria possuem um revestimento mais robusto tanto externamente quanto internamente. Esse revestimento mais robusto proporciona uma redução significativa no ruído do sinal. Entretanto, isso acaba tornando o cabo CAT6A sensivelmente mais grosso que o CAT6, diminuindo a sua flexibilidade e dificultando a instalação, dependendo do local pode ser inviável a utilização do CAT6A.



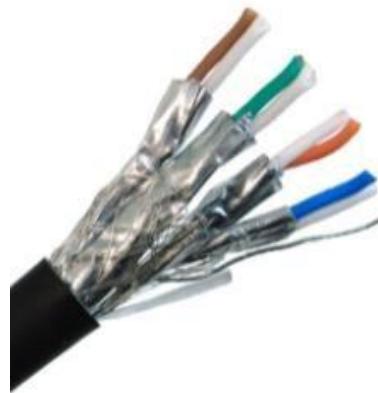
CAT6A

Cooper - Cabos Ethernet – CAT 7

A categoria 7 teoricamente suporta transferências de 10Gbps a 600Mhz a uma distância máxima de 100 metros, porém testes comprovaram que a velocidade pode chegar a 40Gbps quando testado com um comprimento de 50m, e surpreenda-se com 15m ele consegue proporcionar a incrível velocidade de 100Gbps. Isso ocorre devido a atenuação extrema dos ruídos eletromagnéticos, pois o cabo possui blindagem interna de alumínio e cada fio de cobre também possui blindagem de alumínio.

Como a blindagem precisa ser aterrada, todos os cabos desta categoria possuem conectores de metal. Ele é muito mais grosso que os de categoria 6A, além de possuir maior rigidez, por isso, dependendo do local, não será possível instalar o cabo, por conta de seu diâmetro que impede uma torção muito grande.

O CAT7 é o tipo de cabeamento mais indicado caso os cabos tenham que passar próximo a fios condutores de eletricidade (15cm), pois pela sua constituição, o cabo dificilmente será afetado. Estando livre de interferências será obtido um sinal de internet mais consistente, sem oscilações e com boa velocidade.



CAT7

Cooper - Cabos Ethernet – Conector RJ45

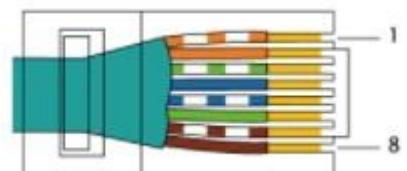
É hora de estudarmos os conectores que são plugados em cada ponta dos cabos, esse conector é conhecido como **RJ45**

Principais características:

- O conector RJ45 possui **8 'encaixes' (slots)**, um para cada fio de cobre. Chamamos esses espaços de '**pin**' e contamos da esquerda para a direita olhando da parte inferior do conector.
- Antigamente, era necessários prestar bastante atenção na hora de montar um cabo ethernet, pois a sequência dos fios fazia toda a diferença. A ordem que colocamos os fios no conector define se teremos um cabo **direto (Straight Through Cable)** ou **cabo cruzado (Crossover Cable)**. Cada tipo de cabo é indicado para um tipo de conexão específica.
- Hoje em dia não faz muita diferença o tipo de cabo, pois os switches conseguem detectar qual dispositivo está plugado na outra ponta, e utilizar somente os fios corretos (**tecnologia auto-mdix**)



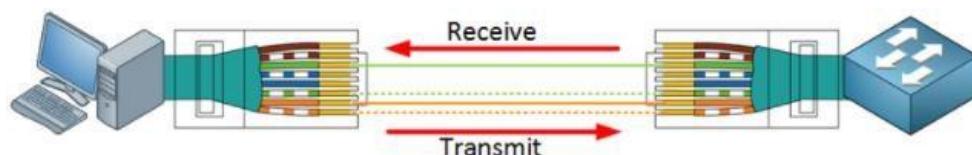
Conector RJ45 crimpado a um cabo ethernet



Vista superior de um conector RJ45 com esquema de cores

Cooper - Cabos Ethernet – Cabo direto (Straight Through)

Os padrões 10BASE-T e 100BASE-T usam apenas dois pares de fios, um para transmissão e outro para recepção. Repare no exemplo abaixo, onde temos um computador conectado a um switch:



Esquema de cores de um cabo direto, o cabo indicado para conexão computador x switch.

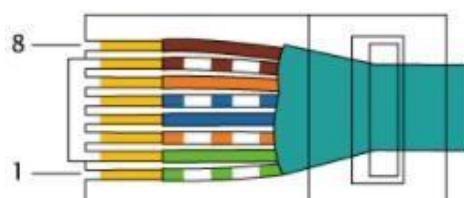
No lado esquerdo, o computador usa os fios laranja/laranja-branco para transmitir dados (pino 1 e 2), e os fios verde/verde-branco (pino 3 e 6) são usados para receber dados, na outra extremidade, o switch recebe os dados nos fios laranja/laranja-branco (pinos 1 e 2) e transmite nos fios verde/verde-branco (pinos 3 e 6).

Por isso são chamados de cabo direto, os fios em ambas as extremidades são conectados diretamente na mesma ordem.

Copper- Cabos Ethernet - Cabo cruzado (Crossover)

E se quisermos conectar dois switches um ao outro? Se ambos transmitirem nos pinos 3 e 6, teremos uma colisão no fio. Para garantir o envio/recebimento nos pinos corretos, para esse tipo de conexão usamos um tipo diferente de cabo, denominado cabo cruzado.

O cabo UTP é o mesmo, mas em uma extremidade do cabo temos os fios em uma ordem diferente no conector RJ45:



Padrão EIA/TIA-568A

Observe a sequência de cores que foi utilizada na montagem do cabo para a conexão entre dois switches:



Switches conectados utilizando cabo cruzado

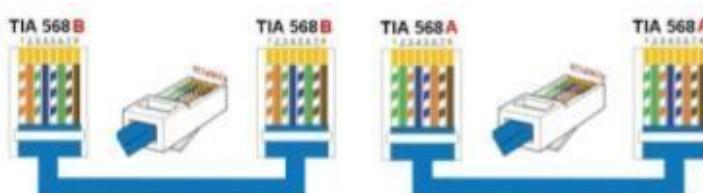
Observe que os cabos estão cruzados dessa forma ao se conectar ao RJ45:

Switch da esquerda	Switch da direita
Pin 1	Pin 3
Pin 2	Pin 6
Pin 3	Pin 1
Pin 6	Pin 2

Para os switches modernos não faz diferença se o cabo é direto ou cruzado, eles possuem uma tecnologia chamada auto-mdix para descobrir o tipo de cabo que está sendo usado, e assim definem automaticamente os fios corretos que serão utilizados.

Copper- Cabos Ethernet - Cabo cruzado x Cabo Direto

Os cabos diretos podem tanto seguir a norma **TIA/EIA T568A** como a norma **TIA/EIA T568B** (geralmente eles são montados seguindo essa última norma).



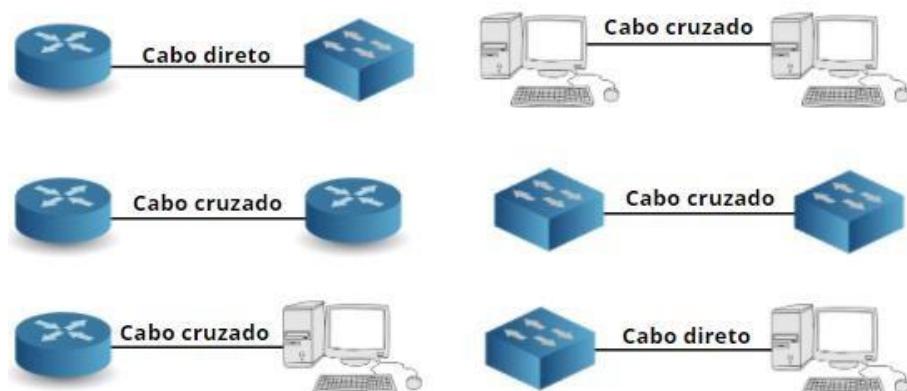
Cabo direto (Straight-through)

Os cabos cruzados são finalizados com uma das pontas com o esquema **T568A** e a outra ponta com esquema **T568B** (ou seja, são trocados os fios 1 e 3 e o 2 e 6), em que cada um deles terá dois pares para transmissão (TX) e dois pares para recepção (RX).



Copper- Cabos Ethernet – Seleção do cabo correto

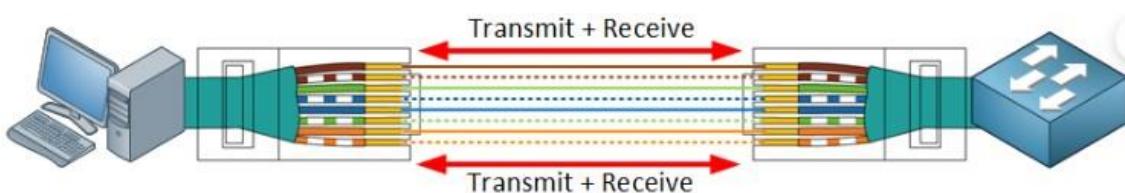
Resumindo: **Cabo direto serve para dispositivos diferentes se comunicarem** (com exceção de computador e roteador que usam cabo crossover), e **cabo cruzado serve para dispositivos iguais se comunicarem**.



Esquema de cabos para conexão entre diferentes dispositivos.

Cabo 1000BASE-T (Gigabit Cabling)

Até aqui todos os exemplos que vimos utilizavam dois pares de fios, sendo um para transmissão e outro para recepção. Porém no caso do Gigabit Ethernet todos os 4 pares de fios são utilizados. Em vez de usar pares de fios diferentes para transmissão/recepção, ele consegue transmitir e receber simultaneamente em cada par de fios. O layout da conexão dos fios no conector RJ45 é o mesmo, porém, são utilizados fios adicionais:



Esquema de cabos para diferentes dispositivos.

Switch da esquerda	Switch da direita
Pin 4	Pin 7
Pin 5	Pin 8
Pin 7	Pin 4
Pin 8	Pin 5

Single-mode fiber, multi mode fiber, copper

Antes de entrarmos nas diferenças entre as fibras single-mode e multimode, é necessário fazer uma pequena introdução sobre fibra óptica:

A transmissão de dados por fibra óptica revolucionou o mercado ao possibilitar a conexão em distâncias muito superiores às que os cabos de cobre (cabo de rede ou coaxial) permitiam. Além da distância, existem outros benefícios:

- Fibra Óptica não sofre interferência eletromagnética;
- Velocidade: Não existe limitação em relação à velocidade, por exemplo, atualmente já existem soluções capazes de transmitir mais de 400Gbps por fibra óptica;
- Tamanho reduzido: fibra óptica ocupa menos espaço que cabos de cobre;
- Economia e Durabilidade: Fibra óptica é produzida a base de sílica, um material resistente a vários tipos de intempéries, a um custo relativamente baixo. Além do mais, uma rede de fibra necessita de menos equipamentos para funcionar, requer menos manutenção e tem maior vida útil.



Foto de um cordão de fibra óptica.

A fibra óptica é um fio feito de vidro ou plástico com espessura menor que um fio de cabo. O conceito de funcionamento da fibra é que a luz emitida em uma ponta se reflete entre as paredes desse fio, acompanhando toda a sua trajetória até a outra ponta do cabo, conforme a imagem abaixo ilustra:

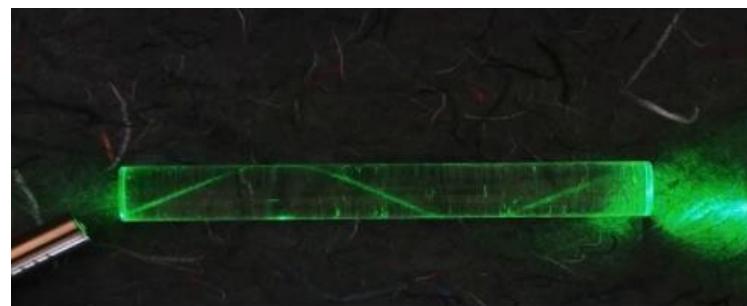


Imagen retirada da Internet

Single-mode ou Monomodo e Multimodo ou Multimodo, possuem características e indicações diferentes, veja as principais diferenças:

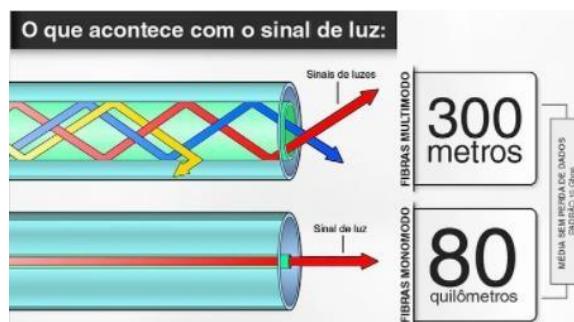


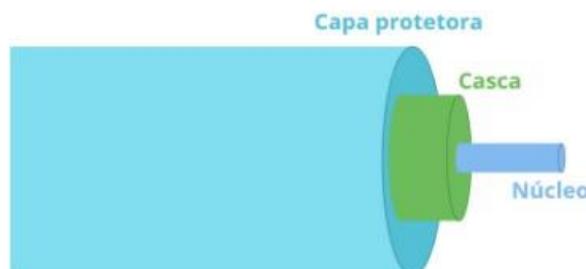
Imagen retirada da Internet

As fibras ópticas são basicamente compostas por **3 elementos**:

1 -Núcleo: Por onde o sinal trafega;

2 - Casca: Fica em volta do núcleo, ajuda a evitar a perda de luz e auxilia na reflexão;

3 -Revestimento ou "Capa": Material de proteção entre o ambiente externo e a casca.



Corte transversal de uma fibra ótica

A maioria dos switches possuem dezenas de portas para cabo UTP e alguns poucos slots para adaptadores para fibra. Nesses slots, é possível inserir um SFP para decidir se queremos que essa porta seja mais uma porta UTP ou uma porta de fibra.



Switch com portas UTP e SFP

Um SFP, ou Small Form Factor Pluggable, é um transceptor compacto e hot-swappable projetado para suportar Ethernet de 100/1000Mbps, Fibre Channel, SONET, entre outros padrões de comunicação. Portas SFP são encontradas em uma gama de dispositivos, de switches Ethernet a roteadores, placas de rede e firewalls.



Módulo Gbic Cisco SFP 1000BASE-T



Switch gerenciado Cisco SG300-10SFP de 10 portas Gigabit SFP

Multimodefiber ou MM

Esse foi o primeiro tipo de fibra a aparecer no mercado, chamado de multimodo ou pela sigla " MM" por permitir diversos sinais de luz na mesma fibra.

A principal característica que a difere de uma fibra monomodo é a espessura maior: 50/125 ou 62.5/125 mícrons e a distância máxima de transmissão ser menor.

O fato da fibra multimodo possuir uma espessura maior, impacta diretamente na quantidade de reflexões que o sinal de luz irá realizar no corpo da fibra, e consequentemente na perda que terá, limitando a distância máxima entre 300 metros a 2km.



Multimode fiber ou MM

Outra característica pouco comentada é o preço. O principal elemento gerador de custo em um cabo de fibra é o núcleo, por possuir um núcleo maior, o preço da fibra multimodo é mais caro que da monomodo.

Características da fibra multimodo (MM):

- Espessura do núcleo 62.5/125 µm
- Distância máxima de 300 metros até **2km, dependendo** do comprimento da onda e qualidade da fibra;
- Custo por quilômetro (km) mais alto;
- Instalação mais barata, o que inclui conectores;
- Recomendada para enlaces internos ou até cascamenteamento em um mesmo rack;

Ex: Lan, San, Data Center, CO.

Single-modefiber ou SM

Surgiu anos depois como uma solução à limitação da distância máxima que a fibra multimodo conseguia atingir. A fibra monomodo é conhecida pela sigla "SM", por conta do seu nome em inglês, ‘single mode’.

Caracterizada por um núcleo menor que da fibra multimodo, possui apenas 9 mícros de espessura. Essa característica torna o sinal muito mais linear e com menor perda durante o caminho. Por isso, consegue transmitir a distâncias superiores a 300km.



Single-modefiber ou SM

O núcleo menor diminui e muito o custo da fibra óptica monomodo, tornando o custo por km muito inferior ao da fibra multimodo.

Características da fibra monomodo (SM):

- Espessura do núcleo 9/125 μm ;
- Distâncias máxima acima de 150km;
- Custo por quilômetro (km) mais baixo;
- Velocidade superior ao multimodo;
- Recomendada tanto para enlaces internos quanto externos:

EX: WAN, MAN, Campus, etc.

1.3.b Connections (Ethernet shared media and point-to-point)

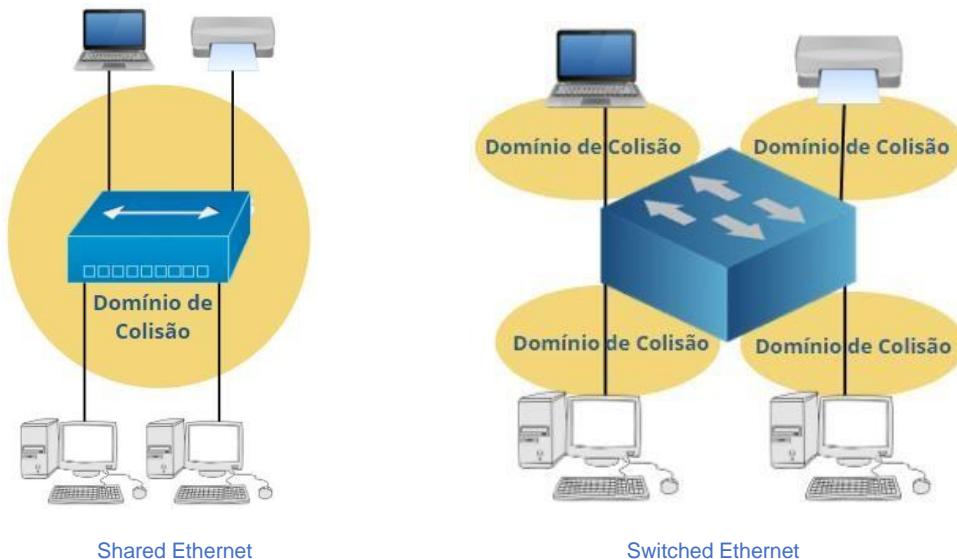
Já sabemos que Ethernet não é um padrão único, mas uma coleção inteira de padrões que opera nas duas camadas inferiores do modelo OSI: **Camada Física e Enlace**.

Na camada física, ela pode operar com diferentes opções de cabos e velocidade. Porém, é na camada de enlace de dados que está uma das maiores vantagens da Ethernet, ela usa o mesmo padrão independente do que foi utilizado na camada abaixo.

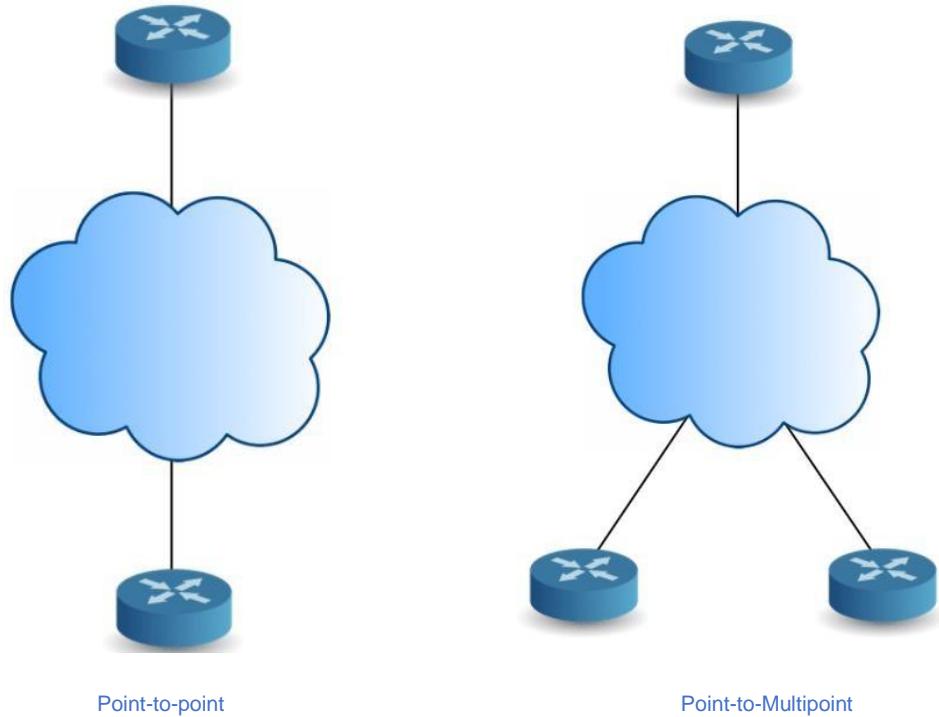
Shared media and point-to-point

Antigamente as conexões Ethernets eram compartilhadas, ou seja, todos os dispositivos conectados à rede compartilhavam o mesmo domínio de colisão (veremos mais sobre domínio de colisão a seguir), o que evidentemente era algo negativo, afinal, quando vários dispositivos estão no mesmo domínio de colisão, aumenta a probabilidade dos dados que estão trafegando nessa rede colidirem. Isso causa degradação e perda desempenho. Podemos concluir que quanto mais domínios de colisão tivermos na rede, melhor será o desempenho.

Atualmente, cada dispositivo final se conecta a uma interface do switch. Nos switches cada interface de rede é um domínio de colisão. Literalmente, cada dispositivo final tem seu próprio domínio de colisão quando utilizamos switches.



Conexões ponto a ponto (point-to-point) ocorrem quando apenas dois dispositivos estão logicamente ou fisicamente conectados.



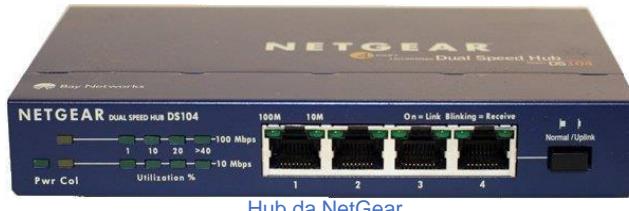
Point-to-point

Point-to-Multipoint

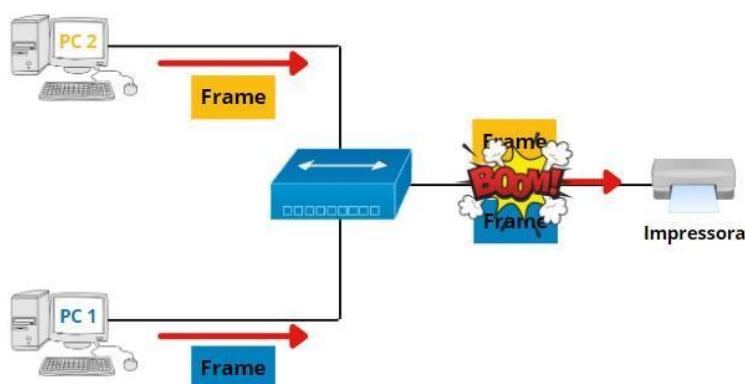
Agora que entendemos 02 conceitos importantes para a prova, vamos estudar o funcionamento do protocolo Ethernet e os sub protocolos que ele utiliza.

CSMA/CD – Duplex

Para entender o duplex, temos que voltar no tempo, mas precisamente para o ano de 1990, onde não havia switches, apenas um dispositivo chamado hub. Por fora, um hub se parece com um switch, com várias portas RJ45 para conectarmos computadores e outros dispositivos finais.



O Hub é um dispositivo ‘burro’, ou seja, quando ele recebe sinal elétrico em uma porta, ele repete esse sinal para todas as outras portas. Esse comportamento causa muitas colisões na rede. Nada melhor que um exemplo ilustrado para explicar:



Exemplo de colisão entre frames.

Note que os computadores PC1 e PC2 enviam um quadro Ethernet ao mesmo tempo. O hub recebe esse quadro, e encaminha para interface que está conectado a impressora. Como os frames foram encaminhados ao mesmo ocorre a colisão, e os dois quadros são perdidos.

A solução encontrada para contornar esse problema foi o **half duplex**.

Half duplex significa que não é possível enviar e receber ao mesmo tempo. Quando um computador está transmitindo, todos os outros dispositivos precisam esperar que ele termine. Quando ninguém está transmitindo, os computadores podem enviar frames.

Isso não significa que não haverá mais colisões. Quando dois computadores decidem que “a linha está livre” e começam a transmitir ao mesmo tempo ocorrerá uma colisão. Para resolver esse problema, criou-se um protocolo chamado **CSMA/CD**:

- **CSMA** - Carrier Sense Multiple Access, ou Acesso múltiplo com detecção de portadora
- **CS** = Carrier Sense (Sentido da portadora)
- **MA** = Multi Access (Acesso múltiplo)
- **CD**= Collision Detection (Detecção de Colisão)

Vamos destrinchar em partes:

Sentido de portadora significa que ficamos “ouvindo” o cabo para saber se algo está acontecendo, ou seja, se outro computador está enviando dados naquele momento.

Acesso múltiplo significa que todos podem acessar o meio físico, desde que não haja nenhum outro computador enviando frames no momento.

No caso de dois ou mais computadores enviarem frames ao mesmo tempo, ocorrerá uma colisão. A função do **CSMA|CD** é justamente detectar isso, o algoritmo funciona da seguinte maneira:

1. Os dois computadores envolvidos na colisão começarão a obstruir o meio físico; isso irá garantir que ninguém mais conseguirá transmitir naquele momento.
2. Cada computador envolvido na colisão iniciará uma contagem regressiva a partir de um número aleatório.
3. Quando a contagem regressiva terminar, eles retransmitirão os dados perdidos na colisão.

Como o número inicial da contagem de tempo é aleatória, os dois computadores terão um cronômetro com contagem regressiva diferente, e um deles enviará seus dados antes do outro. O objetivo de congestionar o meio físico é garantir que nenhum outro computador terá a chance de enviar dados antes dele.

Hoje em dia é difícil encontrar um hub, todos foram substituídos por switches. É bom frisar que switches são bem mais caros que hubs.

Os switches são dispositivos inteligentes, eles leem o quadro Ethernet e encaminham o frame apenas para o dispositivo que o quadro está endereçado. Quando há dois frames com destino a uma mesma porta enviado no mesmo momento, o switch consegue enfileirar esses frames evitando colisões.

Switches operam em full duplex, o que significa que todos podem enviar e transmitir dados ao mesmo tempo. Como não há colisões o protocolo **CSMA/CD** vem desabilitado por padrão.

Ainda encontramos rede operando em half duplex, como é o caso das redes sem fio. Um ponto de acesso sem fio é semelhante a um hub, todos os dispositivos transmitem e recebem na mesma frequência, dessa forma é normal que ocorram colisões. Diferente das redes cabeadas, as redes sem fio não utilizam o CSMA/CD, mas um protocolo chamado CSMA/CA (Collision Avoidance ou prevenção de colisão).

Ethernet - Data Link Layer

Um dos fatos mais surpreendentes sobre a Ethernet é que, embora tenhamos padrões diferentes, todos eles usam um frame Ethernet comum. Este frame não mudou muito desde que os padrões Ethernet originais foram instituídos nos anos 70.

Abaixo, temos os componentes de um frame Ethernet:



Frame Ethernet

Vamos aprender o significado de cada campo desse:

- **Preâmbulo:** Possui um padrão de 7 bytes de uns e zeros, é usado para sincronização.
- **SFD:** o “delimitador de quadro inicial” marca o final do preâmbulo e diz ao receptor que os próximos campos serão o quadro Ethernet real, começando com o campo de destino.
- **Destino:** este é o endereço MAC de destino do receptor.
- **Fonte:** o endereço MAC de origem do dispositivo que enviou o quadro.
- **Tipo:** Aqui é informado o que é transportado dentro do quadro Ethernet. Pode ser um pacote IPv4, pacote IPv6 ou qualquer outro.
- **Dados:** carrega os dados reais que estamos tentando transmitir, por exemplo, um pacote IPv4.
- **FCS:** É a sequência de verificação do quadro, ele ajuda o receptor a descobrir se o quadro está integral ou se foi corrompido.

Os campos marcados em verde são conhecidos como cabeçalho Ethernet (Ethernet header).

Ethernet - MAC Address

Já falamos um pouco sobre endereço MAC, mas faz-se necessário uma nova abordagem, dessa vez utilizando uma visão mais aprofundada do protocolo Ethernet.

Os endereços Ethernet são chamados de endereços **MAC** (Media Access Control), cada dispositivo de rede possui um endereço MAC exclusivo. Quando enviamos um quadro Ethernet, adicionamos nosso próprio endereço MAC como origem e o endereço MAC do receptor como destino.

Podemos dizer sem medo de errar que o endereço MAC é o endereço da placa de rede:



Placa Rede Gigabit Pci-e Rj45 Ethernet 10/100/1000mbps Dp-02

Abaixo temos um exemplo de endereço MAC:

OUI 24	Vendor Assigned 24
-----------	-----------------------

Exemplo de endereço MAC

Vamos entender os dois campos que formam o endereço MAC:

O endereço MAC tem **48 bits** ou **6 bytes** no total, utilizamos a notação hexadecimal. Por exemplo:

- **0000.0c12.3456**

Acima, temos quatro caracteres hexadecimais, separados por um ponto, porém existem duas outras opções de formatação:

- **00: 00: 0c: 12: 34: 56**
- **00-00-0c-12-34-56**

Atente-se que todos os três endereços acima se referem ao mesmo endereço MAC. Os dispositivos Cisco geralmente usam a primeira opção, os computadores Windows usam a terceira opção.

Normalmente, um endereço MAC se refere a um único dispositivo na rede, dessa forma um endereço MAC também é um endereço **unicast**. Porém, existe endereço MAC para tráfego de **broadcast** (o que significa que todos os dispositivos na rede receberão o quadro) e para tráfego **multicast** (apenas um grupo específico de dispositivos receberá o frame).

Os endereços MAC são únicos, caso contrário um mesmo quadro Ethernet poderia chegar em dois ou mais hosts receptores. Para tornar os endereços MAC únicos, cada fabricante de equipamento de redes recebe da IEEE um código exclusivo de **24 bits** denominado **OUI** (Organizationally Unique Identifier).

Por exemplo, todos os endereços MAC que começam com **0000.0c** são propriedade da Cisco.

Os 24 bits restantes do endereço MAC são atribuídos pelo fabricante (Vendor), dessa forma, cada placa de rede acaba tendo um endereço MAC exclusivo. O endereço que o fabricante atribui também é chamado de **BIA** (endereço gravado).

Quando um dispositivo envia uma mensagem broadcast, ele está enviando um quadro com endereço MAC de destino: **FFFF.FFFF. FFFF**.

Por serem extremamente importantes para o CCNA, resolvi detalhar melhor 02 campos do quadro Ethernet:

Ethernet -Type Field

O campo Type (Type Field) no quadro Ethernet informa o tipo de dados que o quadro está transportando. Geralmente são pacotes IPv4 ou IPv6.

Por exemplo, quando o remetente deseja enviar um pacote IPv4, é inserido no campo Type um código em hexadecimal, no caso do IPv4: **0800**. As opções mais comuns são essas abaixo:

- 0800: IPv4
- 86DD: IPv6
- 0806: ARP (Protocolo de Resolução de Endereço)

Ethernet -Error Detection

O campo FCS (Frame Check Sequence) é usado para verificar se um quadro Ethernet está integral ou corrompido. Os quadros podem ser corrompidos devido a placas de redes (NICs - Network Interface Card) com defeito, interferência elétrica no meio físico, etc.

O dispositivo remetente utiliza uma fórmula (que não nos interessa no momento) para criar um determinado valor. Este valor é adicionado ao campo FCS. O dispositivo receptor usará a mesma fórmula para calcular o valor do FCS quando receber o quadro. Se o valor for o mesmo, significa que o quadro está integral. Caso esse valor seja diferente, há uma clara indicação que o quadro foi corrompido durante a transmissão. Quadros corrompidos são descartados.

Importante, na camada 02 não há recuperação de erros, essa recuperação é realizada em camadas superiores, normalmente através do protocolo TCP na camada de transporte.

1.3.c Concepts of PoE

Power over Ethernet (PoE) é uma tecnologia utilizada para energizar dispositivos por meio de cabos Ethernet, simplificando, é a utilização dos cabos de rede para alimentar dispositivos.

PoE foi inventado juntamente com os primeiros telefones VoIP. A necessidade do PoE se deu devido ao funcionamento dos antigos telefones analógicos, que são alimentados diretamente pelos mesmos fios de cobre usados para chamadas de voz. Os telefones VoIP usam cabos Ethernet, só que estes não transmitiam energia, portanto, era necessário um adaptador para que esses telefones fossem ligados a rede elétrica.

Chegamos ao seguinte dilema: Os antigos telefones analógicos exigiam apenas um cabo, já os modernos telefones VoIP necessitavam de dois cabos. Parecia um retrocesso ter mais um cabo passando pelas mesas. Essa é razão pela qual o PoE foi inventado, porém, ainda levou alguns anos para essa tecnologia ficar realmente popular, podemos afirmar que a popularidade definitiva do PoE só veio quando também começou a ser utilizado em **Access Point**.

Hoje em dia, o PoE é utilizado para uma gama gigantesca de dispositivos, tais como:

- Telefones VoIP;
- Câmeras IP;
- Access Point;
- Dispositivos IoT;
- Dispositivos PoS;
- Raspberry Pi;
- Arduinos;
- Roteadores e switches de pequeno porte.

Um dispositivo alimentado por PoE é chamado de **powered device (PD)**, algo como ‘dispositivo alimentado’.

Midspan e Endspan

Um dispositivo que fornece energia por cabo ethernet é chamado de **Power Sourcing Equipment (PSE)**, geralmente esse dispositivo é um switch. Temos duas opções de **PSE**: **Midspan** e **Endspan**. Conheceremos sobre essas suas opções:

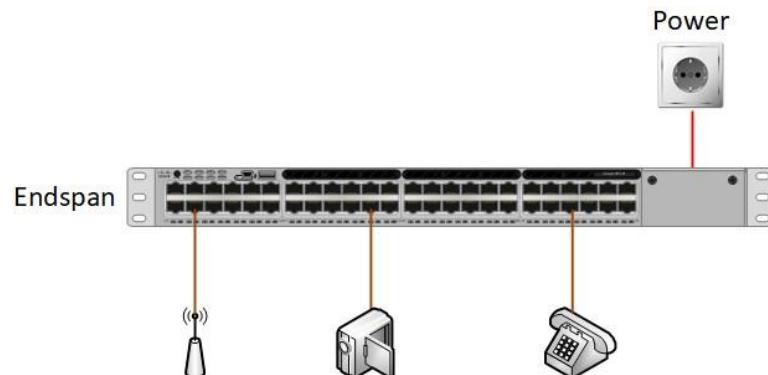
Endspan:

O método endspan usa um switch **com PoE integrado**. Conectamos um dispositivo final ao switch e ele detecta se o dispositivo final é compatível com PoE, em caso afirmativo ele ativa a energia automaticamente.

Os switches PoE vêm em vários tamanhos. Existem switches não gerenciados com 4 portas ou switches gerenciados de grande porte, com dezenas de portas.



Smart Switch Cisco SG200-50P de 50 portas Gigabit PoE



Esquema de um dispositivo Endspan

Midspan:

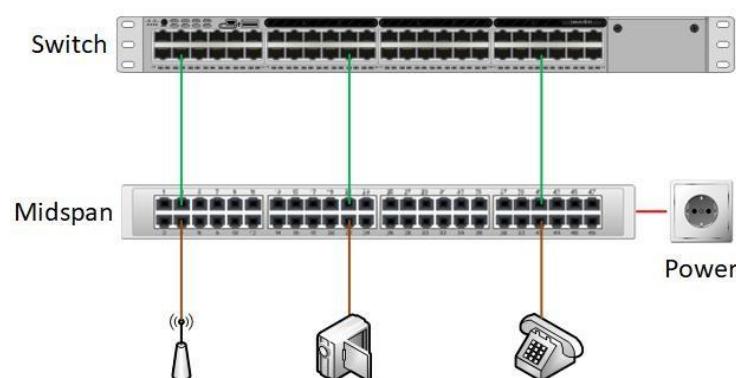
O método Midspan utiliza **injetores PoE (PoE injectors)** para transformar links regulares não PoE, em PoE. Geralmente, eles são utilizados para adicionar dispositivos PoE a rede, **sem que haja necessidade de adicionar ou substituir switches que não suportem a tecnologia PoE**.

Dispositivos Midspan são inteligentes e funcionam de forma similar aos switches PoE, detectando se o **PD** requer energia, em caso afirmativo, ele automaticamente habilita a energia naquela interface. Os injetores PoE são úteis quando temos poucos dispositivos PoE. Em caso de um parque com grande quantidade de dispositivos o melhor a fazer é trocar o parque completo por switches PoE.

Porém, não se engane, existem dispositivos Midspan que suportam dezenas de aparelhos.



Fonte Poe 56v Cisco 16w Power Injector



Esquema de um dispositivo Midspan

Quando usar Midspan ou Endspan:

Qual uma solução deve ser adotada em detrimento da outra? Essa não é uma pergunta difícil de responder, na maioria das vezes o ideal é utilizar o Endspan ou um mix dos dois. Porém, alguns itens a precisam ser considerados antes da escolha:

- Com uma solução midspan, teremos dois dispositivos para gerenciar: O switch e o midspan. Isso por si só já é um problema, pois adiciona outro ponto de falha a rede. Além disso, é necessário espaço extra no rack ou ter vários injetores PoE espalhados no ambiente. Essa é a pior solução possível, pois, teremos vários pontos potenciais de falhas espalhados pelo ambiente, dificultando um possível troubleshooting.
- Midspan é útil caso os switches sejam relativamente novos e não houver verba no momento para substituí-los. Afinal, substituir switches apenas para obter PoE é uma solução cara e inviável financeiramente para algumas empresas.
- Os switches Endspan conseguem fornecer uma quantidade limitada de energia. Em alguns casos, algumas interfaces do switch podem não oferecer a potência máxima exigida pelos dispositivos.

Padrões, classes e tipos PoE:

A Cisco foi a primeira empresa a lançar o PoE, no início dos anos 2000. O PoE criado pela Cisco ficou durante muito tempo como uma tecnologia proprietária, utilizada apenas para alimentar telefones VoIP. Provavelmente você já ouviu falar dessa tecnologia, pois, estamos falando aqui do popular **Cisco Inline Power**.

Com a crescente popularização do PoE, a IEEE, instituto que criou e normatizou diversos padrões Ethernet, agrupou toda tecnologia PoE no padrão 802.3, desenvolvendo assim um padrão aberto.

O padrão inicial adotado foi o 802.3af, porém, a medida que mais dispositivos foram adotando a tecnologia PoE, diversos outros padrões precisaram ser criados além do 802.3af, como por exemplo o 802.3at (conhecido como PoE+) e o 802.3bt (4PPoE).

A tabela abaixo demonstra os principais padrões adotados e que são cobrados no exame CCNA, juntamente com a classe que esses padrões pertencem e para quais tipos de equipamentos são indicados:

Classe	Nome:	Padrão:	Potência máxima PSE*	Potência máxima PD**	Dispositivos suportados ***
1	PoE, PoE de 2 pares	IEEE 802.3af	15,4W	12,95W	Telefones VoIP, Sensores, Access Points (2 antenas), Câmeras CFTV simples.
2	PoE+, PoE Plus	IEEE 802.3at	30W	25,5W	Câmeras CFTV com inclinação e zoom, Access Point (6 antenas), monitores LCD, sensores biométricos, tablets.
3	PoE de 4 pares, PoE 4P, PoE ++, UPOE	IEEE 802.3bt	60W	51W	Dispositivos de Videoconferência, VoIP, dispositivos de automação predial.
4	PoE (High Power)	IEEE 802.3bt	100W	71W	A maioria dos dispositivos que se comunicam por IP.

* Se refere a potência nominal máxima de saída no PSE (Power Sourcing Equipment);

** Potência máxima que chega ao PD; (Powered Device)

*** Podem haver outras aplicações

Como visto na tabela acima, existem 04 tipos diferentes que exigem nossa atenção. Abaixo, discorrerei sobre as principais características e diferenças entre eles:

➤ Tipo 1

O tipo 1 também é conhecido como **PoE** ou **PoE de 2 pares**, este é o primeiro padrão PoE (**802.3af**), adotado no ano 2003. Fornece até 15,4W de energia para cada dispositivo, porém, apenas 12,95W são garantidos devido a perdas que acontecem no cabo. Para este tipo, pode ser utilizado cabos CAT3 ou CAT 5. O Tipo 1 é indicado para telefones VoIP, sensores, Access Points de 2 antenas e modelos mais simples de CFTV.

➤ Tipo 2

Esse é o segundo padrão PoE (**802.3at**), datado do ano de 2009, conhecido também como **PoE + ou PoE Plus**. PoE + pode fornecer até 25,5 W, para um PD e necessita cabos CAT 5 ou superior. Ele é compatível com versões anteriores do tipo 1.

O tipo 2 é útil para dispositivos que requerem mais energia que os dispositivos do tipo 1 podem fornecer. Exemplos são câmeras IP com recursos PTZ, Access Point (6 antenas), monitores LCD, sensores biométricos, tablets.

➤ Tipo 3

O terceiro padrão PoE é baseado em uma implementação proprietária da Cisco de 2014 chamada **Universal Power Over Ethernet (UPOE)**. Essa implementação mais tarde se tornou o padrão **802.3bt**, que foi ratificado em 2018.

O tipo 3 também é conhecido como **PoE ++, 4 PPoE, PoE de 4 pares, Ultra PoE ou PoE de alta potência**. O tipo 3 utiliza todos os quatro pares de fios, e suporta o dobro da potência do tipo 2, ou seja até 60W no PSE e aproximadamente 51W para cada dispositivo.

Ele é utilizado em dispositivos como Thin clients, dispositivos de videoconferência, dispositivos de automação predial, Point of Sale (PoS) systems.

➤ Tipo 4

O tipo 4 também foi baseado no padrão **802.3bt**, e é conhecido como **UPOE + ou PoE de maior potência (higher-power PoE)**, oferece potência máxima de 100W por porta e até 71W para cada PD. Pode ser utilizado em dispositivos como laptops ou até em TVs.

Dispositivos de alimentação PoE:

Existem **três modos** para alimentar os dispositivos PoE: **A, B e 4 pares**:

- **Modo A:** Fornece energia nos mesmos pares de dados utilizado nas redes 10BASE-T e 100BASE-TX. Existem duas opções (MDI e MDI-X) que utilizam polaridades diferentes. Isso permite usar PoE com qualquer tipo de cabo, seja cabo direto, cruzado ou auto MDI-X. Este método é frequentemente usado com switches Endspan PoE.
- **Modo B:** Fornece energia através dos pares de fios sobressalentes. Isso é possível porque as redes 10BASE-T e 100BASE-TX usam apenas dois pares para dados. Os injetores PoE (midspan) costumam utilizar esse modo.
- **4 pares:** Fornece energia em todos os quatro pares de fios em redes 1000BASE-T ou superior, nesse padrão de cabeamento, todos os quatro pares são utilizados para dados, portanto, não há fios sobressalentes para alimentação.

O PSE decide qual modo de energia será usado.

Vantagens da utilização do PoE:

O PoE possui algumas vantagens, abaixo, listarei as quatro vantagens mais significativas:

- 1 **Economia de custos:** A economia está em não ser necessário a instalação de adaptadores de energia e cabeamento elétrico separado. Tudo funciona apenas utilizando cabos Ethernet.
- 2 **Confiabilidade:** PoE vem de uma fonte de energia central (geralmente um switch de rede) em vez de adaptadores de energia que estão espalhados. É recomendável que seja conectado um UPS ao switch PoE, para que os dispositivos permaneçam energizados mesmo quando a energia principal estiver desligada.
- 3 **Segurança:** O fornecimento de energia é feito de forma inteligente, protegendo os equipamentos contra problemas típicos, como sobrecarga ou falta de energia.
- 4 **Flexibilidade:** A possibilidade de posicionar e reposicionar dispositivos finais, como câmeras IP, de maneira fácil e rápida. Só é necessário um cabo de rede, não precisando mais se preocupar se há uma tomada elétrica no local.

Desvantagens da utilização do PoE:

Como tudo na vida, também existem desvantagens na utilização do PoE! Estas desvantagens estão diretamente relacionadas com energia e calor.

- 1 Geração de calor:** Aumentar o fornecimento de energia aumenta o fluxo da corrente, o que resulta na geração de calor extra nos cabos de rede.
- 2 Aumento da temperatura ambiente:** A implantação de PoE de alta potência pode aumentar a média da temperatura ambiente, gerando custo extra com climatização.
- 3 Segurança:** Temperaturas excessivas podem levar ao derretimento do isolamento dos cabos, podendo provocar curtos e a falha do hardware.
- 4 Desgaste do hardware:** Devido às altas temperaturas, pode ocorrer um desgaste excessivo do hardware, encurtando sua vida útil.
- 5 Desempenho:** O calor aumenta a perda de sinal e reduz a distância máxima dos cabos Ethernet. O comprimento máximo de um cabo CAT 5 que não esteja utilizando o PoE, é de 100 metros. Porém, dependendo do tipo de cabo e da quantidade de energia que esteja sendo transportada é bem provável que não seja possível alcançar essa distância utilizando PoE.

Resumo:

Aprendemos os conceitos básicos de PoE para o CCNA e aprendemos também a maneira correta de energizar os dispositivos de rede por meio dos cabos Ethernet. Como foi um tópico extenso, farei um pequeno resumo das partes mais importantes:

- Um dispositivo alimentado através do PoE (por exemplo: câmera IP, VoIP ou Access point) é um **dispositivo energizado (PD) ou powered device**.
- Um dispositivo que fornece energia PoE é um **Power sourcing equipment (PSE)**.
- Existem duas opções de PSE:
 - **Endspan:** Switch com PoE integrado.
 - **Midspan:** Injetor PoE
- Existem **quatro tipos** de PoE e três padrões:
 - **Tipo 1** (802.3af) - até 15,4W de PSE
 - **Tipo 2** (802.3at) - até 30W de PSE
 - **Tipo 3** (802.3bt) - até 60W de PSE
 - **Tipo 4** (802.3bt) - até 100W de PSE
- Existem três maneiras de alimentar os **PDs**:
 - **Modo A:** Fornecendo energia nos mesmos pares de fios utilizados para tráfego de dados na rede: 10BASE-T e 100BASE-TX.
 - **Modo B:** Fornecendo energia nos pares de fios sobressalentes: 10BASE-T e 100BASE-TX.
 - **4 pares:** Fornecendo energia em todos os quatro pares de fios.
- As diferentes classes definem quanta potência um **PSE** oferece a um **PD**.
- PoE tem muitas vantagens, dentre elas: Ser uma fonte de energia central, dessa forma economizando custos; ser mais confiável que adaptadores de energia distribuídos, pois é um ponto único de gerência e falha.
- As desvantagens do PoE estão relacionadas ao aumento do calor nos cabos, equipamentos e ambientes.

Introdução ao Cisco IOS CLI (Interface de linha de comando)

Até aqui, seguimos rigorosamente os tópicos conforme aparecem no blueprint da prova, porém, é hora de entrarmos na parte prática. Esse capítulo será dedicado a ensinar como ligamos, conectamos e administrarmos os dispositivos da Cisco. Faremos isso através de linha de comando, a famosa tela preta.

É bem provável que você não tenha switches e roteadores em casa para praticar, por isso, aconselho que vocês utilizem ferramentas de simulação de redes como o programa Packet Tracer da Cisco ou emuladores como o GNS3 e Eve-NG, que trabalham com o sistema operacional real dos roteadores e switches.

Na maioria dos dispositivos da Cisco (incluindo roteadores e switches) utilizamos CLI (Command Line Interface) para configurar os dispositivos. Como dito anteriormente, a CLI é uma interface baseada em texto (a famosa tela preta), em que digitamos comandos de verificação\configuração para obter informações ou configurar determinado dispositivo.

Também existem GUIs (Graphical User Interface) para os roteadores, switches e firewalls, mas a maior parte do trabalho é realizado na CLI.

Pode parecer estranho ter que digitar comandos em uma tela preta em pleno 2021, porém, dentre diversas vantagens, a CLI permite controle total sobre todos os aspectos do dispositivo, além de facilitar o trabalho quando é necessário copiar configurações inteiras de um dispositivo para outro.

Acesso ao Cisco IOS

Antes de inserir qualquer comando é necessário acessar a CLI. Para isso, existem três opções:

- Console
- Telnet
- SSH

O console é uma porta física no dispositivo, que permite acesso direto a CLI. Normalmente é utilizada na primeira vez que configuramos um switch\roteador. Já às opções de Telnet e SSH são utilizadas para acesso remoto.

Conectando a porta console

Geralmente há uma ou duas portas para entrada do console. Observe a imagem abaixo:



Switch Cisco Catalyst 2960-L Series

No lado esquerdo deste switch Catalyst 2960, destacado com uma cor diferente, há duas portas para conexão do cabo console, uma porta RJ45 e uma porta micro-USB. Os dispositivos mais antigos, possuem apenas a porta para RJ45, os switches mais novos (e outros dispositivos) geralmente possuem as duas opções.

Preste bem atenção, mesmo sendo uma porta RJ45, essa porta não é uma porta Ethernet. Usamos essa conexão para conectar o switch a uma porta serial no computador utilizando o seguinte cabo:



Cabo console popularmente conhecido como cabo azul

Esse cabo é chamado de cabo console, uma ponta dele é ligado na entrada RJ45 do dispositivo e a outra na porta serial do computador. Provavelmente não há uma porta serial no seu computador, isso porque os computadores ou laptops modernos não têm mais essas portas, portanto, é necessário usar um cabo “Serial para USB”, como o cabo abaixo:



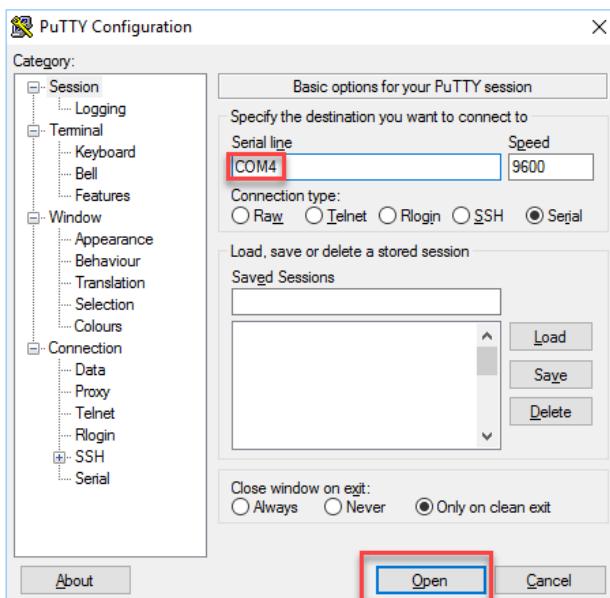
Cabo Serial - Usb

A função deste cabo é emular uma porta serial através de uma conexão USB. Depois de conectá-lo no computador e no switch (ou outro dispositivo qualquer), podemos iniciar um programa emulador de terminal, como o Putty, para acessar a CLI.

Emulador de terminal (terminal emulator)

Há dezenas de aplicativos de emulador de terminal, sendo o Putty o mais aconselhável, pois é amigável para quem está começando. Ele é gratuito, e permite que se conecte ao dispositivo usando uma conexão serial, telnet ou SSH.

Esta é a tela principal do Putty:



Tela inicial do Putty

Para conectarmos aos dispositivos da Cisco como switches e roteadores, é necessário selecionar e preencher os campos da maneira acima:

- Selecionar a opção “Serial”.
- Em “Speed” colocar a velocidade padrão de 9600 (taxa de transmissão).
- A porta COM dependendo computador, pode ser COM1, COM2, etc. Caso você não tenha certeza, é possível verificar através do gerenciador de dispositivos no Windows. Por experiência em 90% das vezes será a COM4, como foi no exemplo que estou apresentado.

Após seguir esses passos, é hora de ligar o switch ou, caso já esteja ligado, retirar o cabo e plugar novamente para que ele possa recarregar.

Primeiro boot (inicialização):

Quando o switch for inicializado, uma série de informações será mostrada no console. Primeiro ele inicializará a memória flash:

```
Boot Sector Filesystem (bs) installed, fsid: 2
Base ethernet MAC Address: 00:11:bb:0b:36:00
Xmodem file system is available.
The password-recovery mechanism is disabled.

Initializing Flash...

flashfs[0]: 14 files, 4 directories
flashfs[0]: 0 orphaned files, 0 orphaned directories
flashfs[0]: Total bytes: 15998976
flashfs[0]: Bytes used: 12794368
flashfs[0]: Bytes available: 3204608
flashfs[0]: flashfsfsck took 9 seconds.

...done

Initializing Flash.

done.
```

A inicialização da memória flash é necessária, pois ela contém a imagem do IOS (sistema operacional) do switch. A próxima etapa é carregar a imagem do IOS que está armazenada na memória flash:

```
Loading "flash:/c3560-ipservicesk9-mz.122-  
55.SE10.bin"...@aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa  
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa  
File "flash:/c3560-ipservicesk9-mz.122-55.SE10.bin" uncompressed and installed,  
entry point: 0x1000000  
  
executing...
```

A imagem do IOS fica compactada na memória flash, o switch então, descompacta a imagem e carrega na memória RAM.

Em seguida, são apresentadas algumas informações legais e informações sobre o switch:

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c) of the Commercial Computer Software - Restricted Rights clause at FAR sec. 52.227-19 and subparagraph

(c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS sec. 252.227-7013.
cisco Systems, Inc.

170 West Tasman Drive
San Jose, California 95134-1706

Cisco IOS Software, C3560 Software (C3560-IPSERVICESK9-M), Version 12.2(55)SE10, RELEASE SOFTWARE (fc2)

Technical Support: <http://www.cisco.com/techsupport>

Copyright (c) 1986-2015 by Cisco Systems, Inc.

Compiled Wed 11-Feb-15 11:34 by prod_rel_team

Image text-base: 0x01000000, data-base: 0x02F00000

Essa tela informa a versão IOS (destacado acima). O sistema operacional agora está instalado e funcionando.

A próxima etapa é a realização do POST (Power on Self Test), aqui ocorre a verificação dos componentes do switch:

```
POST: CPU MIC register Tests: Begin
POST: CPU MIC register Tests: End, Status Passed
POST: Port ASIC Memory Tests: Begin
POST: Port ASIC Memory Tests: End, Status Passed
POST: CPU MIC interface Loopback Tests: Begin
POST: CPU MIC interface Loopback Tests: End, Status Passed
POST: Port ASIC Ring Loopback Tests: Begin
POST: Port ASIC Ring Loopback Tests: End, Status Passed
POST: Inline Power Controller Tests: Begin
POST: Inline Power Controller Tests: End, Status Passed
POST: Port ASIC CAM Subsystem Tests: Begin
POST: Port ASIC CAM Subsystem Tests: End, Status Passed
POST: Port ASIC Port Loopback Tests: Begin
POST: Port ASIC Port Loopback Tests: End, Status Passed
Waiting for Port download...Complete
```

Em seguida, é mostrado informações sobre os recursos criptográficos presente no dispositivo:

This product contains cryptographic features and is subject to United States and local country laws governing import, export, transfer and use. Delivery of Cisco cryptographic products does not imply

third-party authority to import, export, distribute or use encryption. Importers, exporters, distributors and users are responsible for compliance with U.S. and local country laws. By using this product you agree to comply with applicable laws and regulations. If you are unable to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:
<http://www.cisco.com/wwl/export/crypto/tool/stqrg.html>

If you require further assistance please contact us by sending email to export@cisco.com.

Uma dúvida comum aflige a maioria dos estudantes nessa tela, e pode estar lhe afigindo também: O que um switch tem a ver com criptografia?

Dependendo da imagem do IOS, o switch é capaz de trabalhar com SSH, e o SSH por ser seguro necessita de criptografia. Além do SSH, outro recurso que faz uso da criptografia é o SNMP versão 3. SNMP é usado para gerenciamento da rede, com ele é possível ler estatísticas dos dispositivos. Não se preocupe, falaremos sobre SNMP no momento certo.

A parte final do processo de BOOT (inicialização) mostra algumas informações gerais sobre o switch:

```
cisco WS-C3560-24PS (PowerPC405) processor (revision G0) with 131072K bytes of memory.

Processor board ID CAT0832N0G3

Last reset from power-on

1 Virtual Ethernet interface

24 FastEthernet interfaces

2 Gigabit Ethernet interfaces

The password-recovery mechanism is disabled.

512K bytes of flash-simulated non-volatile configuration memory.

Base ethernet MAC Address      : 00:11:BB:0B:36:00
Motherboard assembly number    : 73-9299-01
Power supply part number       : 341-0029-03
Motherboard serial number      : CAT083107CZ
Power supply serial number     : DTH08282MZA
Model revision number          : G0
Motherboard revision number    : E0
```

```

Model number : WS-C3560-24PS-S
System serial number : CAT0832N0G3
Top Assembly Part Number : 800-24791-01
Top Assembly Revision Number : K0
Version ID : N/A
Hardware Board Revision Number : 0x09

Switch Ports Model          SW Version          SW Image
-----
*   1 26    WS-C3560-24PS    12.2(55)SE10    C3560-IPSERVICESK9-M

Press RETURN to get started!

```

A tela acima mostra o modelo do switch, quantidade de interfaces, números de série, imagem utilizada, etc.

O switch está totalmente ligado e pronto para ser configurado, caso essa seja a primeira vez que esteja sendo ligado (caso você esteja utilizando emulador\simulador o processo é exatamente o mesmo), a seguinte mensagem será apresentada:

```

--- System Configuration Dialog ---

Enable secret warning

-----
In order to access the device manager, an enable secret is required
If you enter the initial configuration dialog, you will be prompted for the
enable secret
If you choose not to enter the intial configuration dialog, or if you exit setup
without setting the enable secret,
please set an enable secret using the following CLI in configuration mode-
enable secret 0 <cleartext password>

-----
Would you like to enter the initial configuration dialog? [yes/no]:

```

Se não houver configuração, o switch perguntará se gostaríamos de seguir um assistente, esse assistente é chamado de ‘initial configuration dialog’ (algo como caixa de diálogo de configuração inicial). Digite “no” para que a configuração comece totalmente do zero. Nós mesmos configuraremos o dispositivo em todos os seus mínimos aspectos.

Modo de usuário e privilegiado (User and Enable mode)

Após o switch ter sido inicializado e a tecla enter ter sido pressionada, chegaremos ao chamado **modo de usuário (user mode) ou modo EXEC (EXEC mode)**. Neste modo, há algumas restrições, só é permitido usar alguns comandos simples.

A linha de comando apresentará a saída abaixo:

```
Switch>
```

O símbolo > serve para indicar que estamos no **user mode**. Para obter acesso total ao switch, temos que entrar no modo privilegiado (**privileged mode**), também chamado de **enable mode** (este é o nome que usamos mais frequentemente). Para fazer isso é muito simples, basta digitar o comando abaixo:

```
Switch>enable
```

```
Switch#
```

Repare que o símbolo > mudou para #. Isso nos diz que agora estamos no modo privilegiado, e temos controle e acesso total ao switch, para retornar ao modo de usuário, basta digitar o comando **disable**:

```
Switch# disable
```

```
Switch>
```

Apagando as configurações do dispositivo

Algumas vezes, podemos pegar um hardware usado, com configurações antigas. Nesse caso pode ser necessário apagar (resetar) as configurações e começar do zero. Vamos aprender a fazer isso:

```
Switch# erase startup-config
```

```
Erasing the nvram filesystem will remove all configuration files! Continue?
```

```
[confirm]
```

```
[OK]
```

```
Erase of nvram: complete
```

Basta digitar o comando ‘erase startup-config’, o switch perguntará se temos certeza que desejamos apagar todos os arquivos de configuração. Entre colchetes está escrito “confirmar”, sempre que você ver algo entre [], só precisa pressionar ‘enter’ para aquela opção ser executada. No caso, como explicado, não há necessidade de digitar “confirm”.

Switches também armazenam informações sobre as VLANs (Virtual Local Area Network) em outro arquivo. O que é uma VLAN e o que ela faz será abordado em outro tópico, por enquanto, vamos apenas garantir que o arquivo em que elas ficam armazenadas seja excluído:

```
Switch# delete flash:vlan.dat
```

```
Delete filename [vlan.dat]?
```

```
Delete flash:vlan.dat? [confirm]
```

O comando: delete **flash: vlan.dat** exclui o arquivo vlan.dat que é onde ficam armazenados os dados das vlans. Como dito anteriormente, só é necessário pressionar “enter” para confirmar o que o IOS nos diz entre os colchetes. Caso apareça um erro informando que esse arquivo não existe, não se preocupe. Isso significa que outra pessoa já excluiu as informações da VLAN e o arquivo não está mais presente.

Para finalizar, é necessário reiniciar o switch:

```
Switch#reload  
  
Proceed with reload? [confirm]
```

Comandos Show

O comando show provavelmente é o comando mais usado nos dispositivos Cisco. Com ele é possível achar qualquer informação no dispositivo. Vamos começar com exemplos simples. Todos os comandos abaixo foram aplicados em um switch, mas funcionariam em um roteador:

Show version:

O comando **show version** fornece informações sobre o switch, incluindo o modelo, imagem do IOS, quanto tempo está ligado, número de série, etc:

```
Switch#show version  
  
Cisco IOS Software, C3560 Software (C3560-IPSERVICESK9-M), Version 12.2(55)SE10,  
RELEASE SOFTWARE (fc2)  
  
Technical Support: http://www.cisco.com/techsupport  
  
Copyright (c) 1986-2015 by Cisco Systems, Inc.  
  
Compiled Wed 11-Feb-15 11:34 by prod_rel_team  
  
Image text-base: 0x01000000, data-base: 0x02F00000  
  
  
ROM: Bootstrap program is C3560 boot loader  
  
BOOTLDR: C3560 Boot Loader (C3560-HBOOT-M) Version 12.2(44)SE5, RELEASE SOFTWARE  
(fc1)  
  
  
Switch uptime is 54 minutes  
  
System returned to ROM by power-on  
  
System image file is "flash:/c3560-ipservicesk9-mz.122-55.SE10.bin"  
  
  
  
  
This product contains cryptographic features and is subject to United  
States and local country laws governing import, export, transfer and  
use. Delivery of Cisco cryptographic products does not imply  
third-party authority to import, export, distribute or use encryption.  
Importers, exporters, distributors and users are responsible for  
compliance with U.S. and local country laws. By using this product you  
agree to comply with applicable laws and regulations. If you are unable
```

to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:

<http://www.cisco.com/wwl/export/crypto/tool/stqrg.html>

If you require further assistance please contact us by sending email to
export@cisco.com.

cisco WS-C3560-24PS (PowerPC405) processor (revision G0) with 131072K bytes of
memory.

Processor board ID CAT0832N0G3

Last reset from power-on

1 Virtual Ethernet interface

24 FastEthernet interfaces

2 Gigabit Ethernet interfaces

The password-recovery mechanism is disabled.

512K bytes of flash-simulated non-volatile configuration memory.

Base ethernet MAC Address : 00:11:BB:0B:36:00

Motherboard assembly number : 73-9299-01

Power supply part number : 341-0029-03

Motherboard serial number : CAT083107CZ

Power supply serial number : DTH08282MZA

Model revision number : G0

Motherboard revision number : E0

Model number : WS-C3560-24PS-S

System serial number : CAT0832N0G3

Top Assembly Part Number : 800-24791-01

Top Assembly Revision Number : K0

Version ID : N/A

Hardware Board Revision Number : 0x09

Switch Ports Model	SW Version	SW Image
--------------------	------------	----------

*	1 26 WS-C3560-24PS	12.2(55)SE10	C3560-IPSERVICESK9-M
---	--------------------	--------------	----------------------

Configuration register is 0xF

show running-config

O comando **show running-config** mostra toda configuração ativa do switch. Embora não tenhamos configurado nada ainda assim há uma configuração básica.

```
Switch#show running-config
Building configuration...

Current configuration : 1237 bytes
!
version 12.2
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Switch
!
boot-start-marker
boot-end-marker
!
!
!
!
no aaanew-model
system mtu routing 1504
!
!
!
!
!
!
!
!
spanning-tree mode pvst
```

```
spanning-tree extend system-id
!
vlan internal allocation policy ascending
!
!
!
!
interface FastEthernet0/1
!
interface FastEthernet0/2
!
interface FastEthernet0/3
!
interface FastEthernet0/4
!
interface FastEthernet0/5
!
interface FastEthernet0/6
!
interface FastEthernet0/7
!
interface FastEthernet0/8
!
interface FastEthernet0/9
!
interface FastEthernet0/10
!
interface FastEthernet0/11
!
interface FastEthernet0/12
!
interface FastEthernet0/13
```

```
!
interface FastEthernet0/14
!
interface FastEthernet0/15
!
interface FastEthernet0/16
!
interface FastEthernet0/17
!
interface FastEthernet0/18
!
interface FastEthernet0/19
!
interface FastEthernet0/20
!
interface FastEthernet0/21
!
interface FastEthernet0/22
!
interface FastEthernet0/23
!
interface FastEthernet0/24
!
interface GigabitEthernet0/1
!
interface GigabitEthernet0/2
!
interface Vlan1
    no ip address
!
ip classless
ip http server
```

```

ip http secure-server
!
!
!
!
!
line con 0
line vty 5 15
!
end

```

show mac address-table dynamic

O comando **show mac address-table dynamic** informa todos os endereços MAC que o switch aprendeu. Neste exemplo, ele aprendeu apenas um endereço MAC, na interface Fa0/12 (Interface Fast Ethernet 12):

```

Switch#show mac address-table dynamic

      Mac Address Table

-----
Vlan      Mac Address          Type      Ports
-----



      1      0050.568e.d3c8    DYNAMIC    Fa0/12

Total Mac Addresses for this criterion: 1

```

Existem centenas de opções para o comando ‘**show**’. Há livros com centenas de páginas só sobre comandos, e nesses livros, o comando ‘**show**’ ocupa várias dezenas de páginas, portanto, é impossível listar todos os comandos nesse livro.

Durante o livro, muitos desses comandos serão utilizados para explicar de forma prática os mais diversos conceitos.

Outro comando importante para analisar o comportamento dos dispositivos é o “**debug**” (depuração). Enquanto o comando “**show**” produz apenas informações “estáticas”, o comando de “**debug**” permite ver o que está acontecendo em tempo real. Veremos alguns exemplos do comando ‘**debug**’ e seus complementos mais pra frente.

Configurações iniciais:

Quando compramos um switch novo, ele funcionará assim que ligarmos na tomada com sua configuração default, que no caso é vazia. Ele se comportará como um switch não gerenciado. Começará a aprender endereços MAC e encaminhará os frames Ethernet.

Para que ele se torne um switch gerenciável, teremos que fazer algumas alterações na sua configuração. Essas modificações incluem: Alterar o nome, adicionar endereço IP para possibilitar gerenciamento remoto, etc.

O primeiro passo é entrar no *modo de configuração*. Neste modo, podemos fazer alterações na configuração do switch. Relembre passo a passo como entrar na **configuration mode**:

Primeiro utilizamos o comando “enable”:

```
Switch>enable
```

Agora, temos que utilizar o comando “configure terminal”:

```
Switch#configure terminal
```

Com o comando **configure terminal** entramos no **modo de configuração global**. Estamos prontos para fazer as alterações no switch.

Vamos começar mudando o nome do switch (lembrando que poderia ser um roteador, firewall) com o comando **hostname**, vamos modificar o nome do switch para SW1:

```
Switch(config)#hostname SW1
```

```
SW1(config)#
```

O comando acima foi executado no **modo de configuração “global”**. Quando queremos fazer alterações nas interfaces ou configurações do console, temos que mergulhar em um dos submodos de configuração. Por exemplo, vamos colocar uma senha para acesso ao console:

```
SW1(config)#line console 0
```

```
SW1(config-line)#password cisco123
```

```
SW1(config-line)#login
```

Primeiro, usamos o comando ‘**line console 0**’ para entrar no submodo de configuração do console. Repare que entre parênteses está (config-line). O comando de ‘**password**’ é utilizando para especificar uma senha, no caso a senha escolhida foi cisco123. O comando ‘**login**’ informa ao switch que ele tem de solicitar uma senha para acesso ao console, dessa forma, na próxima vez que acessarmos o console, será solicitada essa senha.

Para retornar para a configuração global existem 02 opções: Digitar ‘**exit**’ ou pressionar **CTRL + Z**:

```
SW1(config-line)#exit
```

```
SW1(config)#
```

Estamos de volta ao modo de configuração global. Vamos para mais um exemplo: Alterar a configuração da primeira interface do nosso switch:

```
SW1(config)#interface FastEthernet 0/1
```

```
SW1(config-if)#
```

O comando utiliza é “interface + interface específica” que desejamos fazer alterações, no caso a FastEthernet 0/1. Observe que novamente mudou o que está dentro dos parênteses. Agora estamos no submodo da interface (config-if).

Importante, o switch não mostra qual interface estamos, apenas que estamos configurando uma interface ou submodo.

Vamos fazer algumas alterações nesta interface:

O primeiro passo é colocar uma descrição para que saibamos o que está conectado naquela interface, configurar a velocidade de conexão e o modo:

```
SW1(config-if)#description CONEXÃO_COM_COMPUTADOR01
```

```
SW1(config-if)#duplex full
```

```
SW1(config-if)#speed 100
```

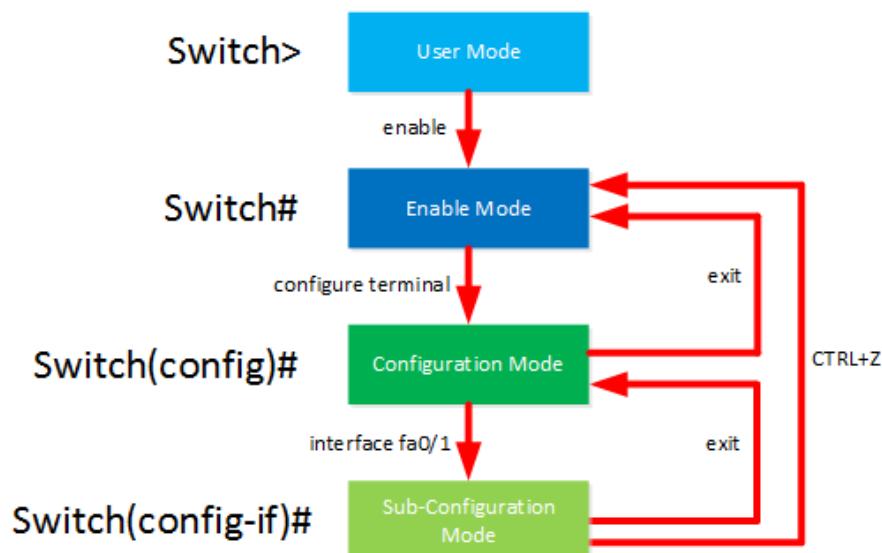
Para retornar basta digitar o comando **exit** ou **CTRL-Z**:

```
SW1(config-if)#exit  
SW1(config)#
```

Para se localizar sempre observe o que está entre parênteses. Na primeira vez que demos o comando “**exit**” o switch voltou para o **modo de configuração global**. Na segunda vez que aplicarmos o comando “**exit**” sairemos do “**configuration mode**” para **enable mode**:

```
SW1(config)#exit  
SW1#
```

Abaixo um esquema para facilitar o aprendizado:



Esquema demonstrando os modos de configuração dos dispositivos Cisco

Salvando às configurações

Até agora fizemos algumas configurações básicas no switch, mas ainda não salvamos nada, corremos sério risco de haver uma queda de energia e perdemos tudo que foi feito.

Todas as configurações estão na **running-config** (algo como configuração em execução), que está armazenada na memória RAM do dispositivo, o que quer dizer que caso o dispositivo seja reiniciado ou desligado, a configuração será perdida.

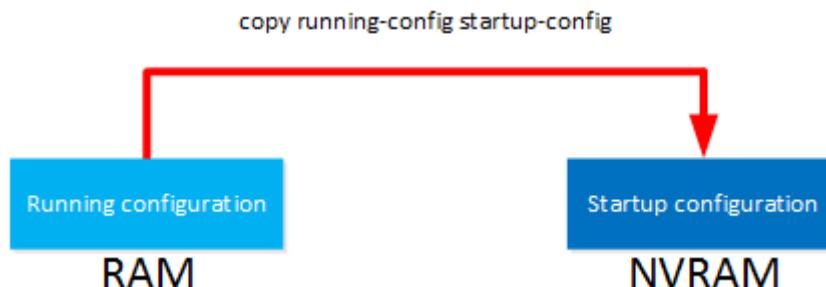
Para salvar essa configuração de forma definitiva, temos que salvá-la para a **startup config** (configuração de inicialização) que é armazenada na NVRAM. Da próxima vez que inicializarmos o switch é lá que ele procurará pela configuração de inicialização para usar.

Vamos copiar a **running-config** para a **startup config**:

```
SW1#copy running-config startup-config  
Destination filename [startup-config]?  
Building configuration...
```

```
[OK]
0 bytes copied in 1.182 secs (0 bytes/sec)
```

O comando **copy** é usado para copiar as configurações que estão na **running-config** para a **startup config**. Abaixo apresento uma pequena ilustração para ajudá-lo a visualizar os dois arquivos de configuração:



Outro comando popular para salvar a configuração é “**wr**”. Este comando é a abreviação de **write**, que é outro comando usado para salvar configurações. O **wr** faz exatamente a mesma coisa que o **copy running-config startup-config**.

Ferramentas de ajuda:

Já sabemos ligar e verificar algumas configurações em um dispositivo Cisco, usando o comando “**show**”, e alguns comandos de configuração. Agora é hora de aprendermos alguns truques para deixar mais fácil a vida de quem opera pela CLI. Vamos analisar alguns:

Ponto de interrogação

Deu branco na hora de digitar um comando? Não lembra exatamente como se digita? Para ocasiões como essa conte com ponto de interrogação. Observe uma das possíveis forma de utilização, em que ele mostra todos os comandos possíveis:

```
SW1#?
Exec commands:
  access-enable      Create a temporary Access-List entry
  access-template    Create a temporary Access-List entry
  archive           manage archive files
  beep               Blocks Extensible Exchange Protocol commands
  cd                Change current directory
  clear              Reset functions
  clock              Manage the system clock
  cns               CNS agents
  configure         Enter configuration mode
```

O ponto de interrogação funciona em todos os modos e submodos, experimente e explore o máximo possível. Outro recurso interessante do “?” é que ele ajuda a descobrir quais comandos são possíveis através das letras iniciais. Por exemplo:

```
SW1#cl?
Clear clock
```

Se digitarmos **cl?** a CLI nos mostrará que existem dois comandos possíveis:

- clear
- clock

Vamos continuar usando o ponto de interrogação para aprofundarmos mais no comando **clock**, pois é um ótimo exemplo para explicar um pouco mais sobre as funcionalidades do ponto de interrogação. Por exemplo: Qual formato é usado para definir a hora? Poderia ser 18:00, 06pm, 6pm ou alguma outra forma. Utilizando o ponto de interrogação descobriremos exatamente a sintaxe do comando:

```
SW1#clock ?
set Set the time and date
```

Primeiro ele nos diz que precisamos usar o comando **set**. Vamos usá-lo:

```
SW1#clock set ?
hh:mm:ss Current Time
```

Descobrimos que o formato correto para configuração do horário é: **hh: mm: ss**. Vamos configurar conforme o dispositivo nos indica:

```
SW1#clock set 18:05:00 ?
<1-31> Day of the month
MONTH Month of the year
```

Apertando ‘?’ novamente, descobrimos que também precisamos configurar o dia e o mês. Começamos pelo mês:

```
SW1#clock set 18:05:00 August ?
<1-31> Day of the month
```

Agora definiremos o dia:

```
SW1#clock set 18:05:00 August 9 ?
<1993-2035> Year
```

E para encerrar, precisamos informar o ano:

```
SW1#clock set 18:05:00 August 9 2021 ?
<cr>
```

Usando o ponto de interrogação novamente no final do comando vemos apenas <cr>, o que significa que o comando **clock** já está com todos os argumentos que precisa para ser aplicado. Vamos retirar ponto de interrogação e pressionar “Enter”:

```
SW1#clock set 18:05:00 August 9 2021
SW1#
```

Com isso, acabamos de configurar o relógio do dispositivo.

Abreviações:

Não há necessidade de digitar o comando exato para a CLI aceitá-lo, podemos abreviar os comandos. Por exemplo, a maneira abaixo é uma maneira de abreviar o comando **copy running-config**, note que não é preciso digitar todas as letras do comando:

```
SW1#copy run st
```

Para as abreviações funcionarem é necessário que elas sejam únicas, ou seja, no exemplo acima após o comando **copy**, não há mais nenhum parâmetro possível que comece com “**run**” apenas ‘**running-config**’. Da mesma forma, o único parâmetro que começa com “**st**” é ‘**startup-config**’.

Eu sei que tudo isso pode parecer um pouco confuso, mas com um pouco de treino com a CLI e depois de se familiarizar com os comandos, você utilizará essas ferramentas de forma automática, sem sequer pensar.

Tratamento de comandos errados e incompletos:

Em um mundo perfeito, lembraríamos de tudo e não cometéramos erros de ortografia. Na vida real, erramos o tempo todo. Felizmente para nós, a CLI possui recursos para nos ajudar. Vamos exemplificar utilizando novamente comando **clock**:

```
SW1#clock set 19:05:00 8
```

```
% Incomplete command.
```

Na saída do comando acima, o switch nos diz que o comando está incompleto. Isso porque não colocamos o mês e o ano. Quando acontecer esse tipo de erro use o ponto de interrogação para descobrir quais outros parâmetros o comando exige.

E qual a saída nos é mostrada quando cometemos um erro de digitação?

```
SW1#clock set 18:05:00 8
```

```
^
```

```
% Invalid input detected at '^' marker.
```

A saída do comando informa que cometemos um erro e indica com o símbolo ‘^’ exatamente onde erramos. Quando isso acontecer, remova tudo o que você digitou acima do símbolo ‘^’ e use o ponto de interrogação para descobrir o que está errado:

```
SW1#clock set 14:05:00 8 ?
```

```
MONTH Month of the year
```

Perceba, nós deveríamos ter escrito August e não 8.

Atalhos de teclado:

Existem alguns atalhos de teclado úteis que podemos utilizar com a CLI.

Por exemplo, o Cisco IOS mantém um histórico dos comandos inseridos anteriormente. Tudo que precisamos fazer para acessar esse histórico é pressionar as **teclas de seta para cima e para baixo**, dessa forma navegamos pelos comandos anteriores.

Usando as teclas de seta para a esquerda e para direita, podemos mover o cursor um caractere em qualquer direção, de acordo com a direção da seta. Caso haja necessidade de fazer alguma alteração em um comando muito longo, existe uma maneira mais simples do que manter as setas apertadas. Pode-se usar as **combinações CTRL + A** ou **Ctrl + E**, esse comando fará com que o cursor salte **para o início ou o fim da linha**.

Não tem ideia de como digitar um determinado comando? O botão TAB completa os comandos automaticamente. Por exemplo, digite este comando abaixo:

```
SW1#show mac ad
```

Agora, pressione a tecla TAB, observe que a CLI completará automaticamente o comando:

```
SW1#show mac address-table
```

Acostume-se a utilizar esse recurso, você economizará tempo na digitação e não precisará pensar em detalhes bobos, do tipo se um comando tem um espaço ou traço.

OBS: Se ao apertar o botão TAB e nada acontecer, utilize o ponto de interrogação. Provavelmente haverá mais de um comando com as mesmas iniciais.

Comando DO

Quando estivermos em modo de configuração, caso tentemos aplicar alguns comandos aparecerá a seguinte mensagem:

```
SW1(config)#show version
^
% Invalid input detected at '^' marker.
```

Entendeu o porquê? Observe, o comando foi digitado corretamente, mas o problema é que este é um comando para ser utilizado no modo enable, não no modo de configuração.

Uma opção seria sair do modo de configuração, mas, em vez disso, podemos adicionar o comando '**do**' na frente do comando show:

```
SW1(config)#do show version
Cisco IOS Software, C3560 Software (C3560-IPSERVICESK9-M), Version 12.2(55)SE10,
RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2015 by Cisco Systems, Inc.
```

Problema resolvido!

Modificadores de saída:

A saída do comando show pode ter milhares de linhas, olhar todas essas linhas até achar o trecho que nos interessa pode ser muito enfadonho. Veja o exemplo abaixo:

```
SW1#show version
Cisco IOS Software, C3560 Software (C3560-IPSERVICESK9-M), Version 12.2(55)SE10,
RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2015 by Cisco Systems, Inc.
Compiled Wed 11-Feb-15 11:34 by prod_rel_team
Image text-base: 0x01000000, data-base: 0x02F00000
ROM: Bootstrap program is C3560 boot loader
BOOTLDR: C3560 Boot Loader (C3560-HBOOT-M) Version 12.2(44)SE5, RELEASE SOFTWARE
(fc1)
[output omitted]
```

É uma saída grande, porém, e se quisermos saber apenas a versão do IOS que está correndo no switch? Para casos assim, podemos usar modificadores de saída:

```
SW1#show version ?  
| Output modifiers
```

Basta ao término do comando show, adicionar o símbolo |. Veja as opções:

```
SW1#show version| ?  
  
append      Append redirected output to URL (URLs supporting append operation  
            only)  
  
begin       Begin with the line that matches  
  
count       Count number of lines which match regexp  
  
exclude     Exclude lines that match  
  
format      Format the output using the specified spec file  
  
include     Include lines that match  
  
redirect   Redirect output to URL  
  
tee        Copy output to URL
```

Pessoalmente, uso com mais frequência o **begin** e **include**. Observe o funcionamento de ambos:

```
SW1#show version | include IOS  
  
Cisco IOS Software, C3560 Software (C3560-IPSERVICESK9-M), Version 12.2(55)SE10,  
RELEASE SOFTWARE (fc2)
```

Ao adicionarmos o ao comando: “| **include IOS**”, só aparecerá as linhas que contém a palavra “IOS”.

Já o Begin iniciará a saída do comando com a palavra que estivermos procurando. Por exemplo, se estivermos interessados apenas nas configurações das interfaces, utilizaremos o comando show running-config | begin interface:

```
SW1#show running-config | begin interface  
  
interface FastEthernet0/1  
  
description CONNECTION_TO_DESKTOP  
  
speed 100  
  
duplex full  
  
!  
  
interface FastEthernet0/2  
  
!  
  
interface FastEthernet0/3  
  
[output omitted]
```

Em vez de ver toda a configuração da running config, ele mostrará o bloco em que começa as configurações das interfaces.

Modo de usuário e modo privilegiado seguros

O que acontece se plugarmos um cabo console em um switch? Por padrão, não será solicitado autenticação para acessarmos o user mode ou privileged mode. Como você pode imaginar, essa opção não é nem um pouco segura. Eis o que acontece ao conectarmos um cabo de console a um switch ou roteador Cisco:

```
Switch con0 is now available
```

```
Press RETURN to get started.
```

```
Switch>
```

Assim que pressionamos o botão Enter, estaremos dentro do modo de usuário. Não é necessário uma senha ou outro tipo de autenticação. A mesma coisa se aplica ao enable mode:

```
Switch>enable
```

```
Switch#
```

Observe, temos acesso total ao dispositivo. Vamos entender como podemos mudar essa situação e assim elevar a segurança dos nossos dispositivos (replique essas configurações no Packet Tracer).

Modo de usuário seguro.

Vamos começar com o modo de usuário.

Autenticação simples

A opção mais simples para proteger o modo de usuário é adicionando uma senha:

```
Switch(config)#line console 0
```

Primeiro, precisamos entrar na configuração da linha de console. Depois, temos que adicionar dois comandos:

```
Switch(config-line)#password cisco
```

```
Switch(config-line)#login
```

Configuramos a senha com a palavra “cisco” e usamos o comando ‘**login**’ para dizer ao sistema operacional que ele deve solicitar esta senha para quem tentar logar no equipamento. Na próxima vez que abrirmos o console, a informação abaixo aparecerá:

```
Switch con0 isnowavailable
```

```
Press RETURN toget started.
```

```
User Access Verification
```

```
Password:
```

```
Switch>
```

Agora, a CLI solicita uma senha. Porém podemos deixar o dispositivo ainda mais seguro, acrescentando uma outra forma de autenticação.

Usuário e senha

Invés de somente exigir uma senha, também é possível fazer que o dispositivo solicite nomes de usuário e senha. Esta é a opção mais recomendada caso haja várias pessoas que acessam o roteador ou switch:

```
Switch(config)#line console 0
Switch(config-line)#login local
Switch(config-line)#exit
Switch(config)#username admin password cisco
```

Nas configurações do console, utilize o comando **login local** para instruir o dispositivo a referenciar um banco de dados local, com username e senha para autenticação. No quadro anterior, em configuração global criamos um nome de usuário “admin” e a senha “cisco”.

Eis o que aparecerá na próxima vez que abrirmos o console:

```
Switch con0 is now available

Press RETURN to get started.

User Access Verification

Username: admin
Password:
Switch>
```

O dispositivo exigirá nome de usuário e senha para completar a conexão.

Modo de Segurança no Enable

É possível e recomendável adicionarmos senha ao modo enable e ao modo privilegiado, para isso basta entrarmos em modo de configuração:

```
Switch#configure terminal
```

E em seguida definirmos a senha que desejamos colocar, no caso abaixo, utilizaremos como senha a palavra ‘cisco’:

```
Switch(config)#enable password cisco
```

Para testar, basta sair do modo enable:

```
Switch#disable
```

E depois retornar:

```
Switch>enable
Password:
```

Observe que agora foi-nos solicitado a senha para obtermos acesso.

Criptografia de senhas

Já temos nossas senhas configuradas, porém há um problema: Todas elas aparecem em texto claro quando verificamos a configuração:

```
Switch#show running-config | include password  
  
no service password-encryption  
  
enable password cisco  
  
username admin password 0 cisco
```

Tudo está em texto claro, fácil de ser identificado. Se alguém obtiver acesso aos nossos switches e roteadores, também terão acesso as senhas. É importante notar, que as senhas também estarão em texto claro quando fizermos backup da configuração.

O sistema operacional da Cisco tem um comando que permite criptografar todas as senhas que estejam sem criptografia na configuração dos dispositivos. Esse comando é o **service password-encryption**:

```
Switch(config)#service password-encryption
```

O comando **service password-encryption**, criptografará todas as senhas que estejam em texto simples. Vejamos o resultado desse comando:

```
Switch#show running-config | include password  
  
service password-encryption  
  
enable password 7 13061E010803  
  
username admin password 7 110A1016141D
```

Um incauto talvez pense que agora todas as senhas estejam protegidas, mas você que está estudando por esse livro, saberá que ainda não! O algoritmo de criptografia utilizado já foi decifrado há muito tempo (quebrado). Existem sites que permitem descriptografar essas strings criptografadas instantaneamente, como esse site abaixo:

<http://password-decrypt.com/>

Inclusive, observe a tela a baixo como foi fácil decodificar a senha que configuramos:

PASSWORD-DECRYPT.com

This page allows you to decrypt Juniper \$9\$ passwords and Cisco 7 passwords.

Juniper :

Enter password and click submit

(password-decrypt.com/juniper.cgi v0.9)

Cisco 7 password:

Decrypted Password: **cisco**

(password-decrypt.com/cisco.cgi v1.0)

Exemplo da senha criptografada **110A1016141D** sendo descriptografada e retornando **cisco**.

Comando Secret

O Cisco IOS oferece um comando chamado secret (segredo) como alternativa ao comando password. Embora a função seja a mesma, o nível de segurança é muito mais alto. Vejamos um exemplo utilizando o modo enable:

```
Switch(config)#enable secret ?  
0      Specifies an UNENCRYPTED password will follow  
5      Specifies a MD5 HASHED secret will follow  
8      Specifies a PBKDF2 HASHED secret will follow  
9      Specifies a SCRYPT HASHED secret will follow  
LINE   The UNENCRYPTED (cleartext) 'enable' secret  
level  Set exec level password
```

Observe, esse switch em específico suporta hash MD5, PBKDF2 e SCRYPT. Dispositivos com imagem IOS mais antigos suportam apenas autenticação MD5.

Vamos configurar:

```
Switch(config)#enable secret cisco
```

Mais uma vez utilizamos a palavra “cisco” como senha, mas dessa vez chamamos de ‘secret’. Ou seja, o secret é a palavra cisco. Vamos retornar a configuração e ver como ela estará:

```
Switch#show running-config | include secret  
enable secret 5 $1$CANW$U9Y806KeFhrFR411Qo07h/
```

Agora encontraremos um hash MD5 na configuração. O “5” que vemos antes do “enable secret” representa o algoritmo que estamos usando, nesse caso, 5 significa MD5.

Porém, MD5 também não é considerado seguro hoje em dia. É muito fácil utilizar técnica de força bruta para descobrir senhas simples. Se quiser tentar, experimente acessar o site abaixo e colar o hash MD5 que foi criamos anteriormente. Descobrir qual é a senha levará apenas alguns segundos:

<https://www.ifm.net.nz/cookbooks/cisco-ios-enable-secret-password-cracker.html>

The screenshot shows a web page titled "Cisco IOS Enable Secret Type 5 Password Cracker". At the top, there is a logo for "ifm Network Experts" and a navigation menu with links for HOME, SOLVE MY PROBLEMS, SERVICES, TOOLS, and CONTACT. Below the title, there is some small text about the service and a note about using client-side JavaScript. The main part of the page contains a form where a user can enter a hashed password (Type 5 Password) and click a "Crack Password" button. There is also a plain text input field labeled "Plain text". At the bottom, there is a note about trying the Cisco-type 7 password cracker instead if you have a Type 7 password.

Exemplo da senha criptografada por MD5 sendo descriptografada e retornando **cisco**.

Vamos utilizar algoritmos que são considerados seguros (ainda não foram quebrados). Observe abaixo quais os algoritmos suportados:

```
Switch(config)#enable algorithm-type ?  
md5      Encode the password using the MD5 algorithm  
scrypt   Encode the password using the SCRYPT hashing algorithm  
sha256   Encode the password using the PBKDF2 hashing algorithm
```

Vamos usar o algoritmo de hash PBKDF2 (SHA256):

```
Switch(config)#enable algorithm-type sha256 secret cisco
```

Quando olhamos novamente a configuração, veremos o novo hash:

```
Switch#show running-config | include secret  
enable secret 8 $8$dvX/fx/FJ0Snk2$HhqrOUaEtBgk4zJvG2IQuAJNUicZmmELelC/L6.Fc12
```

O número “8”, antes do “enable secret”, refere-se ao algoritmo de hash PBKDF2 que usamos.

No exemplo acima, alteramos o algoritmo de hash no enable mode, mas também podemos fazer isso com username:

```
Switch(config)#username Luiz algorithm-type sha256 secret cisco
```

O nome de usuário Luiz, passou a utilizar o SHA256 também para a senha “cisco”:

```
Switch#show running-config | include Luiz  
username Luiz secret 8 $8$dyzsAmZjA3w.aY$YBZn8LBI6CK04ij5ZmqQ/880rFdc3jzGb6v7SSQI0cw
```

OBS: Certifique-se sempre de usar senhas fortes. Não faz diferença o algoritmo de hash que esteja sendo utilizado se as senhas forem fracas como “cisco”, pois elas ainda poderão ser facilmente quebradas.

Servidores de autenticação externa

Aprendemos até aqui a configurar nomes de usuário, senhas, e a utilizar o recurso ‘secret’ nos dispositivos, essa é uma boa prática que você precisa introduzir no seu dia a dia, mas há um problema. Em algum momento, precisaremos de escalabilidade.

Em uma rede com dezenas de dispositivos, teremos que configurar username e senha em todos esses dispositivos, o que seria um extenuante por si só. Mas imagine se necessitarmos alterar uma senha? E se um colaborador novo entrar na empresa e precisar ter acesso aos dispositivos? Não é nada inteligente ter que fazer essa mudança manualmente em cada dispositivo.

Por esse motivo, que em redes maiores, normalmente usamos servidores de autenticação, os populares servidores RADIUS ou TACACS +. Com a utilização de um servidor de autenticação, os usuários e suas senhas são configurados nele de forma centralizada.

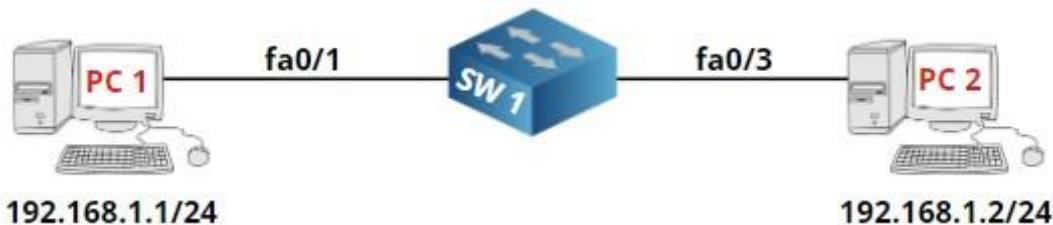
Com a utilização dos servidores de autenticação, quando alguém tentar acessar um dispositivo, seja por cabo console ou remotamente, o dispositivo consultará o servidor de autenticação e não sua base de dados locais. Isso facilita muito a administração da rede.

1.4 Identify interface and cable issues (collisions, errors, mismatch duplex, and/or speed)

Os dispositivos de rede podem ter seu desempenho degradado se houver incompatibilidades na configuração das interfaces ou algum problema no cabeamento. Tanto roteadores quanto switches possuem comandos específicos para ajudar na identificação do problema, que pode ser físico ou de configuração.

Duplex/velocidade

Vamos utilizar a seguinte topologia:



Nesta topologia, temos um switch e dois computadores conectados a ele. Cada computador possui um endereço IP, e estão na mesma rede, ou seja, eles devem ser capazes de ‘pingar’ entre si.

Assumindo que os computadores estão configurados corretamente, vamos tentar realizar um ping:

```
C:\Documents and Settings\PC1>ping 192.168.1.2  
Pinging 192.168.1.2 with 32 bytes of data:  
Request timed out.  
Request timed out.  
Request timed out.  
Request timed out.  
Ping statistics for 192.168.1.2:  
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Os pings não tiveram retorno. Qual é a primeira coisa que devemos verificar? Como as configurações estão corretas, a próxima etapa é verificar as interfaces!

Vamos verificar a interface do fa0/1:

```
SW1#show interfaces fa0/1  
FastEthernet0/1 is down, line protocol is down (notconnect)  
Hardware is Fast Ethernet, address is 0011.bb0b.3603 (bia 0011.bb0b.3603)  
MTU 1900 bytes, BW 100000 Kbit, DLY 100 usec,  
reliability 255/255, txload 1/255, rxload 1/255  
Encapsulation ARPA, loopback not set  
Keepalive set (10 sec)  
Half-duplex, Auto-speed, media type is 10/100BaseTX  
input flow-control is off, output flow-control is unsupported
```

```
ARP type: ARPA, ARP Timeout 04:00:00
Last input 00:26:47, output 00:19:17, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/40 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
3457 packets input, 309301 bytes, 0 no buffer
Received 2407 broadcasts (1702 multicasts)
0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
0 watchdog, 1702 multicast, 0 pause input
0 input packets with dribble condition detected
42700 packets output, 8267872 bytes, 0 underruns
0 output errors, 0 collisions, 1 interface resets
0 babbles, 0 late collision, 0 deferred
0 lost carrier, 0 no carrier, 0 PAUSE output
0 output buffer failures, 0 output buffers swapped out
```

O comando que utilizamos para verificar o funcionamento de uma interface é o: ‘show interfaces’, mas a interface que desejamos verificar, no caso, a interface FastEthernet 0/1.

A saída do comando nos retorna que a interface está down! Ao que tudo indica temos um problema na camada 1, pode ser um cabo quebrado, cabo errado (crossover em vez straight-through), ou até mesmo uma placa de rede com defeito.

Continuando a análise da saída do comando, um pouco mais abaixo, leremos que a interface está sendo executada em ‘**half duplex**’. Se for nosso dia de sorte, receberemos uma mensagem de erro através do protocolo CDP, informando que há incompatibilidade duplex (duplex mismatch). Porém, na maioria dos casos, a interface estará ‘down’ e não receberemos mensagem nenhuma.

Se leremos um pouco mais abaixo, encontraremos que a interface está configurada como half-duplex, velocidade automática (auto speed), porém, interfaces Gigabit não oferecem suporte ao modo half-duplex, logo, essa pode ser uma das possíveis causas da interface estar down. Vamos definir o duplex para automático (duplex auto):

```
SW1(config)#interface fa0/1
SW1(config-if)#duplex auto
```

Quando configuramos uma interface para ‘duplex auto’, o switch negocia por conta própria a forma que ele adotará:

```
SW1#
```

```
%LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up  
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up
```

Logo após a mudança, conforme mostra o log acima, a interface ‘subiu’ e ficou ‘up’. Hora de voltarmos ao PC1 e testarmos o ping novamente.

```
C:\Documents and Settings\PC1>ping 192.168.1.2  
  
Pinging 192.168.1.2 with 32 bytes of data:  
  
Request timed out.  
Request timed out.  
Request timed out.  
  
Ping statistics for 192.168.1.2:  
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Opa, o ping ainda não está funcionando. Vamos verificar a interface do switch que conecta o PC2, novamente utilizando o comando ‘show interfaces’:

```
SW1#show interfaces fa0/3  
  
FastEthernet0/3 is down, line protocol is down (notconnect)  
  
Hardware is Fast Ethernet, address is 0011.bb0b.3605 (bia 0011.bb0b.3605)  
  
MTU 1900 bytes, BW 10000 Kbit, DLY 1000 usec,  
reliability 255/255, txload 1/255, rxload 1/255  
  
Encapsulation ARPA, loopback not set  
  
Keepalive set (10 sec)  
  
Auto-duplex, 10Mb/s, media type is 10/100BaseTX  
  
input flow-control is off, output flow-control is unsupported  
  
ARP type: ARPA, ARP Timeout 04:00:00  
  
Last input 00:38:09, output 00:01:42, output hang never  
  
Last clearing of "show interface" counters never  
  
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0  
  
Queueing strategy: fifo  
  
Output queue: 0/40 (size/max)  
  
5 minute input rate 0 bits/sec, 0 packets/sec
```

```

5 minute output rate 0 bits/sec, 0 packets/sec

    1908 packets input, 181819 bytes, 0 no buffer

        Received 858 broadcasts (826 multicasts)

        0 runts, 0 giants, 0 throttles

        0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored

        0 watchdog, 826 multicast, 0 pause input

        0 input packets with dribble condition detected

    46861 packets output, 9365341 bytes, 0 underruns

        0 output errors, 0 collisions, 1 interface resets

        0 babbles, 0 late collision, 0 deferred

        0 lost carrier, 0 no carrier, 0 PAUSE output

        0 output buffer failures, 0 output buffers swapped out

```

A interface Fa0/3 é a interface do switch que está conectada ao PC2. Observe que ela também está down. Após verificar os cabos e conectores, o próximo passo é verificar possíveis erros de duplex e velocidade.

O Duplex já está configurado em modo automático, podemos concluir que não deve ser este o problema. No entanto, a velocidade foi definida para 10Mbit, embora a interface seja um link FastEthernet (100Mbit). Provavelmente esta é a causa do problema, vamos configura-la para auto:

```

SW1(config)#interface fa0/3
SW1(config-if)#speed auto

```

Aguardamos alguns segundos para ver se houve alguma mudança, e essa é a mensagem que aparece na tela:

```

SW1#
%LINK-3-UPDOWN: Interface FastEthernet0/3, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/3, changed state to up

```

Mais uma vez, o problema estava na incompatibilidade de velocidade (speed mismatch). A diferença na velocidade causou a queda da interface. Ao configurarmos a velocidade como auto, a interface retornou para o status up (esses problemas embora simples, são os mais usuais no dia a dia). Vamos verificar se todas as interfaces estão ativas:

SW1#show ip interface brief					
Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/1	unassigned	YES	unset	up	up
FastEthernet0/3	unassigned	YES	unset	up	up

Observe a saída. As interfaces que estamos utilizando, estão **up/up**. Podemos afirmar, que nesse momento, não há erros de cabo, velocidade ou duplex. Hora de testarmos o ping novamente:

```
C:Documents and SettingsPC1>ping 192.168.1.2
```

```

Pinging 192.168.1.2 with 32 bytes of data:

Reply from 192.168.1.2: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

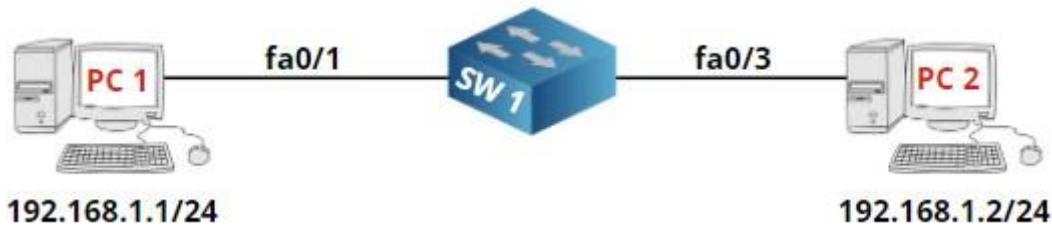
```

Como era esperado, agora nosso ping funcionou e obteve resposta.

Lição aprendida: Devemos sempre verificar inconsistência nas configurações de velocidade e duplex quando as interfaces não estejam com status up/up.

Violações de segurança nas interfaces

Vamos utilizar a mesma topologia anterior para entender as violações de segurança e como elas podem afetar a conectividade.



Vimos que a topologia e o endereço IP das máquinas são os mesmos, hora de testar o ping:

```

C:\Documents and Settings\PC1>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

Request timed out.
Request timed out.

Ping statistics for 192.168.1.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

```

O PC1 não consegue ‘pingar’ o PC2. O primeiro passo, como vimos anteriormente é verificar as interfaces:

```
SW1#show ip interface brief
```

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/1	unassigned	YES	unset	down	down
FastEthernet0/3	unassigned	YES	unset	up	up

A interface FastEthernet 0/3 está Up/Up, porém, algo está errado com a interface FastEthernet 0/1. Vamos verificar:

```
SW1#show interfaces fa0/1
```

```
FastEthernet0/1 is down, line protocol is down (err-disabled)
```

Encontramos uma mensagem de ‘err-disabled’ e confirmamos que a interface está realmente down. Vamos entender o que gerou essa mensagem com o comando abaixo:

```
SW1#show interfaces status err-disabled
```

Port	Name	Status	Reason	Err-disabled Vlans
Fa0/1		err-disabled	psecure-violation	

Com o comando “**show interfaces status err-disabled**”, verificamos que a interface está desabilitada, pois houve uma violação de segurança. Vamos aprofundar um pouco mais e descobrir qual violação foi essa:

```
SW1#show port-security interface fa0/1
```

```
Port Security : Disabled
```

```
Port Status : Secure-shutdown
```

```
Violation Mode : Shutdown
```

```
Aging Time : 0 mins
```

```
Aging Type : Absolute
```

```
SecureStatic Address Aging : Disabled
```

```
Maximum MAC Addresses : 1
```

```
Total MAC Addresses : 1
```

```
Configured MAC Addresses : 1
```

```
Sticky MAC Addresses : 0
```

```
Last Source Address:Vlan : 000c.2928.5c6c:1
```

```
Security Violation Count : 1
```

Analisando a configuração de segurança da interface, é possível verificar que apenas 1 endereço MAC é permitido (endereço da placa de rede). O último endereço MAC que tentou conectar na interface foi o 000c.2928.5c6c. Vamos ver se esse endereço MAC é o mesmo que foi configurado como permitido nas configurações de segurança de porta:

```
SW1#show port-security interface fa0/1 address
```

```
Secure Mac Address Table
```

Vlan	Mac Address	Type	Ports	Remaining Age (mins)
1	0019.569d.5742	Secure Configured	Fa0/1	-
Total Addresses: 1				

O endereço MAC autorizado nessa interface é diferente do que estava tentando se conectar. Esta é a razão pela qual a porta entrou em erro e foi desabilitada. Podemos corrigir isso alterando o endereço MAC ou desativando o modo de segurança da porta (port-security). Vamos retirar o comando port-security e corrigir o problema.

```
SW1(config)#interface fa0/1
SW1(config-if)#no switchport port-security
```

Para retirar o erro e consequentemente voltarmos com a interface para o status up\up basta aplicarmos os comandos abaixo:

```
SW1(config)#interface fa0/1
SW1(config-if)#shutdown
SW1(config-if)#no shutdown
```

A seguinte mensagem aparecerá na tela:

```
SW1#
%LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up
```

Agora, temos a mensagem informando que a interface agora está up\up. Vamos tentar o ping novamente:

```
C:Documents and SettingsPC1>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

Reply from 192.168.1.2: bytes=32 time<1ms TTL=128
Reply from 192.168.1.2: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

Approximate round trip times in milli-seconds:

Minimum = 0ms, Maximum = 0ms, Average = 0ms

Lição aprendida: Verifique sempre se a interface está em **err-disabled**, e em caso afirmativo:

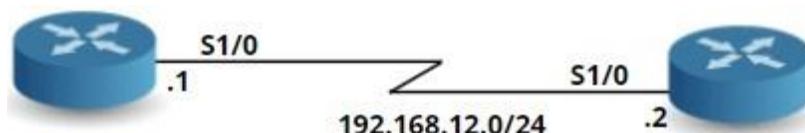
1. verifique por que isso aconteceu; e
2. resolva o problema.

OBS: Não ver err-disabled não significa automaticamente que não há problemas de segurança na interface. O modo de violação padrão para a port-security é o shutdown da interface colocando em err-disabled. Porém, há outros modos, por exemplo: O modo de restrição (restrict mode), que manterá a interface ativa, mas mostrará uma mensagem de log no console. Já o modo de proteção (Protect mode) também mantém a interface ativa, mas não mostra nenhuma mensagem no console.

Consiste em uma boa prática sempre verificar se o security-port está habilitado ou não, assim como usar o comando show mac address-table para verificar se o switch aprendeu os endereços MAC dos dispositivos conectados as suas interfaces.

Incompatibilidade de protocolos

Vamos trabalhar com uma topologia diferente:



Todas as interfaces estão com endereços IPs corretamente configurados, porém, mesmo assim não conseguimos realizar o ping:

```
R1#ping 192.168.12.2
Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.12.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

O primeiro passo é verificar o status das interfaces:

```
R1#show ip interface brief | include 1/0
Serial1/0          192.168.12.1    YES manual up      down
```

```
R2#show ip interface brief | include 1/0
Serial1/0          192.168.12.2    YES manual up      down
```

O primeiro status “UP” indica que a parte física está ativa, o que indica que o clock (interface serial exige que configuremos clock) foi configurado, e que o cabo está funcionando. Porém, o protocolo está inativo. Nessa situação o melhor a fazer é verificar os detalhes das interfaces:

```
R1#show interfaces Serial 1/0
Serial1/0 is up, line protocol is down
Hardware is M4T
Internet address is 192.168.12.1/24
```

```
MTU 1500 bytes, BW 1544 Kbit/sec, DLY 20000 usec,  
reliability 255/255, txload 1/255, rxload 1/255  
Encapsulation HDLC, crc 16, loopback not set
```

```
R2#show interfaces Serial 1/0  
  
Serial1/0 is up, line protocol is down  
  
Hardware is M4T  
  
Internet address is 192.168.12.2/24  
  
MTU 1500 bytes, BW 1544 Kbit/sec, DLY 20000 usec,  
reliability 255/255, txload 1/255, rxload 1/255  
  
Encapsulation PPP, LCP Listen, crc 16, loopback not set
```

Observe atentamente as duas saídas acima, observe que existe uma incompatibilidade no encapsulamento. R1 está usando o protocolo HDLC enquanto o roteador R2 está usando o protocolo PPP. Vamos consertar:

```
R1(config)#interface Serial 1/0  
R1(config-if)#encapsulation ppp
```

Após alterarmos o protocolo L2, o link funcionará normalmente:

```
R1# %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0, changed state to up
```

```
R1#ping 192.168.12.2  
  
Type escape sequence to abort.  
  
Sending 5, 100-byte ICMP Echos to 192.168.12.2, timeout is 2 seconds:  
!!!!!  
  
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/21/40 ms
```

Lição aprendida: Certifique-se sempre de estar usando os mesmos protocolos de encapsulamento em ambas as pontas.

Dica: A maioria dos problemas relacionados à interface, podem ser resolvidos facilmente verificando e comparando a saída do comando “show interfaces”.

1.5 Compare TCP to UDP

Neste tópico, estudaremos os protocolos de transporte. Os mais populares são o **TCP** e **UDP**. O protocolo IP exige que haja um protocolo de transporte para que os pacotes sejam enviados.

Vamos nos concentrar nesses dois protocolos de transporte que são cobrados no exame CCNA:

- **TCP (Transmission Control Protocol)**
- **UDP (User Datagram Protocol)**

Provavelmente você está se perguntando: Por que temos 2 protocolos de transporte diferentes? Qual a diferença que nos faz escolher um em determinado momento e não o outro? São essas dúvidas que responderei a seguir.

A resposta mais curta e assertiva que posso dar é:

- **TCP** é um protocolo **confiável**.
- **UDP** é um protocolo **não confiável** ou de **melhor esforço (best-effort)**.

Mais uma vez você deve estar se perguntando: Por que alguém usaria um protocolo de transporte de dados que não seja confiável? Aparentemente, não faz sentido nenhum! Porém, há sentido em tudo isso. Leia a história a seguir para entender a diferença exata entre os dois protocolos e porque em determinado momento é melhor usar um que o outro.

Sexta à noite, depois de uma semana cansativa de trabalho, você chega em casa, liga o computador, conecta na entrada HDMI da televisão, abre seu programa preferido de torrent e começa a baixar o filme mais recente da Marvel, detalhe importante, em 4k. O arquivo com o filme possui 60GB, porém, depois de baixar 55GB, alguma coisa acontece e alguns pacotes IPs não chegam até seu computador. Assim que o download acaba, você tenta reproduzir o filme, mas, infelizmente, o arquivo está com erro, o que impossibilita você de assistir o filme até o final. Frustrado, só lhe resta abrir a Netflix e assistir qualquer um daqueles filmes que já passaram milhares de vezes na sessão da tarde.

Talvez eu tenha exagerado um pouco na história, mas acho que deu para entender o espírito da coisa. Em atividades assim, você quer ter certeza que o transporte dos dados do servidor até o seu computador seja **confiável**, por isso, nesse caso, faz todo sentido usarmos TCP. Afinal, caso alguns pacotes IP não cheguem no seu computador, o TCP providenciará que esses dados sejam retransmitidos!

Agora uma segunda história para clarificar ainda mais a importância de cada protocolo: Você é engenheiro de rede em uma grande empresa, responsável por toda parte de comunicação. Em uma reunião, você apresentou aos diretores uma nova solução de voz sobre IP de código aberto, é uma solução de custo baixo e que pode atender perfeitamente toda a demanda da empresa. Com aval da direção você decide implementar esta nova solução VoIP, e assim se livrar de todos os telefones analógicos. Porém, os usuários começaram a reclamar, e muito, que a qualidade das ligações telefônicas estão horríveis.

Desesperado, você abre um incidente com a provedora da solução VoIP, que lhe vendeu a tecnologia, e em contato com o suporte deles, descobre que o analista responsável pela implementação na sua empresa pensou que seria uma boa ideia usar um protocolo de transporte **confiável** como o TCP, já que queremos que as chamadas telefônicas sejam confiáveis, certo?

Embora pareça lógico, esse pensamento é totalmente errado, e possivelmente seja ele a causa de todo o transtorno! O TCP faz correção de erros, o que significa que os dados que não chegaram ao destino serão retransmitidos.

Imagine o quanto estranho uma ligação soará se você estiver falando com alguém e ouvir algo que essa pessoa disse há alguns segundos no meio de outra frase? Esse tipo de conexão é realizado em tempo real, então não queremos retransmissão. É melhor perder alguns pacotes VoIP do que retransmitir esses pacotes alguns milésimos de segundos depois. Além do mais, o codec VoIP pode corrigir a perda de pacotes até um certo grau. Neste caso, é preferível usar um protocolo de **melhor esforço** ou protocolo **não confiável**, no caso o UDP.

	TCP	UDP
Tipo de conexão:	Coneção Orientada	Sem conexão
Sequenciamento:	Sim	Não
Uso:	Transferências	VoIP
	Compartilhamento de arquivos	Vídeo (Streaming)
	Impressão	

Vamos analisar a tabela acima:

Na primeira linha temos: **Tipos de conexão**.

- O protocolo **TCP** é **orientado à conexão**, o que significa que ele “**construirá**” uma conexão e, em seguida, começará a transferir os dados.
- O protocolo **UDP** não tem conexão, o que significa que ele **apenas enviará os dados**, sem se importar se estes dados chegarão ou não até o destino.

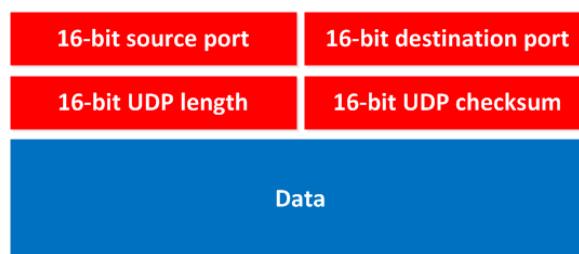
O TCP ‘constrói’ a conexão através do processo chamado de “**3 way handshake**” ou “**aperto de mãos de 3 vias**”, veremos sobre seu funcionamento em seguida.

Na segunda linha temos: **Sequenciamento**, que significa que é utilizado um **número de sequência** para organizar os pacotes de dados. Por exemplo, quando baixamos um arquivo grande, precisamos de um mecanismo que garanta que os pacotes serão reorganizados na sequencia correta. Como podemos ver na tabela, o UDP não oferece esse recurso, simplesmente porque ele não se preocupa com isso.

Já sei o que você está pensando, e quanto ao VoIP? Não precisamos colocar esses pacotes de volta em ordem, no lado do receptor? Essa é uma boa pergunta, e a resposta é claro, nós precisamos colocar esses pacotes na ordem certa, ou do contrário teríamos conversas sem sentido. No entanto, como dito anteriormente, o UDP não oferece esse recurso de “sequenciamento”. Porém, quando usamos o VoIP, não estamos utilizando apenas o protocolo UDP, junto a ele estamos usando um protocolo importantíssimo chamado RTP, e esse sim oferece o recurso de sequenciamento, junto com alguns outros recursos interessantes que o VoIP utiliza.

Protocolo UDP

Vamos analisar o cabeçalho UDP:



Cabeçalho UDP.

O cabeçalho UDP é bem simples, temos apenas o número da porta de origem e de destino (source e destination port number), é através das portas que podemos identificar para qual aplicação os dados se destinam, há também um campo de soma de verificação (checksum) e comprimento (length).

Vamos resumir o que já aprendemos sobre o UDP:

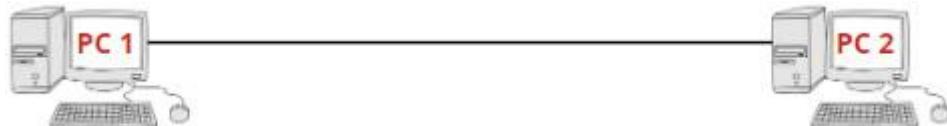
- Opera na camada de transporte do modelo OSI.
- É um protocolo sem conexão, ou seja, não estabelece uma conexão antes de enviar os dados.
- Correção de erros limitada, por isso, utiliza o checksum.
- Pode-se classificá-lo como um protocolo de **melhor esforço (best-effort)** e também como protocolo **não confiável (unreliable)**.
- Não oferece nenhum recurso para recuperação de dados.

Protocolo TCP

Hora de verificarmos o protocolo TCP.

O primeiro ponto é entender que o TCP é um protocolo confiável, sendo assim, ele “estabelecerá” uma conexão antes de começar a enviar quaisquer dados. Esta conexão é estabelecida através do “**3 way handshake**”. E como prometido anteriormente, é hora de explicar como o “**3 way handshake**” funciona.

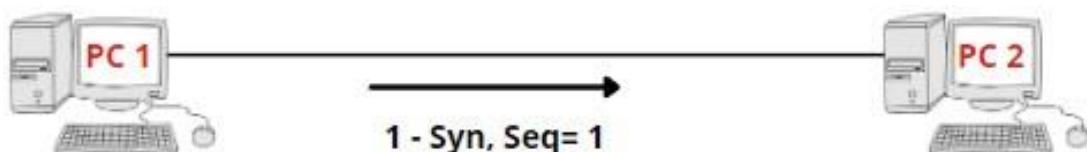
A melhor forma de explicar o “**3 way handshake**” é detalhando o processo entre dois computadores que desejam enviar dados um para o outro de maneira confiável:



PC1 deseja enviar dados para o PC2 de maneira confiável, como você já decorou, quando trata-se de enviar dados de maneira confiável estamos falando do protocolo TCP.

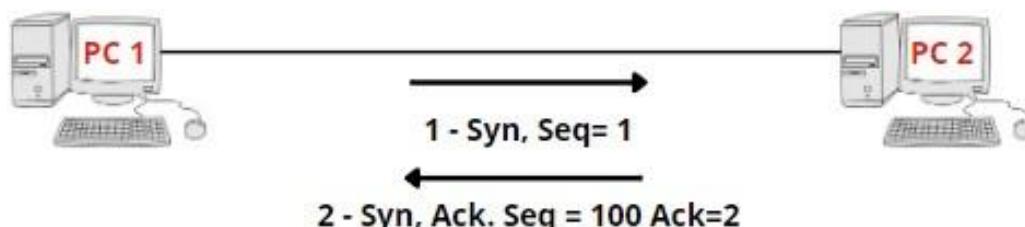
Primeiro, é estabelecido uma conexão usando o **3 way handshake**:

PC1 que está iniciando a comunicação envia um **TCP SYN**, informando ao PC2 que deseja estabelecer uma conexão. Anexo a essa mensagem também há um número de sequência, para simplificar, utilizaremos o número 1.

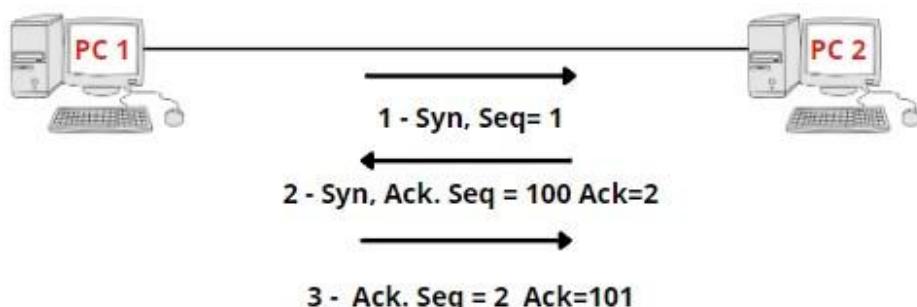


O PC2 responderá ao PC1 enviando uma mensagem **SYN, ACK**. Observe, que o PC2 escolheu seu próprio número de sequência, no caso, 100 (Esse é um número aleatório) e envia **ACK = 2**.

ACK = 2 significa que ele está reconhecendo que recebeu o **TCP SYN** do PC1, que tinha o número de sequência 1, e que está pronto para a próxima mensagem com o número de sequência 2.



Agora partimos para a última etapa, o PC1 enviará uma **confirmação (acknowledgement)** para PC2 em resposta ao **SYN** que PC2 enviou para PC1. Observe que ele enviou **ACK = 101**, o que significa que ele reconheceu **SEQ=100** do PC2. Como PC2 enviou um **ACK=2** para PC1, o computador 01 sabe que pode enviar a próxima mensagem com o número de sequência 2.



Resumindo e simplificando todo o processo:

- PC1 envia um **TCP SYN** para o PC2, o que traduzindo para o bom português seria algo como: - PC2, quero falar com você!
- PC2 envia um **TCP SYN, ACK** para o PC1, o que traduzindo para o bom português seria algo como: - Ok PC1, Eu aceito falar com você, e quero que você confirme que também quer falar comigo!
- PC1 envia um **TCP ACK**. Que seria algo como: - Ok, PC2, Eu confirmo que você quer falar comigo.

Observe no **Wireshark** como esse processo ocorre em uma rede real:

1 0.000000	192.168.1.2	174.143.213.184	TCP	54841 > http [SYN] Seq=0 Win=5840 L
2 0.046770	174.143.213.184	192.168.1.2	TCP	http > 54841 [SYN, ACK] Seq=0 Ack=1
3 0.046803	192.168.1.2	174.143.213.184	TCP	54841 > http [ACK] Seq=1 Ack=1 Win=

1. No caso em tela, o computador com endereço IP 192.168.1.2 deseja estabelecer uma conexão com o host 174.143.213.184, para isso ele está enviando um **TCP SYN**.
2. O Host 174.143.213.184 está respondendo enviando um **TCP SYN, ACK**.
3. Finalmente, 192.168.1.2 envia um **TCP ACK** para finalizar o **3 way handshake**.

Vamos aprofundar um pouco mais e ver esses pacotes em detalhe, primeiro olhamos o TCP SYN:

Observe na seção “**Flags**” que o **SYN-bit** foi definido. No canto superior direito há o número de sequência escolhido, no caso “**Seq: 0**”.

```
▼ Transmission Control Protocol, Src Port: 54841 (54841), Dst Port: http (80), Seq: 0, Len: 0
  Source port: 54841 (54841)
  Destination port: http (80)
  [Stream index: 0]
  Sequence number: 0      (relative sequence number)
  Header length: 40 bytes
  ▶ Flags: 0x02 (SYN)
  Window size: 5840
  ▶ Checksum: 0x85f0 [validation disabled]
  ▶ Options: (20 bytes)
```

Abaixo, é possível ver na seção “**flags**” que os bits **SYN** e **ACK** estão definidos, na parte superior também é possível ver “**Seq: 0**” e “**Ack: 1**”. O que significa que este computador está reconhecendo o **SYN-bit** do outro computador.

```
▼ Transmission Control Protocol, Src Port: http (80), Dst Port: 54841 (54841), Seq: 0, Ack: 1, Len: 0
  Source port: http (80)
  Destination port: 54841 (54841)
  [Stream index: 0]
  Sequence number: 0      (relative sequence number)
  Acknowledgement number: 1    (relative ack number)
  Header length: 40 bytes
  ▶ Flags: 0x12 (SYN, ACK)
```

Chegamos na etapa final do processo, em que o computador que iniciou o **3 way handshake** define o **ACK-bit** e reconhece o **SYN** do outro computador.

```
▼ Transmission Control Protocol, Src Port: 54841 (54841), Dst Port: http (80), Seq: 1, Ack: 1, Len: 0
  Source port: 54841 (54841)
  Destination port: http (80)
  [Stream index: 0]
  Sequence number: 1      (relative sequence number)
  Acknowledgement number: 1    (relative ack number)
  Header length: 32 bytes
  ▶ Flags: 0x10 (ACK)
  Window size: 5888 (scaled)
  ▶ Checksum: 0x9529 [validation disabled]
  ▶ Options: (12 bytes)
  ▶ [SEQ/ACK analysis]
```

Conhecer o funcionamento nessa profundidade pode ser um pouco maçante no início, mas acredite, depois que você aprende os conceitos básicos, passa a ser gostoso de estudar, além de dar uma perspectiva muito maior.

Caso você tenha gostado e deseja brincar um pouco, inicie o Wireshark e tente capturar um **3 way handshake** em seu computador. Dê uma olhada nos diferentes pacotes TCP e veja se consegue encontrar **SYN**, **SYN-ACK** e **ACKs**. Verifique também os diferentes números de sequência e veja se consegue encontrar um padrão.

Bom, acabamos de estabelecer uma conexão **3 way handshake!** Os dados já podem ser enviados. É uma boa hora de aprender outra ferramenta importante que o TCP nos oferece: **Controle de fluxo ou flow control**.

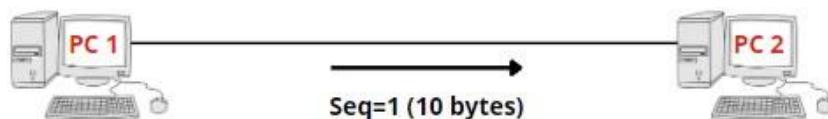
Você já deve ter reparado que eu gosto de explicar os conceitos através de exemplos, mostrando o funcionamento na prática, isso torna assimilação do novo conteúdo mais palpável e fácil de relacionar. Pedagogicamente é infinitamente melhor.

Imagine que você possui um computador de última geração, super rápido, e deseja transmitir dados para um smartphone. Obviamente, o computador poderia sobrecarregar o smartphone com tráfego, afinal a diferença de velocidade dos componentes é imensa, por isso o TCP utiliza o **controle de fluxo**. Em cada segmento TCP, o receptor pode especificar no campo “**receive window**” (janela de recepção) quantos bytes de dados ele deseja receber.

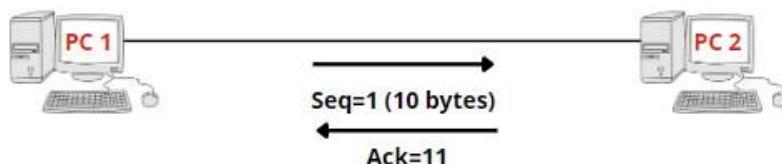
O computador remetente só pode enviar dados para o smartphone até o tamanho definido no **receive window**, assim o smartphone não ficará sobrecarregado. É fácil deduzir que quanto mais dados pudermos enviar de uma vez, maior será a taxa de transmissão.

Vamos ilustrar tudo que vimos até agora com um passo a passo:

Vamos usar a mesma topologia do exemplo anterior, em que o PC1 estabeleceu uma conexão com PC2 usando o **3 way handshake**. No momento, ele está enviando 10 bytes de dados, o que significa que o “**tamanho da janela**” (**window size**) é de 10 bytes, e o número da sequência é 1.

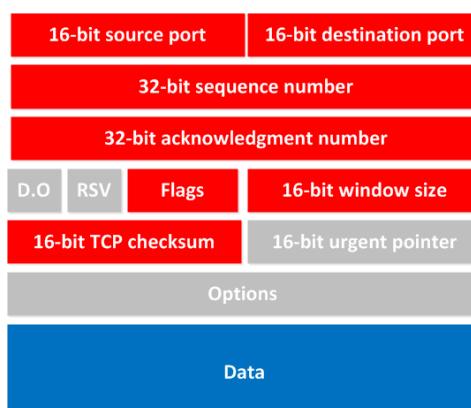


O PC2 irá responder enviando “**ACK=11**”, o que significa: “Obrigado, recebi seus 10 bytes, agora me envie o resto”. O **TCP é um protocolo confiável**, por isso, ele sempre envia um **ACK** confirmado tudo que recebeu.



Quanto maior o tamanho da janela, maior será o throughput (taxa de transmissão). Isso faz sentido porque estaremos enviando menos **ACKs**, além de enviarmos mais dados ao mesmo tempo.

O TCP é um protocolo complexo, quando olhamos o cabeçalho vemos muito mais campos que o cabeçalho UDP:



Os campos em cinza não são importantes nesse momento; vamos focar nos campos em destacados.

Na primeira linha, há dois campos, um com a porta de origem e outro com a porta de destino, ambos os campos possuem 16 bits. Esses dois campos são muito importantes, pois, são os **números da porta que determinam qual aplicativo\aplicações** esses dados se destinam.

Na segunda linha, temos um campo de 32 bits, esses bits são usados para formar os números de sequência, e na terceira linha temos outro campo com 32 bits, esses bits são utilizados para o acknowledgment (ACK).

O campo “Flags”, é o campo onde o TCP define os diferentes tipos de mensagens, como “SYN” ou “ACK”.

No campo ao lado do ‘flags’, temos o campo Window size, com espaço de 16 bits, é nesse campo que o TCP especifica quantos bytes de dados enviará.

Por último, há o campo de checksum, neste campo ocorre a soma de verificação para garantir que o pacote está integral. Logo abaixo, temos o campo ‘Data’, que é o campo que carrega os dados que desejamos enviar para outros dispositivos.

Tal como no UDP, vale a pena fazermos um pequeno resumo sobre o que aprendemos do TCP:

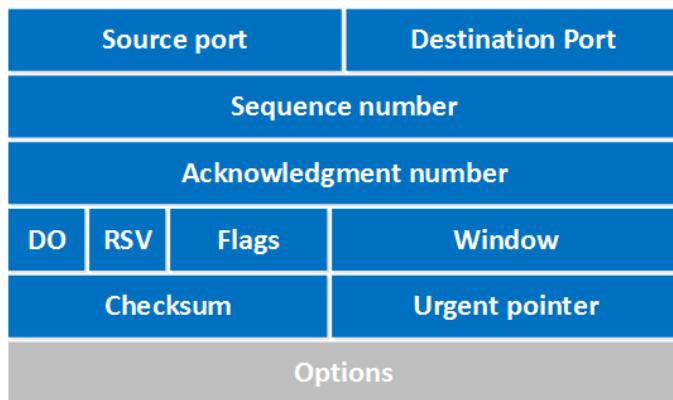
- É um protocolo confiável.
- Antes de enviar dados, ele estabelece uma conexão usando o **3 way handshake**.
- Depois de enviar uma quantidade X de bytes, ele recebe uma confirmação (ACK) do receptor.
- A quantidade de bytes que é enviado antes de obter um ACK, é controlado pelo campo “window size”.
- TCP pode fazer retransmissões dos dados caso haja necessidade.

Cabeçalho TCP

Embora tenhamos visto um pouco sobre o cabeçalho TCP (Transmission Control Protocol) anteriormente, acho importante aprofundar um pouco mais. Mas, ressalto, tudo que você precisa para o CCNA está no tópico anterior.

O TCP é um protocolo de transporte confiável, pois estabelece uma conexão antes de enviar qualquer dado, além disso, tudo que ele envia é confirmado pelo receptor. Vamos examinar novamente o cabeçalho TCP e todos os seus campos.

Este é o esquema de um cabeçalho TCP:



Vamos percorrer todos esses campos, detalhando o máximo possível cada componente:

1. **Source Port:** Este campo possui 16 bits, a função dele é especificar o número da porta do remetente.
2. **Destination Port:** Este campo também possui 16 bits, a função dele é especificar o número da porta do receptor.
3. **Sequence Number:** O número de sequência é um campo de 32 bits que indica a quantidade de dados que já foram enviados durante a sessão TCP. Quando você estabelece uma nova conexão TCP (3 way handshake), o número de sequência inicial é um valor aleatório de 32 bits. O receptor usará este número de sequência e enviará de volta uma

- confirmação. Analisadores de protocolo, como o Wireshark, geralmente usam um *número de sequência relativo* de 0, pois é mais fácil para ler que um número aleatório alto.
4. **Acknowledgment number:** Este campo de 32 bits é usado pelo receptor para solicitar o próximo segmento TCP. O valor será o número de sequência incrementado em 1.
 5. **DO:** É o campo de deslocamento de dados, possui 4 bits. Ele também é conhecido como ‘header length’ (comprimento do cabeçalho). Possui a função de indicar o comprimento do cabeçalho TCP, dessa forma é possível identificar onde os dados realmente começam separando assim o que é cabeçalho e o que é dado.
 6. **RSV:** É um campo composto por 3 bits reservados. Ele não é utilizado e os valores são sempre definidos como 0.
 7. **Flags:** Existem 9 bits para as flags (sinalizadores), também são conhecidos como bits de controle. Eles são usados para estabelecer conexões, enviar dados e encerrar conexões:

- 7.1 **URG:** Este é o ponteiro urgente. Quando este bit está definido, os dados devem ser tratados como prioridade sobre outros dados.
- 7.2 **ACK:** Utilizado para acknowledgment (reconhecimento).
- 7.3 **PSH:** Esta é a função push. Ela informa ao aplicativo que os dados devem ser transmitidos imediatamente, que ele deve esperar para preencher todo o segmento TCP.
- 1.4 **RST:** Reinicia a conexão, quando o receptor recebe esta flag, ele encerra a conexão imediatamente. Só é usado quando há erros irrecuperáveis, definitivamente, não é uma maneira normal de finalizar a conexão TCP.
- 1.5 **SYN:** Utilizado no início do processo de 3 way handshake, é usado para definir o número de sequência inicial.
- 1.6 **FIN:** O bit de encerramento é usado para encerrar a conexão TCP. O TCP é full duplex, então ambas as partes terão que usar o bit FIN para encerrar a conexão. Este é o método normal utilizado para encerrar uma conexão.
8. **Window:** Este campo possui 16 bits, e serve para especificar quantos bytes o receptor está disposto a receber. É usado para que o receptor possa dizer ao remetente que gostaria de receber mais ou menos dados do que está recebendo no momento.
9. **Checksum:** Esse campo possui 16 bits, e é utilizado para realização da soma de verificação, onde através de cálculos matemáticos o TCP verifica se o pacote está integral ou não.
10. **Urgent pointer:** São 16 bits que somente são utilizados quando a ‘flag’ com o bit URG está definida, o ‘urgent pointer’ é usado para indicar onde os dados urgentes terminam.
11. **Options:** Este campo é opcional e possui um tamanho variável de 0 e 320 bits.

A melhor forma de ver tudo isso junto e funcionando é através do Wireshark. Observe abaixo, os campos destacados, eles compõe a primeira parte do 3 way handshake:

```

Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
Ethernet II, Src: C2:01:0c:b4:00:00 (C2:01:0c:b4:00:00), Dst: C2:02:13:98:00:00 (C2:02:13:98:00:00)
Internet Protocol Version 4, Src: 192.168.12.1 (192.168.12.1), Dst: 192.168.12.2 (192.168.12.2)
Transmission Control Protocol, Src Port: 41417 (41417), Dst Port: 23 (23), Seq: 0, Len: 0
Source Port: 41417 (41417)
Destination Port: 23 (23)
[Stream index: 0]
[TCP Segment Len: 0]
Sequence number: 0 (relative sequence number)
Acknowledgment number: 0
Header Length: 24 bytes
... 0000 0000 0010 = Flags: 0x002 (SYN)
    ... . .... = Reserved: Not set
    ... 0 .... = Nonce: Not set
    ... 0.... = Congestion Window Reduced (CWR): Not set
    ... 0.. = ECN-Echo: Not set
    ... 0... = Urgent: Not set
    ... 0... = Acknowledgment: Not set
    ... 0... = Push: Not set
    ... 0... = Reset: Not set
    ... 0... = SYN: Set
    ... 0... = Fin: Not set
    window_size_value: 4128
    [calculated window size: 4128]
    checksum: 0xe46a [validation disabled]
        [Good Checksum: False]
        [Bad Checksum: False]
    urgent_pointer: 0
Options: (4 bytes), Maximum segment size
    Maximum segment size: 1460 bytes
    Kind: Maximum Segment Size (2)
    Length: 4
    MSS Value: 1460

```

Observe os campos marcados: O campo ‘destination port’ indica que estamos conectando na porta 23, que é a porta telnet. O número de sequência é 0, mas o wireshark nos diz que este é um número de sequência relativo, ou seja, não reflete a realidade.

Observe que o bit-SYN foi definido nas flags, também é possível ver o tamanho da janela, a soma de verificação, o ponteiro urgente e as opções (traduzi a maioria dos campos para melhor compreensão, mas acostume-se com o nome em inglês)

TCP Window Size Scaling

Podemos traduzir ‘TCP Window Size Scaling’ como ‘tamanho da janela flutuante’, mas, mais importante que a tradução simples ao pé da letra fria, é entendermos o conceito, e é isso que faremos agora.

Como dito anteriormente, o TCP (Transmission Control Protocol) é um protocolo orientado à conexão, o que significa que dentre outros recursos, ele também controla a quantidade de dados transmitidos. Quando o dispositivo remetente transmite dados para o host receptor, este deve confirmar o recebimento de cada um desses dados, quando o remetente não recebe a confirmação dentro de um tempo específico, os dados são retransmitidos.

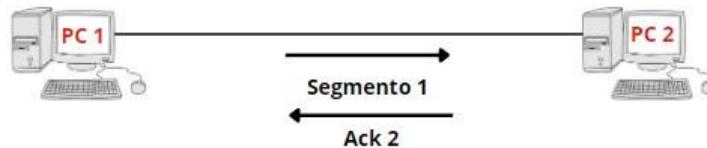
O protocolo TCP usa “janelas”, o que significa que, o remetente enviará um ou mais segmentos de dados, e o receptor reconhecerá um ou todos esses segmentos. Quando iniciamos uma conexão TCP, os hosts usam um buffer para recepção, onde os dados são armazenados temporariamente até que o aplicativo possa processá-los.

Quando o receptor envia uma confirmação, ele está informando ao remetente quantos dados ele consegue lidar, e dessa forma o remetente sabe a quantidade de dados que ele pode transmitir antes que o receptor envie sua próxima confirmação. Esse processo é chamado de **Window Size Scaling** ou **tamanho da janela flutuante**. Basicamente, o tamanho da janela indica o tamanho do buffer de recepção.

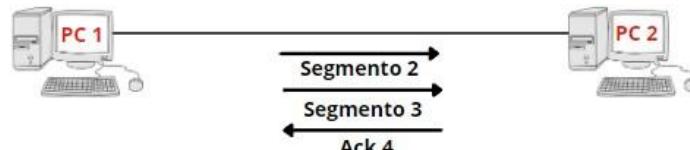
Normalmente, a conexão TCP começa com um tamanho de janela pequeno, e sempre que houver uma confirmação (acknowledgment) informando que a transmissão foi bem-sucedida, o tamanho da janela aumentará.

Eu sei, é um conceito um tanto quanto confuso, mas um exemplo com ilustrações do funcionamento passo a passo sempre torna as coisas mais fáceis. Eis o exemplo:

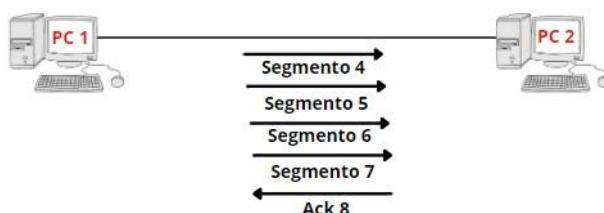
Trabalharemos com a mesma topologia dos exemplos anteriores, 02 computadores que desejam transmitir dados de maneira confiável. O computador do lado esquerdo enviará um segmento para o host do lado direito, este, enviará uma ‘acknowledgment’ de recebimento. Sempre que houver esse ‘acknowledgment’, informando que a transmissão foi bem sucedida, o tamanho das janelas aumentará:



O host do lado esquerdo, enviará dois segmentos e o host do lado direito retornará uma única confirmação. Por enquanto, tudo está funcionando bem, então o tamanho da janela aumentará ainda mais:



O host PC1, enviará quatro segmentos e o host do lado direito responderá com uma única confirmação.



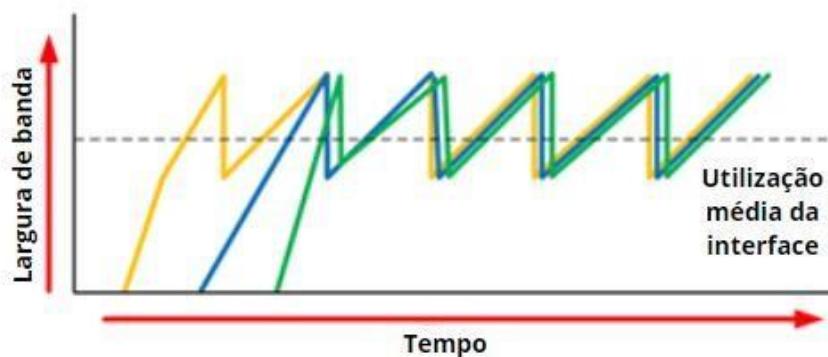
No exemplo acima, o tamanho da janela continuará aumentando enquanto o receptor enviar ‘acknowledgment’ para todos os segmentos, ou quando o tamanho da janela atingir um certo limite. Quando o receptor não enviar um ‘acknowledgment’ dentro de um determinado período de tempo (chamamos esse período de tempo de **round-trip time**, algo como, tempo de ida e volta), o tamanho da janela será reduzido.

Outro recurso interessante que o TCP possui é a forma que ele é capaz de lidar com interfaces congestionadas. Quando há congestionamento em uma interface, é possível que os pacotes IP sejam descartados. Para lidar com isso, o TCP tem vários algoritmos que atuam no controle de congestionamento. Um desses algoritmos é o ‘slow start’ (início lento).

O congestionamento ocorre quando a interface precisa transmitir mais dados do que pode suportar. Isso provoca uma fila, e quando essa fila atingir um certo limite, pacotes serão descartados.

Com o ‘slow start’, o tamanho da janela aumentará exponencialmente (o tamanho da janela dobra), porém, uma vez que um pacote é descartado, o tamanho da janela é reduzido para apenas um segmento. Ele então crescerá novamente de forma exponencial, até que o tamanho da janela seja a metade do que era quando ocorreu o congestionamento. Nesse momento, o tamanho da janela aumentará linearmente e não mais de forma exponencial.

Quando uma interface fica congestionada, é possível que todas as conexões TCP ativas no momento, recebam o tratamento do ‘slow start’. Ou seja, os pacotes serão descartados e todas as conexões TCP terão uma janela de tamanho reduzido. Esse processo é chamado de **‘TCP global synchronization’** (sincronização global do TCP). Vamos ilustrar novamente para ficar mais simples assimilação:



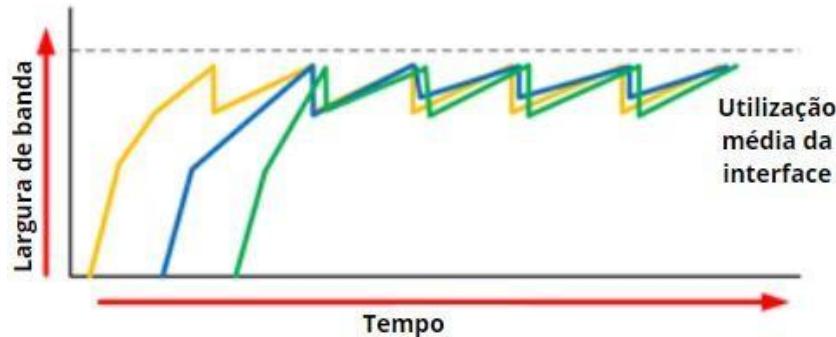
Vamos entender o gráfico: As linhas representam três conexões TCP diferentes. Essas 03 conexões TCP começaram em momentos diferentes, e depois de algum tempo, a interface ficou congestionada, isso causou o descarte de todos os pacotes de todas as conexões TCP. Nesse processo, o tamanho das janelas de todas as conexões TCP cairá para um e, quando não houver mais congestionamento na interface, todos os tamanhos de janela aumentarão novamente.

Quando a interface congestionar novamente, o tamanho da janela cairá para um, e a história se repetirá. O resultado disso é que não usamos toda a largura de banda disponível suportada pela interface. Observe na linha tracejada que a utilização média da interface não é alta, ou seja, estamos desperdiçando recursos.

Para evitar o **‘TCP global synchronization’**, podemos utilizar um recurso chamado **RED (Random Early Detection)**. Este é um recurso que descarta pacotes “aleatórios” de fluxos TCP com base no número de pacotes em uma fila e na marcação **TOS (Tipo de Serviço)** dos pacotes.

Quando os pacotes são descartados antes que a fila esteja cheia, podemos evitar o **‘global synchronization’**. Se você não entendeu bem a função do **RED**, não se preocupe, isso é assunto mais avançado, porém coloquei aqui porque é importante você saber que existe, mesmo não entendendo totalmente o funcionamento!

O resultado final será semelhante a este:



Análise o gráfico, observe que quando usamos RED, a utilização média de interface melhora e muito.

O UDP, ao contrário do TCP, é um protocolo sem conexão e continuará enviando tráfego independe do que aconteça. Como não há a janela flutuante, é possível que toda a ‘banda’ dos links seja consumida, por isso, em muitas vezes é interessante limitar o tráfego UDP, sob o risco de faltar ‘banda’ para o tráfego TCP quando houver congestionamento.

1.7 Configure and verify IPv4 addressing and subnetting

O protocolo IP usa ‘pacotes IP’ para transportar informações. Cada pacote IP é uma unidade única de informação, que além dos dados, carrega também informações para determinar o destino e caminho dos pacotes. Essa decisão para onde e por onde enviar os pacotes é feita observando o endereço IP de destino.

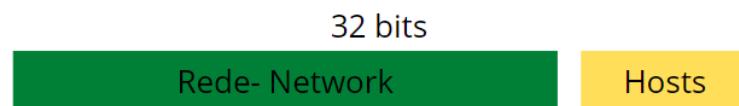
Vamos aprofundar e estudar as principais características do protocolo IP:

- Opera na camada de rede do modelo OSI (camada 03).
- Protocolo sem conexão: O protocolo IP não estabelece uma conexão para transportar dados, isso é trabalho da camada de “transporte”.
- Cada pacote é tratado de forma independente; não havendo uma ordem estabelecida para os pacotes chegarem ao destino.
- Hierárquico: Os endereços IPs têm uma hierarquia; estudaremos isso mais a frente, quando falarmos sobre sub-rede e máscaras de sub-rede.

Os endereços IPs servem para identificar o dispositivo na rede. Um endereço IP é como um número de telefone: Cada número de telefone é exclusivo, possibilitando assim, que entremos em contato com a pessoa escolhida. Os endereços IPs funcionam da mesma forma.

Agora que já entendemos um pouco do protocolo IP, vamos entender o que é um endereço IPv4:

O endereço IPv4 é formado por 32 bits, dividido em duas partes: A parte da rede e a parte do host:



Embora o endereço IP seja formado por 32 bits, convencionou separá-los em 4 blocos de 8 bits. 8 bits formam “1 byte”. Portanto, podemos representar o endereço anterior da seguinte maneira:



A parte da rede nos informa a qual “rede” o endereço IP pertence. Podemos compará-la com o código da cidade (DDD) de um número de telefone. A parte “host” identifica exclusivamente o dispositivo na rede; são como os últimos dígitos do número de telefone.

Provavelmente você já viu o endereço IP 192.168.1.1 configurado em algum dispositivo, pois este é um endereço IP muito usado em redes locais. Neste endereço IP, os primeiros 3 bytes são o endereço de “rede” e o último byte é o endereço do “host”:

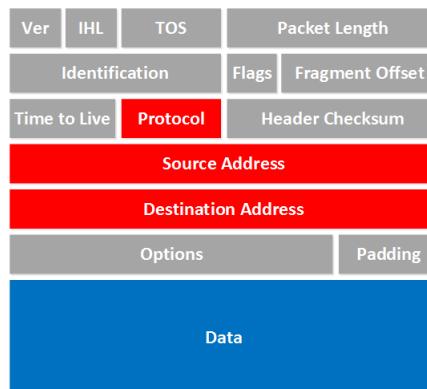


A pergunta que está na sua cabeça agora é: Por que os primeiros 3 bytes são a parte “rede” e por que o último byte é a parte “host”? Boa pergunta!

Até o momento, apenas informei o endereço IP, mas lembre-se, quando configuramos um endereço IP, precisamos especificar a **máscara de sub-rede (subnet)**. Então, ao configurar esse endereço IP 192.168.1.1, teríamos que configurar junto a máscara de sub-rede, nesse caso: 255.255.255.0.

A máscara de sub-rede informa ao dispositivo qual parte é a parte “rede” e qual parte é a parte “host”. Daqui a pouco falaremos mais sobre os temidos cálculos binários e de sub-rede, por enquanto, apenas saiba que a máscara de sub-rede nos diz qual parte do endereço IP é a parte “rede” e qual parte é para “hosts”.

Hora de verificar os campos que compõe um pacote IP:



Observe a complexidade do pacote, a quantidade de campos! Mas vamos por partes, nesse momento vamos ignorar todos os campos que estão pintados de cinza e focaremos nos outros campos.

- Protocolo:** Neste campo encontraremos o protocolo que esta sendo utilizado na camada acima (de transporte), ou seja, é assim que especificamos qual protocolo da **camada de transporte** esta sendo usado, provavelmente será o TCP ou UDP.
- Endereço de origem:** Neste campo encontraremos o endereço IP do dispositivo que criou o pacote.
- Endereço de destino:** Neste campo encontraremos o endereço IP do dispositivo que deve receber o pacote IP.
- Dados:** Neste campo estão os dados que estão sendo transmitidos para o outro host.

Fácil de entender, certo? Não há necessidade de se preocupar com os outros campos se você está estudando para o CCNA.

É importante observarmos um pacote IP real, para isso, vamos utilizar novamente o Wireshark, e ver como é essa separação de cada campo:

```
▼ Internet Protocol, Src: 192.168.69.2 (192.168.69.2), Dst: 192.168.69.1 (192.168.69.1)
  Version: 4
  Header length: 20 bytes
  ▶ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
  Total Length: 497
  Identification: 0xf5db (62939)
  ▶ Flags: 0x02 (Don't Fragment)
  Fragment offset: 0
  Time to live: 64
  Protocol: TCP (6)
  ▶ Header checksum: 0x37d7 [correct]
  Source: 192.168.69.2 (192.168.69.2)
  Destination: 192.168.69.1 (192.168.69.1)
```

Sensacional, não? Está tudo ai! Vamos voltar para a formação do endereço IPv4, analisando o endereço 192.168.1.1.

Sabemos que o endereço 192.168.1.1 é um valor de 32 bits. Em binário, ele é representado dessa forma:

1100000010101000000000100000001

Este número não é nada amigável para seres humanos, então, para tornar nossa vida mais fácil, podemos dividir esse número em “blocos” de 8 bits. Sempre bom lembrar, 8 bits, podem ser chamados de byte ou octeto:

11000000 10101000 00000001 00000001

Agora podemos converter cada byte em decimal, vamos pegar o primeiro bloco e convertê-lo de binário para decimal usando a seguinte tabela (essa tabela ajuda muito nessas conversões):

Bits	128	64	32	16	8	4	2	1
	0	0	0	0	0	0	0	0

Primeiro Byte:

11000000

Bits	128	64	32	16	8	4	2	1
	1	1	0	0	0	0	0	0

$$128 + 64 = 192$$

Segundo Byte:

10101000

Bits	128	64	32	16	8	4	2	1
	1	0	0	1	0	0	0	0

$$128 + 32 + 8 = 168$$

Terceiro Byte:

00000001

Bits	128	64	32	16	8	4	2	1
	0	0	0	0	0	0	0	1

Apenas o último bit, então é 1

00000001

Bits	128	64	32	16	8	4	2	1
	0	0	0	0	0	0	0	1

Igual ao terceiro byte, número decimal 1

Isso nos dá o endereço IP:

192	168	1	1
-----	-----	---	---

Agora já sabemos o processo de formação dos endereços IPs, e ainda fizemos alguns cálculos básicos transformando números binários em decimais.

Mas, ainda há mais detalhes: Existem diferentes classes de redes. Provavelmente você já tenha ouvido falar de redes de classe **A**, **B** ou **C**. Por exemplo, o endereço IP que acabamos de usar: 192.168.1.1 é um endereço de rede classe C.

Nós temos 3 classes diferentes de redes para trabalhar:

- Classe A
- Classe B
- Classe C

A maneira que nos faz classificar uma rede em determinada classe, é a quantidade de hosts podemos colocar em cada rede, observe o exemplo abaixo:

192	168	1	1
Rede	Rede	Rede	Hosts

Os primeiros 3 octetos que estão na cor mais escura são a parte de “rede” deste endereço. A parte clara é reservada para “hosts”. Portanto, podemos usar o último octeto (ou byte) para distribuir endereço de IPs para os dispositivos da rede, lembrando sempre que, estes endereços devem ser únicos na rede.

Desta forma, outros computadores que estão na mesma rede podem obter os seguintes endereços:

- 192.168.1.1
- 192.168.1.2
- 192.168.1.3

Observe, a parte de “rede” dos hosts é a mesma em todos os endereços IPs.

Um computador que esteja com endereço de IP 192.168.2.1 não está na mesma rede do exemplo acima, pois sua parte de “rede” é diferente: **192.168.2.X** em comparação com **192.168.1.X**.

Qual será o comportamento de um computador quando ele necessitar enviar um pacote IP para outra rede? Dê uma olhada no seu próprio computador para encontrar a resposta:

Se você estiver usando o Windows, clique no botão **Iniciar**, digite **CMD** e pressione **Enter**. Na tela preta que vai abrir, use o comando **ipconfig** para pesquisar as informações de IP:

```
C:\Documents and Settings\Computer>ipconfig  
Windows IP Configuration  
Ethernet adapter Local Area Connection:  
    Connection-specific DNS Suffix . :  
    IP Address..... : 192.168.1.1  
    Subnet Mask..... : 255.255.255.0  
    Default Gateway ..... : 192.168.1.254
```

O computador acima está na rede **192.168.1.X**. Quando esse computador precisar enviar algo para outra rede, ele usará seu **default gateway**. Geralmente, o default gateway (ou gateway padrão) é um roteador; no exemplo acima, o roteador possui o endereço IP **192.168.1.254**. Observe que a porção de rede é igual ao do computador: **192.168.1.X**

Vejamos agora, a diferenças entre as classes:

Classe A: Ideal para grandes empresas, onde é necessário um grande número de computadores em cada rede.



Classe B: Aqui é mais equilibrado, sendo possível construir mais redes, e ter um pouco menos de computadores em cada rede.



Classe C: Ideal para pequenas empresas, aqui podemos ter um grande número de redes, mas poucos hosts em cada rede.



Se você me acompanhou até agora, deve estar se perguntando: Como é possível definir que o IP 192.168.1.1 por exemplo faz parte da classe C?

Por convenção ficou definido que:

- **Classe A:** O primeiro bit sempre será 0;
- **Classe B:** Os dois primeiros bits sempre serão 10;
- **Classe C:** Os três primeiros bits sempre serão 110.



Portanto, transformando de binário para decimal, obteremos os seguintes intervalos:

- **Classe A** começa em 0.0.0.0
- **Classe B** começa em 128.0.0.0
- **Classe C** começa em 192.0.0.0

Dessa forma, os intervalos exatos para cada classe de IP são:

- **Classe A:** 0.0.0.0 - 126.255.255.255
- **Classe B:** 128.0.0.0 - 191.255.255.255
- **Classe C:** 192.0.0.0 - 223.255.255.255

Os olhos mais atentos devem ter notado 2 questões:

Se você analisar bem, verá que não há a sub-rede 127.0.0.0! Ela não está na faixa de classe A e nem na classe B, então onde está essa rede? Garanto a vocês que não é erro de digitação!

Os olhos mais atentos também notaram que a Classe C para em 223.255.255.255, mas porque não em 255.255.255.255?

Vamos analisar a primeira pergunta e descobrir onde se encaixa a rede 127.0.0.0: Vá até o prompt de comando do seu computador e digite “ping 127.0.0.1”, apesar de você não ter esse dispositivo na sua rede, você obterá uma resposta. Este intervalo de rede é usado como “loopback”. Interface de loopback serve para você verificar se sua pilha IP está OK, ou seja, serve para testar se o protocolo IP na sua máquina está funcionando.

Agora vamos a segunda pergunta: Existe mais uma classe de intervalo IP, essa classe é denominada classe D. A diferença dessa classe para as outras, é que não usamos os endereços IPs dessa faixa para atribuir endereços aos hosts, os endereços dessa classe são usados para “multicast”. O intervalo da classe D começa em 224.0.0.0.

Subneting

Chegou a hora de fazermos contas e transformações de binário para decimal e vice versa.

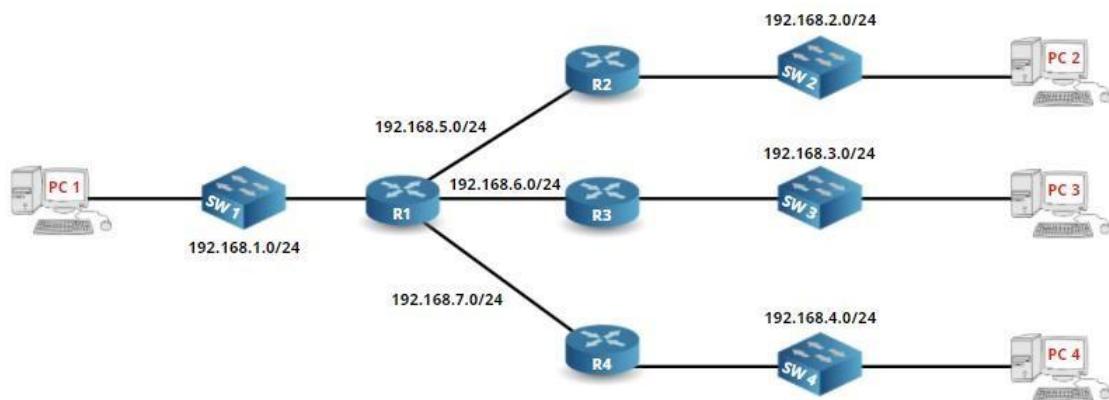
Até aqui aprendemos que há três classes de sub-redes:

- **Classe A:** 16777216 endereços no total.
- **Classe B:** 65536 endereços no total.
- **Classe C:** 256 endereços no total.

Uma sub-rede é uma porção de uma rede que se enquadra na faixa de classe A, B ou C.

Por exemplo, 172.16.0.0/16 é uma rede de classe B. Esta rede é muito grande, ela começa com 172.16.0.0 e termina com 172.16.255.255. Em vez de uma grande rede, podemos usar uma “porção” menor. Um exemplo é 172.16.1.0/24. Essa sub-rede se enquadra na rede 172.16.0.0/16 classe B, por isso é chamada de “sub-rede”.

Observe a topologia abaixo para entender melhor porque precisamos de sub-redes:



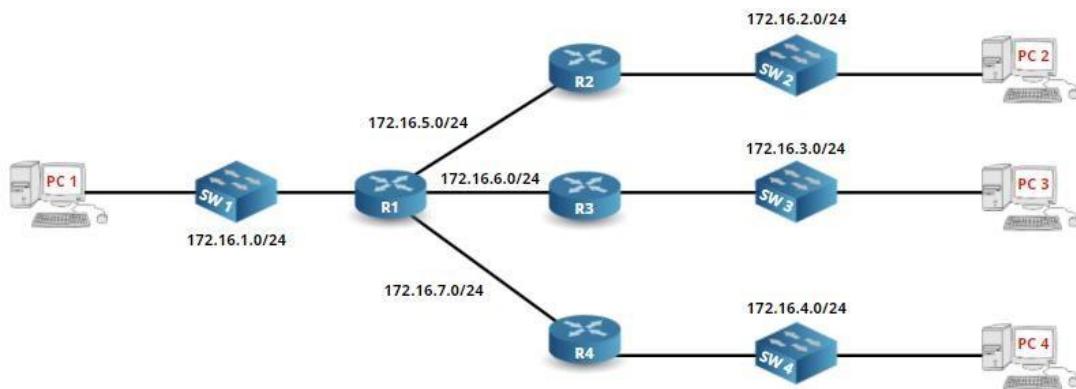
Acima, temos a rede de uma pequena empresa com quatro roteadores. Cada roteador representa uma filial dessa empresa. Atrás de cada roteador, há um switch com um host. Nessa topologia, estamos usando as seguintes redes de classe C:

- R1: 192.168.1.0/24
- R2: 192.168.2.0/24
- R3: 192.168.3.0/24
- R4: 192.168.4.0/24
- R1-R2: 192.168.5.0/24
- R1-R3: 192.168.6.0/24
- R1-R4: 192.168.7.0/24

Tecnicamente, essa escolha funcionará; mas usar essas redes é uma escolha ruim por duas razões:

- Há uma quantidade limitada de redes de classe C (privadas). Podemos escolher entre 192.168.0.0/24 e 192.168.255.0/24. E se tivermos mais de 256 filiais? Não haverá espaço suficiente, então, teremos que escolher outro intervalo de rede.
- Os links entre os roteadores são links ponto a ponto, portanto, precisamos apenas de dois endereços IP, um para cada roteador. Quando usamos uma máscara de sub-rede /24, estamos perdendo 252 endereços IP. Isso não é problema quando usamos endereços IPs privados, mas é um problema quando usamos endereços IPs públicos. Os endereços IPs públicos são limitados, então temos que usar sub-redes menores para que não haja desperdício.

Vamos melhorar este diagrama de rede, usando sub-redes da classe A ou B:



Agora, estamos usando sub-redes da rede classe B 172.16.0.0/16. Esta rede oferece muito mais espaço, o intervalo privado começa com 172.16.0.0 e termina com 172.31.255.255. As opções para escolha da sub-rede aumentaram muito.

Ainda estamos desperdiçando endereços IP entre os links dos roteadores, portanto, há espaço para melhorias.

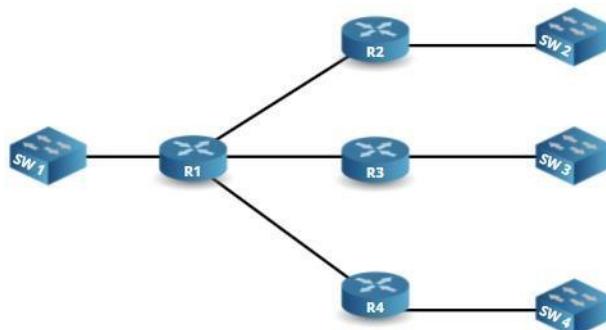
Quando falamos em sub-redes, há duas coisas que precisamos pensar em primeiro lugar:

- Criação do design:** Como nos exemplos acima, você deve pensar em quantas sub-redes precisa e quais serão usadas. Por exemplo, se adicionarmos mais um roteador, qual sub-rede usaremos? E se criarmos várias VLANs no SW1, quais sub-redes usaremos para essas VLANs? É necessário pensar e planejar esses passos com antecedência para evitar maiores complicações.
- Cálculos:** No primeiro exemplo, usamos máscara sub-rede /24, que é mais fácil, mas não a mais eficiente. Entre os roteadores, o ideal é utilizar uma sub-rede menor, formada por endereços IPs suficientes para apenas os dois roteadores. Esses cálculos podem ser feitos com calculadoras de sub-rede (é possível encontrar dezenas dessas pesquisando no Google), mas nos exames (Cisco), você deverá calculá-las sozinho.

Vamos para mais um exemplo, assim ficará mais claro as escolhas que temos que fazer para projetar uma rede.

Design de sub-redes:

Vamos utilizar a mesma topologia:



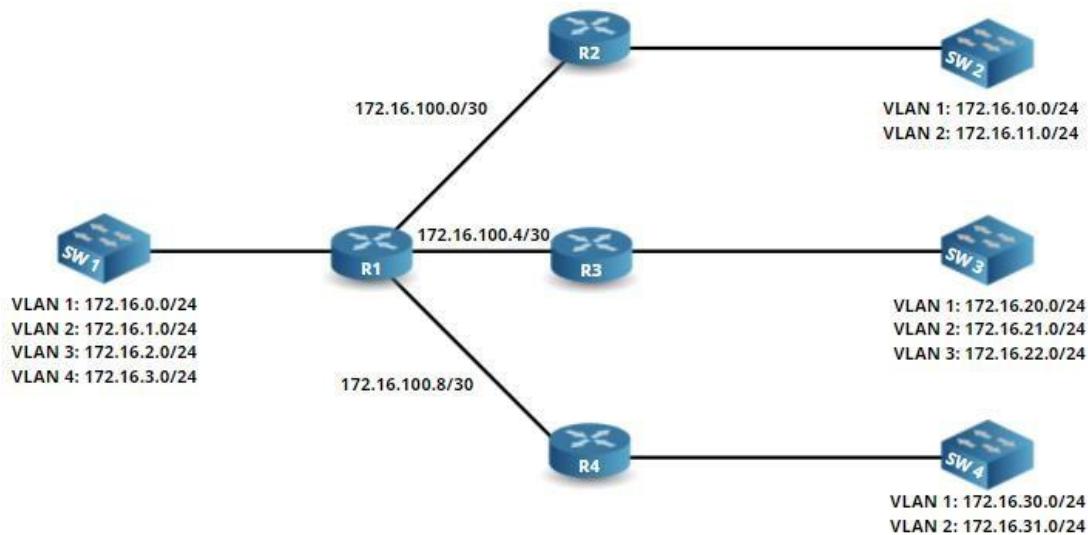
Vamos imaginar que estamos participando da montagem da rede de uma empresa. O roteador 1 é o roteador da matriz e os demais roteadores estão em filiais. Temos que montar o endereçamento IP obedecendo as seguintes exigências:

- R1: Deve ter 04 vlans com capacidade para 100 usuários;
- R2: Deve ter 02 vlans com capacidade para 30 usuários;
- R3: Deve ter 03 vlans com capacidade para 20 usuários;
- R4: Deve ter 02 vlans com capacidade para 10 usuários;

A empresa tem a expectativa que o número de usuários e vlans vá dobrar no próximo ano e haverá a criação de mais duas filiais. Precisamos levar em conta todos esses fatos na hora do planejamento.

1^a Opção: Sub-redes com tamanho único:

A maior vlan possui 100 usuários, isso significa que essa vlan tem que ter capacidade para 200 endereços IP. Nós podemos usar uma máscara /24 que possui capacidade para 254 endereços IP. A rede, poderia então, ser dividida da seguinte forma:



Na topologia acima, adicionamos VLANs para cada filial, e usamos /24 em todos os lugares.

Você deve estar se perguntando: Porque?

Está é uma rede corporativa que usa endereços IPs privados, e embora algumas VLANs exigissem apenas uma sub-rede para 10 usuários (20 ao levar em consideração o crescimento), é mais conveniente usar o mesmo tamanho de sub-rede em todos os lugares. As pessoas estão familiarizadas com as máscaras de sub-rede /24. Nessas redes, o primeiro endereço IP utilizável começa com .1, e o último termina com .254, é comum que um desses endereços seja usado como gateway padrão para cada VLAN.

Há muito espaço para os endereços IPs, portanto, não é necessário usar pequenas sub-redes. Também há bastante espaço entre as redes de cada filial. Na filial do roteador R1, podemos usar de 172.16.0.0 - 172.16.9.255. Na filial do R2, apesar de necessitarem apenas de duas VLANs (quatro ao levar em conta o crescimento), deixamos reservado muito espaço também.

A única exceção são os links dos roteadores. Começamos com 172.16.100.0 para que esse seja o range para o link entre eles, de 172.16.0.0 - 172.16.99.255. Como precisamos apenas de dois endereços IPs entre os roteadores, decidi escolher a menor sub-rede que podemos usar, um /30.

2^a opção: Sub-redes de tamanho variado

E se a rede que estamos projetando não fosse uma LAN, mas uma rede de provedor de serviços que usa endereços IPs públicos? Cada roteador representando um cliente diferente. Nesse caso, precisamos ser o mais eficiente possível com os endereços IP.

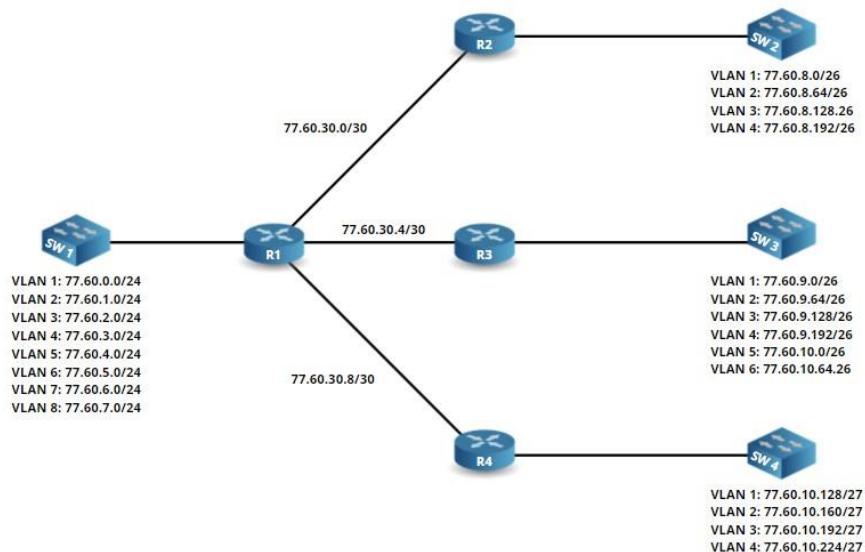
Vejamos os novos requisitos:

- R1: quatro VLANs, cada uma com 100 servidores.
- R2: duas VLANs, cada uma com 30 servidores.
- R3: três VLANs, cada uma com 20 servidores.
- R4: duas VLANs, cada uma com 10 servidores.

Digamos que o número de clientes, VLANs e servidores possa dobrar no próximo ano. Precisaremos projetar um plano de sub-rede que leve em conta:

- R1: oito VLANs, cada uma com 200 servidores.
- R2: quatro VLANs, cada uma com 60 servidores.
- R3: seis VLANs, cada uma com 40 servidores.
- R4: quatro VLANs, cada uma com 20 servidores.

Além disso, precisamos deixar espaço reservado para potenciais novos cliente: R5, R6, R7 e R8.



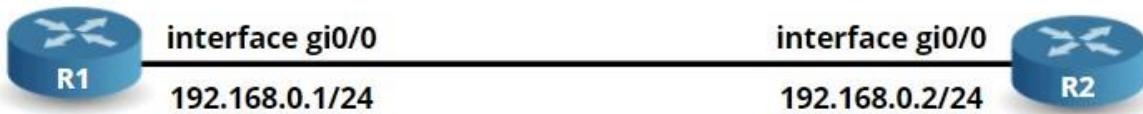
Vamos entender por que escolhi as sub-redes acima:

- As VLANs por trás do R1 terão 200 servidores, isso significa que precisamos de um /24 pelo menos, já que permite 254 endereços IP utilizáveis.
- As VLANs por trás de R2 terão 60 servidores, a menor sub-rede que podemos usar é um /26, que permite 62 endereços IP utilizáveis.
- As VLANs por trás de R3 terão 40 servidores, a menor sub-rede que podemos usar é um /26, que permite 62 endereços IP utilizáveis.
- As VLANs por trás do R4 terão 20 servidores, a menor sub-rede que podemos usar é um /27, que permite 30 endereços IP utilizáveis.
- Entre os roteadores, podemos usar /30, pois permite 2 endereços IP utilizáveis.

Começamos com com 77.60.30 /30, pois usamos o range de 77.60.0.0 - 77.60.10.255 para os quatro roteadores. Se o número de clientes dobrar, provavelmente usaremos algo como 77.60.11.0 - 77.60.20.255. A escolha até o 77.60.30.0 foi para reservar um pouco mais de espaço.

Configuração de endereço IPv4 em roteadores Cisco

Para demonstrar essa configuração, usaremos a topologia abaixo:



Vamos configurar o R1 primeiro:

```

R1(config-if)#interface g0/0
R1 (config-if)#ip address 192.168.0.1 255.255.255.0
R1 (config-if)#description Link para R2
R1 (config-if)#no shutdown
%LINK-3-UPDOWN: Interface GigabitEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state
to up

```

Primeiro entramos na interface, depois definimos o endereço IPv4 juntamente com a máscara de rede. Na linha debaixo colocamos a descrição, isso ajuda muito na hora do troubleshooting, e por último ativamos a interface. Como você pode ver, é uma configuração bem simples.

Vamos configurar o R2, observe que os comandos são os mesmos, com exceção do endereço IPv4 e a descrição da interface:

```

R2(config-if)#interface g0/0
R2 (config-if)#ip address 192.168.0.2 255.255.255.0
R2 (config-if)#description Link para R1
R2 (config-if)#no shutdown
%LINK-3-UPDOWN: Interface GigabitEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state
to up

```

Para finalizar, vamos realizar um ping do R1 para o R2, testando assim a conectividade:

```

R1#ping 192.168.0.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.0.2, timeout is 2 seconds:
!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 36/39/40 ms
R1#

```

Configuração realizada com sucesso.

Noções básicas de números binários

Antes de começarmos a calcular sub-redes e nos aprofundarmos no endereçamento IP, vamos verificar alguns fundamentos de cálculos binários.

Todos nós estamos acostumados a trabalhar com números decimais onde contamos de 1 a 10. Contar até 10 é um processo natural porque temos 10 dedos, então não precisamos realizar contas ‘de cabeça’.

No sistema binário, trabalhamos apenas com 0 ou 1.

- 0 = desligado
- 1 = ligado

Vejamos alguns exemplos de como podemos usar o binário para criar alguns números:

128	64	32	16	8	4	2	1
-----	----	----	----	---	---	---	---

Acima temos 8 bits. O bit mais à esquerda é chamado de **bit mais significativo (most significant bit - MSB)**, pois é o bit com o valor mais elevado (128). O bit do lado direito é chamado de **bit menos significativo (least significant bit - LSB)** porque possui o valor mais baixo (1).

Vamos ver alguns exemplos de como convertemos números decimais em binários.

Decimal para binário:

Se quisermos o número decimal “0” em binário, isso significa que deixamos todos os bits “desligados”:

128	64	32	16	8	4	2	1
0	0	0	0	0	0	0	0

Número zero em binário

Vamos pegar o número decimal 178 e transformá-lo em binário. Fazemos isso começando da esquerda e, em seguida, tentamos “encaixar” os bits até compormos o número:

128	64	32	16	8	4	2	1
1	0	1	1	0	0	1	0

$128 + 32 + 16 + 2 = 178$

Vamos para outro exemplo, transformando o número decimal 31 em binário. Comece da esquerda e veja quais bits “se encaixam”:

128	64	32	16	8	4	2	1
0	0	0	1	1	1	1	1

$16 + 8 + 4 + 2 + 1 = 31$

Vamos transformar o número decimal 255 em binário:

128	64	32	16	8	4	2	1
1	1	1	1	1	1	1	1

$128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255$

Quando trabalhamos com 8 bits, o número mais alto que conseguimos alcançar é o 255.

Binário para Decimal

Podemos fazer o contrário, transformar binário em decimal. Vamos utilizar o seguinte número em binário: 10111110

128	64	32	16	8	4	2	1
1	0	1	1	1	1	1	0

Sempre que aparecer o número um, some o valor da casa decimal correspondente. Neste exemplo, será $128 + 32 + 16 + 8 + 4 + 2 = 190$.

Mais um exemplo, o número binário: 00010110

128	64	32	16	8	4	2	1
0	0	0	1	0	1	1	0

Desta vez, temos $16 + 4 + 2 = 22$.

Cada bit adicional, dobra o valor decimal. Por exemplo: 2, 4, 8, 16, 32, 64, 128, etc. Isso é chamado “potência de dois”.

Endereço de Network e Broadcast

Antes de prosseguir, preciso falar da existência de 2 endereços IPs que não podemos usar em nossas redes. São eles:

- Endereço de rede (**Network Address**)
- Endereço de transmissão (**Broadcast Address**)

O **endereço de rede** não pode ser usado como endereço IP em um computador, pois esse endereço, é usado para “definir” a rede.

Já o endereço de **broadcast** não pode ser usado em um computador como endereço IP, pois ele é usado por aplicativos para enviar mensagens de broadcast. **Broadcast** é um pacote IP que será recebido por **todos os dispositivos** da rede.

Hora de aprender como reconhecemos esses dois endereços IPs em uma determinada rede:

Como exemplo, vamos utilizar o mesmo endereço IP que estamos utilizando: **192.168.1.1**, como já aprendemos anteriormente, este é um endereço IP da **classe C**.



Os 3 **primeiros octetos são a parte da rede**, e como dito anteriormente **são imutáveis**, isso significa que precisamos focar no último octeto.

O **endereço de rede** sempre será o **endereço mais baixo**, no caso dessa rede o endereço mais baixo seria o IP 192.168.1.0.

O **endereço broadcast** sempre será o **endereço IP mais alto**, logo o broadcast dessa rede é o IP 192.168.1.255.

Podemos resumir também da seguinte maneira:

- Definir todos os bits da parte do host como 0 fornece o endereço de rede:



- Definir todos os bits da parte de host como 1 fornece o endereço de broadcast:



Esses 2 endereços IP não podem ser usados para hosts.

Com essa explicação em mente, vamos para o próximo tópico:

Sub-redes em binário

Vamos aprender na prática a calcular sub-redes com números binários.

Sub-rede Classe C

Começamos com uma rede de classe C, a mais fácil de entender. Utilizaremos o endereço 192.168.1.0 com a máscara de sub-rede padrão 255.255.255.0.

Em binário, é assim:

192	168	1	0
11000000	10101000	00000001	00000000

Sabemos que uma rede de classe C tem 3 bytes para a parte da rede e um byte para hosts:



O dispositivo de rede sabe qual parte é da rede e qual a parte do host por causa da máscara de sub-rede. A máscara de sub-rede padrão para a rede 192.168.1.0 é 255.255.255.0. Vamos trabalhar com o esquema abaixo:

Endereço IP Decimal	192	168	1	0
Endereço IP binário	11000000	10101000	00000001	00000000
Máscara de sub-rede (decimal)	255	255	255	0
Máscara de sub-rede (binário)	11111111	11111111	11111111	00000000

Os números ‘1’ na máscara de sub-rede, indicam a parte do endereço de rede, os ‘0’ indicam a parte do host. Vamos remover os números decimais para que você possa ver o endereço de rede e a máscara de sub-rede um imediatamente abaixo do outro:

Endereço IP binário	11000000	10101000	00000001	00000000
Máscara de sub-rede (binário)	11111111	11111111	11111111	00000000

Em outras palavras, a máscara de sub-rede informa que os primeiros 24 bits (192.168.1) são a parte da rede e os 8 bits restantes (.0) são para hosts.

Como dito anteriormente, o valor mais alto que podemos formar com 8 bits se definirmos todos como 1 é 255:

128	64	32	16	8	4	2	1
1	1	1	1	1	1	1	1
$128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255$							

O valor mais alto que podemos criar é 255, mas isso não significa que podemos ter 255 hosts nesta rede, lembre-se, existem 2 endereços que não podemos usar:

Endereço de rede: É o endereço onde todos os bits do host são definidos como 0.

192	168	1	0
11000000	10101000	00000001	00000000

Endereço de broadcast: É o endereço onde todos os bits do host são definidos como 1.

192	168	1	0
11000000	10101000	00000001	11111111

Então podemos ter nessa rede $255 - 2 = 253$, o que significa que podemos ter no máximo 253 hosts em nossa rede?

A resposta é não! O valor mais alto que podemos criar com 8 bits não é 255, mas 256. Por quê? Porque também é possível usar o valor de “0”.

Isso significa que podemos usar 192.168.1.1 - 192.168.1.254 como endereços IP para nossos hosts.

Bom, agora ficou claro como é uma rede em binário, o que a máscara de sub-rede faz, quais são os endereços de rede e broadcast, e que podemos acomodar 254 hosts em uma rede Classe C.

Digamos que eu não queira ter uma única rede onde possa caber em 254 hosts, mas quero ter 2 redes? Isso é possível? Com certeza! Basicamente, o que estamos fazendo é pegar uma rede Classe C e dividi-la em 2 partes, e isso é o que chamamos de sub-rede. Vamos dar uma olhada em binário:

Endereço IP Decimal	192	168	1	0
Endereço IP binário	11000000	10101000	00000001	00000000
Máscara de sub-rede (decimal)	255	255	255	0
Máscara de sub-rede (binário)	11111111	11111111	11111111	00000000

A máscara de sub-rede define o tamanho da rede, portanto, se quisermos criar mais sub-redes, teremos que “emprestar” bits da parte do host.

Para cada bit emprestado, dobramos o número de sub-redes. Pegando emprestado 1 bit, criamos 2 sub-redes dessa única rede. Existem 8 bits de host, portanto, se roubarmos um para criar mais sub-redes, significa que temos apenas 7 bits restantes para os hosts. Vamos para a parte prática. Esta, será a nova máscara de sub-rede:

255	255	255	128
11111111	11111111	11111111	10000000

Os primeiros 24 bits são iguais, porém, pegamos emprestado o primeiro bit do quarto octeto. Este bit tem o valor decimal de 128, portanto, nossa máscara de sub-rede se torna 255.255.255.128.

Vamos ver como são essas novas sub-redes, lembre-se, ainda temos 7 bits que sobraram e podem ser alocados para hosts:

128	64	32	16	8	4	2	1
N\ D	0	0	0	0	0	0	0

Não podemos usar o primeiro bit, pois agora ele é usado para o endereço de rede (por isso está N\|D – Não disponível), graças à nossa máscara de sub-rede. Qual é o maior número decimal que podemos criar com 7 bits?

$64 + 32 + 16 + 8 + 4 + 2 + 1 = 127$. Não se esqueça de que começamos a contar do 0, portanto, no total, temos 128 endereços.

Nossa rede classe C original foi dividida em duas sub-redes, cada uma com 128 endereços. Observe abaixo como ficaram as duas sub-redes!

Sub-rede 1:

Endereço IP	192	168	1	0
	11000000	10101000	00000001	00000000
Máscara de sub-rede	255	255	255	128
	11111111	11111111	11111111	10000000

Endereço de rede:

O endereço de rede tem todos os bits de host definidos como 0, por isso é 192.168.1.0:

192	168	1	0
11000000	10101000	00000001	00000000

Primeiro endereço utilizável:

O primeiro endereço IP utilizável é aquele que vem depois do endereço de rede, no caso em tela, será 192.168.1.1:

192	168	1	1
11000000	10101000	00000001	00000001

Último endereço utilizável:

O último endereço IP que podemos usar para host é aquele antes do endereço de broadcast, portanto, será 192.168.1.126:

192	168	1	126
11000000	10101000	00000001	01111110

Endereço de broadcast:

O endereço de broadcast tem todos os bits de host definidos como 1, portanto, o endereço será 192.168.1.127:

192	168	1	127
11000000	10101000	00000001	01111111

Sub-rede 2:

A próxima sub-rede dessa rede que acabamos de dividir começará em .128, pois a sub-rede anterior acabou em .127:

Endereço IP	192	168	1	128
	11000000	10101000	00000001	10000000
Máscara de sub-rede	255	255	255	128
	11111111	11111111	11111111	10000000

Endereço de rede:

O endereço de rede tem todos os bits de host definidos como 0, por isso é 192.168.1.128:

192	168	1	128
11000000	10101000	00000001	10000000

Primeiro endereço utilizável:

O primeiro endereço IP de host utilizável é aquele que vem depois do endereço de rede, será 192.168.1.129:

192	168	1	129
11000000	10101000	00000001	10000001

Último endereço utilizável:

O último endereço IP que podemos usar para um host é aquele antes do endereço de broadcast, portanto, será 192.168.1.254:

192	168	1	254
11000000	10101000	00000001	11111110

Endereço de broadcast:

O endereço de broadcast tem todos os bits de host definidos como 1, portanto, o endereço será 192.168.1.255:

192	168	1	255
11000000	10101000	00000001	11111111

É trabalhoso, mas não é complicado. Acabamos de dividir em 2 sub-redes e descobrimos quais são os endereços de rede e broadcast, e qual range de endereços IPs podemos usar nos hosts.

Sub-rede Classe B

Vamos agora para a rede classe B, observe que o processo é o mesmo. Vamos trabalhar o endereço 172.16.0.0 e a máscara de sub-rede 255.255.0.0 e a partir dela vamos criar duas sub-redes:

Endereço IP	172	16	0	0
	10101100	00010000	00000000	10000000
Máscara de sub-rede	255	255	0	0
	11111111	11111111	00000000	00000000

Se quisermos criar mais sub-redes, precisamos pegar bits emprestados da parte do host. Para cada bit emprestado, você pode dobrar o número de sub-redes. Pegando emprestado 1 bit, criamos 2 sub-redes dessa única rede. A diferença para uma rede de classe C é que temos **mais bits de host para usar, só isso**.

Qual será a nova máscara de sub-rede? Vamos dar uma olhada em binário:

255	255	255	128
11111111	11111111	10000000	10000000

Como você pode ver, a máscara de sub-rede da rede será 255.255.128.0 e temos $7 + 8 = 15$ bits de host restantes para usar. Quão “grande” são essas 2 sub-redes? Bem, nós temos 15 bits, então vamos calcular com apoio da tabela abaixo:

N\D	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
N\D	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

$$16384 + 8192 + 4096 + 2048 + 1024 + 512 + 256 + 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 32767$$

Não se esqueça do 0! Portanto, o valor mais alto que podemos criar com 15 bits é **32768**.

Se você quiser saber quantos endereços IP de host utilizáveis você possui, basta subtrair: 32768-2 (por causa da rede e do endereço de broadcast).

$32768 - 2 = 32766$ endereços IPs utilizáveis.

Uma maneira muito mais rápida de realizar esse cálculo é através da "potência de 2" que expliquei anteriormente:

2 à potência de 15 (ou $2 \times 2 \times 2$) = 32768.

32.768 menos 2 (rede + endereço de transmissão) = 32766.

Subrede 1:

Vamos calcular a primeira sub-rede:

Endereço IP	172	16	0	0
	10101100	00010000	00000000	00000000
Máscara de sub-rede	255	255	128	0
	11111111	11111111	10000000	00000000

Endereço de rede:

O endereço de rede tem todos os bits de host definidos como 0, por isso é 172.16.0.0:

172	16	0	0
10101100	00010000	00000000	00000000

Primeiro endereço utilizável:

O primeiro endereço IP utilizável é aquele que vem depois do endereço de rede, será 172.16.0.1:

172	16	0	1
10101100	00010000	00000000	00000001

Último endereço utilizável:

O último endereço IP que podemos usar para um host é aquele antes do endereço de broadcast, portanto, será 172.16.127.254:

172	16	127	254
10101100	00010000	01111111	11111110

Endereço de broadcast:

O endereço de broadcast tem todos os bits de host definidos como 1, portanto, o endereço será 172.16.127.255:

172	16	127	255
10101100	00010000	01111111	11111111

10101100	00010000	01111111	11111111
----------	----------	----------	----------

Subrede 2:

A próxima sub-rede dessa rede que acabamos de dividir começará em 172.16.128.0, pois a sub-rede anterior terminou em 172.16.127.255:

Endereço IP	172	16	128	0
	10101100	00010000	10000000	00000000
Máscara de sub-rede	255	255	128	0
	11111111	11111111	10000000	00000000

Endereço de rede:

O endereço de rede tem todos os bits de host definidos como 0, por isso é 172.16.128.0:

172	16	128	0
10101100	00010000	10000000	00000000

Primeiro endereço utilizável:

O primeiro endereço IP de host utilizável é aquele que vem depois do endereço de rede, será 172.16.128.1:

172	16	0	1
10101100	00010000	10000000	00000001

Último endereço utilizável:

O último endereço IP que podemos usar para um host é aquele antes do endereço de broadcast, portanto, será 172.16.255.254:

172	16	255	254
10101100	00010000	11111111	11111110

Endereço de broadcast:

O endereço de broadcast tem todos os bits de host definidos como 1, portanto, o endereço será 172.16.255.255:

172	16	255	255
10101100	00010000	11111111	11111111

Acabamos de dividir esta rede 172.16.0.0 da classe B em 2 sub-redes. Observe que estamos fazendo exatamente a mesma coisa que fizemos anteriormente, mas agora você temos mais bits para manipular!

Sub-rede Classe A

Vimos que criar sub-redes em uma rede de classe B também não é difícil, apesar de ser trabalhoso. Vamos partir agora para uma rede Classe A, e ver o que acontece:

Vamos pegar a rede 10.0.0.0 com máscara de sub-rede 255.0.0.0 e criar pelo menos 12 sub-redes a partir dela:

Endereço IP	10	0	0	0
	00001010	00000000	00000000	00000000
Máscara de sub-rede	255	0	0	0
	11111111	00000000	00000000	00000000

Se quisermos criar mais sub-redes, precisamos pegar bits emprestados da parte do host. Para cada bit emprestado, você pode dobrar o número de sub-redes (lembra da “potência de 2”?). Pegando emprestado 4 bits, podemos criar 16 sub-redes dessa única rede. 3 bits não seriam suficientes porque só podemos criar 8 sub-redes.

Qual será a nova máscara de sub-rede? Vamos dar uma olhada em binário:

255	240	0	0
-----	-----	---	---

11111111	11110000	00000000	00000000
----------	----------	----------	----------

Como você pode ver, a máscara de sub-rede será 255.240.0.0 e temos $4 + 8 + 8 = 20$ bits de host restantes para usar.

Quão “grandes” são essas 16 sub-redes? Bem, nós temos 20 bits, então vamos usar “potência de 2” para resolver esta questão:

$$2 \text{ à potência de } 20 = 1.048.576$$

Se você quiser saber quantos endereços IP de host utilizáveis você possui, é só calcular $1.048.576 - 2$ (por causa da rede e do endereço de broadcast).

$$1.048.576 - 2 = 1.048.574 \text{ endereços IP de host utilizáveis.}$$

Vamos calcular as sub-redes, não faremos para todas pois o processo é igual. Faremos apenas para 3 sub-redes.

Sub-rede 1:

Vamos aplicar a nova máscara de sub-rede, temos 20 bits reservados para hosts.

Endereço IP	10	0	0	0
	00001010	00000000	00000000	00000000
Máscara de sub-rede	255	240	0	0
	11111111	11110000	00000000	00000000

Endereço de rede:

O endereço de rede tem todos os bits de host definidos como 0, por isso é 10.0.0.0:

10	0	0	0
00001010	00000000	00000000	00000000

Primeiro endereço utilizável:

O primeiro endereço IP de host utilizável é aquele que vem depois do endereço de rede, será 10.0.0.1:

10	0	0	1
00001010	00000000	00000000	00000001

Último endereço utilizável:

O último endereço IP que podemos usar para um host é aquele antes do endereço de broadcast, portanto, será 10.0.0.15:

10	15	255	254
00001010	00001111	11111111	11111110

Endereço de broadcast:

O endereço de broadcast tem todos os bits de host definidos como 1, portanto, o endereço será 10.0.0.15:

10	15	255	255
10101100	00001111	11111111	11111111

Sub-rede 2:

O endereço de broadcast da sub-rede 1 era 10.0.0.15, então nossa próxima sub-rede começará em 10.0.0.16:

Endereço IP	10	16	0	0
	00001010	00010000	00000000	00000000
Máscara de sub-rede	255	240	0	0
	11111111	11110000	00000000	00000000

Endereço de rede:

O endereço de rede tem todos os bits de host definidos como 0, por isso é 10.16.0.0:

10	16	0	0
00001010	00010000	00000000	00000000

Primeiro endereço utilizável:

O primeiro endereço IP de host utilizável é aquele que vem depois do endereço de rede, será 10.16.0.1:

10	16	0	1
00001010	00011111	00000000	00000001

Último endereço utilizável:

O último endereço IP que podemos usar para um host é aquele antes do endereço de broadcast, portanto, será 10.31.255.254:

10	31	255	254
00001010	00011111	11111111	11111110

Endereço de broadcast:

O endereço de broadcast tem todos os bits de host definidos como 1, portanto, o endereço será 10.31.255.255:

10	31	255	255
10101100	00011111	11111111	11111111

Sub-rede 3:

O endereço de broadcast da sub-rede 1 era 10.31.255.255, então nossa próxima sub-rede começará em 10.32.0.0

Endereço IP	10	32	0	0
	00001010	00100000	00000000	00000000
Máscara de sub-rede	255	240	0	0
	11111111	11110000	00000000	00000000

Endereço de rede:

O endereço de rede tem todos os bits de host definidos como 0, por isso é 10.32.0.0:

10	32	0	0
00001010	00100000	00000000	00000000

Primeiro endereço utilizável:

O primeiro endereço IP de host utilizável é aquele que vem depois do endereço de rede, será 10.32.0.1:

10	32	0	1
00001010	00100000	00000000	00000001

Último endereço utilizável:

O último endereço IP que podemos usar para um host é aquele antes do endereço de broadcast, portanto, será 10.47.255.254:

10	47	255	254
00001010	00101111	11111111	11111110

Endereço de broadcast:

O endereço de broadcast tem todos os bits de host definidos como 1, portanto, o endereço será 10.47.255.255:

10	47	255	255
10101100	00101111	11111111	11111111

A mesma lógica pode ser utilizada para as demais sub-redes dessa rede.

Classless InterDomain Routing (CIDR)

Quando o esquema de endereçamento IP foi inventado, as pessoas que o desenvolveram achavam que apenas 3 classes diferentes de IP seriam suficientes: redes de classe A, B e C. Com isso, havia apenas três máscaras de sub-rede:

- Classe A: 255.0.0.0 (endereços 16.777.216)
- Classe B: 255.255.0.0 (65.536 endereços)
- Classe C: 255.255.255.0 (256 endereços)

Essas redes também são conhecidas como **redes classful**.

Quando a internet começou a crescer rapidamente no início dos anos 90, isso causou alguns problemas. Grandes empresas como a IBM, Microsoft receberam redes inteiras de classe A com milhões de endereços.

As empresas menores poderiam obter uma rede classe B com 65.536 endereços ou redes de classe C com 256 endereços. Com isso, muitos endereços foram perdidos, então, algo teve que ser feito.

A solução para este problema foi a criação do Classless InterDomain Routing (**CIDR**), ou seja, não trabalhamos mais só com as redes **classful**, mas passamos a trabalhar também com as redes classless.

Classless networks significa que não usamos apenas as redes classe A, B ou C, mas **podemos usar qualquer máscara de sub-rede que quisermos**. Além disso, em vez de escrever a máscara de sub-rede como 255.255.255.0, agora usamos uma notação de “bit” como /24. Isso representa o número de bits usados para a máscara de sub-rede. Por exemplo:

- 192.168.1.0 com máscara de sub-rede 255.255.255.0 é a mesma coisa que 192.168.1.0 / 24.
- 172.16.0.0 com máscara de sub-rede 255.255.0.0 é a mesma coisa que 172.16.0.0 / 16.
- 10.0.0.0 com máscara de sub-rede 255.0.0.0 é a mesma coisa que 10.0.0.0 / 8.

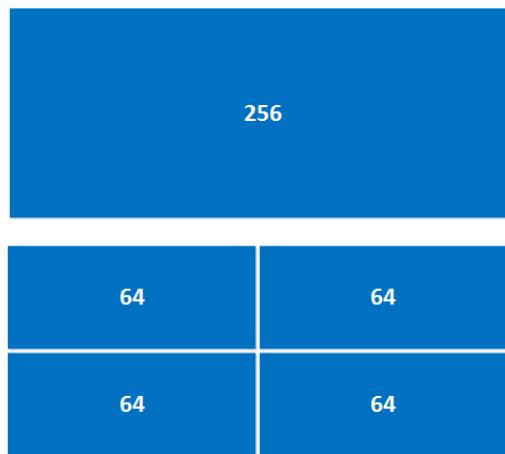
Esta é uma pequena visão geral com máscaras de sub-rede e a notação CIDR:

255.0.0.0	/8
255.128.0.0	/9
255.192.0.0	/10
255.224.0.0	/11
255.240.0.0	/12
255.248.0.0	/13
255.252.0.0	/14
255.254.0.0	/15
255.255.0.0	/16
255.255.128.0	/17
255.255.192.0	/18
255.255.224.0	/19
225.225.240.0	/20
255.255.248.0	/21
255.255.252.0	/22
255.255.254.0	/23
255.255.255.0	/24
255.255.255.128	/25
255.255.255.192	/26
255.255.255.224	/27
255.255.255.240	/28
255.255.255.248	/29

A notação CIDR é mais fácil de escrever do que digitar a máscara de sub-rede inteira. Infelizmente, a maioria dos sistemas operacionais e dispositivos de rede ainda exigem que digitemos a máscara de sub-rede completa.

Variable Length Subnet Mask (VLSM)

Nos exemplos de criação de sub-redes que trabalhamos até aqui, todas as nossas sub-redes tinham um “tamanho fixo”, ou seja, cada sub-rede tinha o mesmo tamanho. Por exemplo, pegamos uma rede de classe C, 192.168.1.0, e a dividimos em 4 blocos:



Esta não é a maneira mais eficiente de criar sub-redes! Por exemplo, imaginemos que temos uma rede em que temos que cumprir os seguintes requisitos:

- Uma sub-rede para 12 hosts.
- Uma sub-rede para 44 hosts.
- Uma sub-rede para 2 hosts (links ponto a ponto são um bom exemplo em que você só precisa de 2 endereços de host IPs).
- Uma sub-rede para 24 hosts.

Temos 4 sub-redes de tamanhos variados, podemos dividir em quatro blocos iguais, porém, perderíamos muitos endereços IPs. Se usarmos um bloco de 64 IPs para a sub-rede onde precisamos de apenas de 2 endereços IP, significa que estamos jogando 62 endereços IP fora.

Você pode estar se perguntando porque isso é importante, afinal, podemos usar endereços de rede privadas como 192.168.1.0. Isso é verdade, mas e quanto à Internet? Não queremos jogar fora endereços públicos, que são extremamente valiosos.

No caso em tela, necessitamos criar sub-redes a partir da rede 192.168.1.0 que atenda aos requisitos mencionados anteriormente, da maneira mais eficiente possível:

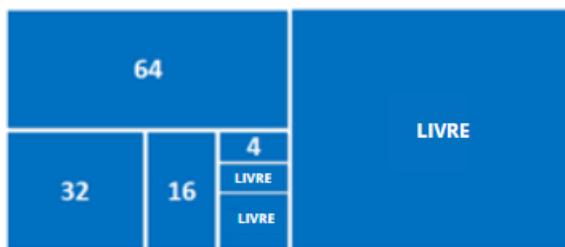
- 12 hosts, sendo que a menor sub-rede que ‘cabe’ esses hosts é um bloco com 16 IPs.
- 44 hosts, sendo que a menor sub-rede que ‘cabe’ esses hosts é um bloco com 64 IPs.
- 2 hosts, sendo que a menor sub-rede que ‘cabe’ esses hosts é um bloco com 4 IPs.
- 24 hosts, sendo que a menor sub-rede que ‘cabe’ esses hosts é um bloco com 32 IPs.

Vamos calcular as sub-redes.

Primeiro, pegamos o bloco com “256” endereços:



Segundo, dividimos o bloco em blocos menos que acabamos de especificar:



Acabamos de salvar alguns endereços IPs valiosos, agora a próxima coisa a fazer é responder às seguintes perguntas:

- Quais são os endereços de rede?
- Quais são os endereços de broadcast?
- Qual é a máscara de sub-rede?
- Quais são os endereços IPs utilizáveis?

Dica: Ao usar VLSM, certifique-se de começar com a maior sub-rede primeiro, ou você terá espaços de endereços sobrepostos.

Vamos responder a essas perguntas. Começaremos com endereçamento de rede:

- Sub-rede 1: (tamanho 64)
 - endereço de rede: 192.168.1.0
- Sub-rede 2: (tamanho 32)
 - endereço de rede: 192.168.1.64
- Sub-rede 3: (tamanho 16)
 - endereço de rede: 192.168.1.96
- Sub-rede 4: (tamanho 4)
 - endereço de rede: 192.168.1.112
- Sub-rede 5: (aqui começa o espaço livre)
 - endereço de rede: 192.168.1.116

Agora podemos preencher os endereços de broadcast:

- Sub-rede 1: (tamanho 64)
 - endereço de rede: 192.168.1.0
 - endereço de broadcast: 192.168.1.63
- Sub-rede 2: (tamanho 32)
 - endereço de rede: 192.168.1.64
 - endereço de broadcast: 192.168.1.95
- Sub-rede 3: (tamanho 16)
 - endereço de rede: 192.168.1.96
 - endereço de broadcast: 192.168.1.111
- Sub-rede 4: (tamanho 4)

- endereço de rede: 192.168.1.112
- endereço de broadcast: 192.168.1.115

Como temos diferentes tamanhos de sub-rede, precisamos calcular a máscara de sub-rede para cada uma delas. Para encontrar a máscara de sub-rede, podemos usar este macete:

256 - Tamanho da sub-rede = máscara de sub-rede

- Sub-rede 1: $256 - 64 = 192$, portanto, a máscara de sub-rede é 255.255.255.192
- Sub-rede 2: $256 - 32 = 224$, então a máscara de sub-rede é 255.255.255.224
- Sub-rede 3: $256 - 16 = 240$, portanto, a máscara de sub-rede é 255.255.255.240
- Sub-rede 4: $256 - 4 = 252$, então a máscara de sub-rede é 255.255.255.252

A única coisa que ainda nos resta a fazer, é preencher os endereços IPs utilizáveis:

- Sub-rede 1: (tamanho 64)
 - endereço de rede: 192.168.1.0
 - primeiro host: 192.168.1.1
 - último host: 192.168.1.62
 - endereço de broadcast: 192.168.1.63
- Sub-rede 2: (tamanho 32)
 - endereço de rede: 192.168.1.64
 - primeiro host: 192.168.1.65
 - último host: 192.168.1.94
 - endereço de broadcast: 192.168.1.95
- Sub-rede 3: (tamanho 16)
 - endereço de rede: 192.168.1.96
 - primeiro host: 192.168.1.97
 - último host: 192.168.1.110
 - endereço de broadcast: 192.168.1.111
- Sub-rede 4: (tamanho 4)
 - endereço de rede: 192.168.1.112
 - primeiro host: 192.168.1.113
 - último host: 192.168.1.114
 - endereço de broadcast: 192.168.1.115

A mesma lógica se aplica aos endereços de Classe A e B.

1.7 Describe the need for private IPv4 addressing

Esse talvez seja o tópico mais curto de toda prova, sendo direto, podemos definir da seguinte forma:

- Os endereços IPs públicos são **usados na Internet**.
- Os endereços IPs privados **são usados nas redes locais**, e não podem ser usados na Internet.

Estes são os intervalos de endereços IPs privados definidos pela RFC-1918:

- **Classe A**: 10.0.0.0 - 10.255.255.255, essa é a maior rede privada, esse range tem capacidade para até **16.777.214** endereços IPs. Essa rede possui o prefixo: **10.0.0.0/8**
- **Classe B**: 172.16.0.0 - 172.31.255.255, essa é a segunda maior rede privada, esse range tem capacidade para até de **1.048.576** endereços IPs. Essa rede possui o prefixo: **172.16.0.0/12**
- **Classe C**: 192.168.0.0 - 192.168.255.255, essa é a menor rede privada, esse range tem capacidade para até de **65.536** endereços IPs. Essa rede possui o prefixo: **192.168.0.0/16**

Decore esses intervalos, eles serão cobrados.

O endereço IP 192.168.1.1, que usamos até agora em nossos exemplos, se enquadra na **classe C**, e é um endereço IP privado! Esta faixa de endereço IP é a mais comum, pois é utilizada praticamente em todas redes domésticas e roteadores **SOHO** (Small Office Home Office).

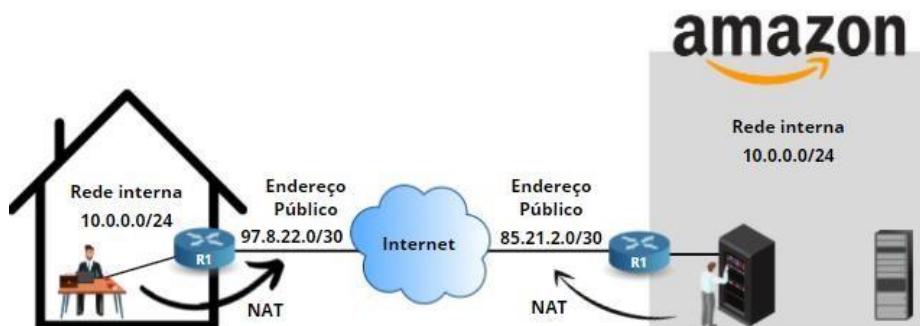
Agora que já sabemos o que são endereços privados, vamos analisar porque precisamos deles!

Na verdade, usamos endereços IP privados pois não há endereços IPv4 suficientes para configurar todos os hosts (dispositivos) que necessitam acessar a internet. É possível repetir endereços IP privados, desde que estejam em redes internas diferentes; por óbvio, os IPs públicos, são sempre únicos.

Há mais dispositivos que necessitam acessar a Internet que endereços IPv4.

Os endereços privados são usados em conjunto com NAT (Network address translation). O NAT, é um protocolo aplicado na camada de rede, que tem como função, fazer a tradução dos endereços IPs da rede local para uma rede externa (geralmente internet). Veremos mais sobre NAT na parte 4.

Vamos para um exemplo: Na minha casa eu utilizo a rede de endereço IP 10.0.0.0/24, então para acessar a rede da Amazon, por exemplo, que utiliza a mesma faixa de IP, precisamos do NAT em ambos os roteadores, ele fará a tradução dos endereços da rede interna para o endereço público dos roteadores:



1.8 Configure and verify IPv6 addressing and prefix

Endereços privados trazem uma série de restrições (não entrarei nessas restrições por enquanto), além de não resolverem por completo a falta de endereços IPv4. Por isso, uma solução definitiva precisava ser criada, e essa solução foi o IPv6.

O protocolo IPv4 possui 32 bits, o que fornece aproximadamente 4.294.467.295 endereços IPs. Houve um erro de projeção nas priscas eras da Internet.

No início, não se imaginava que a Internet teria o tamanho que tem hoje, para se ter uma ideia, era possível obter uma rede inteira Classe A, B ou C, pois ninguém acreditava que necessitariam de tantos endereços IP. Em uma rede classe C, por exemplo, pode-se fornecer um bloco com 256 endereços IPs, na classe B 65.535 endereços IP e na classe A até 16.777.216 endereços IPs.

O erro de projeção foi tão grotesco que empresas como Apple, Microsoft, IBM possuem uma ou mais redes Classe A. Embora sejam realmente empresas gigantescas, elas precisam de mais de 16 milhões de endereços IP? A quantidade de endereços IP desperdiçados foi gigantesca.

Começamos a usar VLSM (Variable Length Subnet Mask) para que pudéssemos usar qualquer máscara de rede e formar sub-redes menores, criamos NAT\PAT para que ter mais endereços IPs privados atrás de um único endereço IP público. Porém a Internet continuou crescendo de forma surpreendente, e apesar de todos esses mecanismos, foi necessário criar algo definitivo, no caso o IPv6.

Se você estiver curioso para saber o que aconteceu para pularmos do IPv4 para IPv6, pesquise sobre a RFC 1918. Lá você encontrará detalhes sobre o IPv5, mas para matar a curiosidade, já deixou o spoiler que o IPv5 foi usado para um projeto experimental chamado “Internet Stream Protocol”, projeto que não deu certo.

Voltando ao IPv6, a principal diferença é que ele possui 128 bits, o que é bem maior que o IPv4 que possui endereçamento de 32 bits. Vamos as comparações: Com IPv4 temos a possibilidade de formar 4 bilhões de endereços. Lembre-se, cada bit adicional dobra o número de endereços IP, logo, com 33 bits saltamos de 4 bilhões de endereços para 8 bilhões e assim sucessivamente até chegarmos em 128 bits. Com 128 bits, conseguimos criar esse número total de endereços:

340.282.366.920.938.463.463.374.607.431.768.211.456

Não me arrisco a pronunciar esse número. Veja abaixo, uma pequena comparação do espaçamento IPv4 e IPv6:

- **IPv6:** 340282366920938463463374607431768211456
- **IPv4:** 4294467295

O **IPv6** oferece alguns recursos que não são possíveis com **IPv4**:

- **Não há tráfego de broadcast:** Não utilizamos mais broadcasts. Em vez disso, usamos multicast. Isso significa que alguns protocolos como o ARP foram substituídos por outras soluções.
- **Stateless Autoconfiguration:** Funciona como um “mini servidor DHCP”. Os roteadores que executam IPv6 são capazes de anunciar o prefixo IPv6 e o endereço do gateway aos hosts, dessa forma, eles se configuraram automaticamente, conseguindo obter acesso a redes externas.
- **Facilidade para reorganizar endereços IP:** Reorganizar (mudar) endereços IPv4 que foram configurados de forma estática é um trabalho braçal intenso, além da provável dor de cabeça que causará. Mas, no IPv6, isso se tornou fácil com **Stateless Autoconfiguration**.
- **Mobilidade:** IPv6 possui suporte integrado para dispositivos móveis. Os hosts poderão se mover de uma rede para outra mantendo o endereço IPv6 atual.
- **Sem NAT/PAT:** Com a quantidade disponível de endereços IPv6, não precisamos mais de NAT ou PAT, cada dispositivo pode ter seu próprio endereço IPv6 público.
- **IPsec:** IPv6 tem suporte nativo para IPsec (protocolo de segurança), sua utilização não é obrigatória, mas ele está embutido no protocolo.
- **Cabeçalho aprimorado:** o cabeçalho IPv6 é mais simples e não requer checksum. Ele também possui um rótulo de fluxo (**flow label**), com esse rótulo é possível verificar rapidamente se determinado pacote pertence àquele fluxo ou não.
- **Ferramentas de migração:** IPv4 e IPv6 **não** são **compatíveis**, por isso foram criadas várias ferramentas de migração. Geralmente são técnicas de encapsulamento, com elas é possível transportar protocolos IPv6 em redes IPv4 (ou vice-versa). A execução de IPv4 e IPv6 simultaneamente é chamada de **dual stack** (pilha dupla).

O formato do endereço IPv6 é totalmente diferente do formato do endereço IPv4:

X: X: X: X: X: X: X onde X é um campo **hexadecimal** de **16 bits**.

No IPv6 não são utilizados números decimais como no IPv4, utilizamos números hexadecimais. Eis um exemplo de um endereço IPv6 real:

2041: 1234: 140F: 1122: AB91: 564F: 875B: 131B

Transformando números decimais e binários em hexadecimal

Em redes é normal usarmos valores binários, decimais e hexadecimais. Dois bons exemplos em que usamos valores hexadecimais são endereços MAC e endereços IPv6.

Para trabalharmos com endereços IPv6, é fundamental compreender como transformamos números hexadecimais em binários, hexadecimais em decimais e vice versa.

No sistema decimal contamos de 1 a 10, no sistema hexadecimal contamos de 1 a F:

Decimal	Hexadecimal	Binário
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Até agora sem mistérios ou dificuldade! Porém, se você quiser calcular de binário para hexadecimal, há um pequeno macete que vai te ajudar. Vamos transformar o número decimal 255 em binário:

128	64	32	16	8	4	2	1
1	1	1	1	1	1	1	1

Se quisermos converter esse número para hexadecimal, precisamos cortar os 8 bits em duas partes de 4 bits (4 bits também são conhecidos como **nibble**).

Primeiro nibble:

1	1	1	1
---	---	---	---

Segundo nibble:

1	1	1	1
---	---	---	---

Agora vamos converter os **nibbles** de binários para decimais:

8	4	2	1
1	1	1	1

Os dois os **nibbles** são iguais, e em cada um deles fazemos a conta: $8 + 4 + 2 + 1 = 15$

Agora dê uma olhada na tabela de conversão de decimal para hexadecimal: O número 15 em decimal é igual a “F” em hexadecimal.

Portanto, o valor hexadecimal é **FF**. Os valores hexadecimais, algumas vezes, são escritos da seguinte forma: **0xFF**. Portanto, se você ler “**0x**”, pode concluir que estamos falando de um valor hexadecimal.

Para fixar, vamos transformar o valor decimal 118 em hexadecimal.

Primeiro, convertemos 118 em binário:

128	64	32	16	8	4	2	1
0	1	1	1	0	1	1	0

Vamos dividir nossos 8 bits em dois **nibbles**:

Primeiro **nibble**:

0	1	1	1
---	---	---	---

Segundo **nibble**:

0	1	1	0
---	---	---	---

Agora vamos converter os **nibbles** de binários para decimais:

8	4	2	1
0	1	1	1

O primeiro **nibble** será $4 + 2 + 1 = 7$. O valor decimal **7** é o mesmo em hexadecimal.

8	4	2	1
0	1	1	0

O segundo **nibble** será $4 + 2 = 6$. O valor decimal **6** é o mesmo em hexadecimal.

Portanto, o valor hexadecimal será **0x76**. Vamos fazer mais um exercício para melhor fixação, tente fazer por conta própria antes de olhar a resposta.

Vamos converter 206 decimal para binário e depois para hexadecimal:

128	64	32	16	8	4	2	1
1	1	0	0	1	1	1	0

Ou seja: $128 + 64 + 8 + 4 + 2 = 206$

Vamos dividir nossos 8 bits em dois **nibbles**:

Primeiro **nibble**:

1	1	0	0
---	---	---	---

Segundo **nibble**:

1	1	1	0
---	---	---	---

Agora vamos converter os **nibbles** de binários para decimais:

8	4	2	1
1	1	0	0

O primeiro **nibble** será $8 + 4 = 12$. O valor decimal 12 é **C** em hexadecimal.

8	4	2	1
1	1	1	0

O segundo **nibble** será $8 + 4 + 2 = 14$. O valor decimal 14 é **E** em hexadecimal.

O resultado da transformação do número **206** em hexadecimal é **0xCE**.

Abreviando endereços IPv6

O IPv6 é formado por 128 bits em números hexadecimais, portanto, são bem longos. É possível tornar o endereço mais amigável deixando-o mais curto. Vamos ver alguns exemplos de como funciona esse ‘encurtamento’:

- **Endereço Original:** 2041:0000:140F:**0000:0000:0000**:875B: 131B
- **Endereço abreviado:** 2041:0000:140F::875B:131B

Quando há uma **sequência de zeros**, podemos removê-los **uma única vez**. No exemplo acima, removi toda a parte **0000:0000:0000**. Só podemos fazer isso **uma vez**, o dispositivo preencherá o espaço restante automaticamente com zeros até que tenha completado os 128 bits do endereço.

Esse endereço pode ser reduzido ainda mais:

- **Endereço abreviado:** 2041:**0000**:140F::875B:131B
- **Endereço ainda mais abreviado:** 2041:0:140F::875B:131B

Se houver um “hekteto” com 4 zeros, podemos removê-los e deixar um único zero. O dispositivo IPv6 adicionará os 3 zeros restantes.

Os zeros à esquerda também podem ser removidos:

- **Endereço Original:** 2001:**000 1:000 2:000 3:000 4:000 5:000 6:000 7**
- **Endereço abreviado:** 2001:1:2:3:4:5:6:7

Quando removemos esses zeros, obtemos um endereço IPv6 bem curto e amigável.

Resumo:

- Uma sequência inteira de zeros pode ser removida, porém, isso só pode ser realizado uma única vez.
- 04 zeros contínuos podem ser removidos, deixando apenas um único zero.
- Os zeros à esquerda também podem ser removidos.

Prefixo IPv6

Os endereços IPv6 possuem algo semelhante a máscara de sub-rede do IPv4, mas em vez de digitar 255.255.255.0, usamos **comprimento do prefixo** (prefix length). Nada melhor que um exemplo para demonstrar o que estou falando:

2001:1111:2222:3333::/64

É praticamente o mesmo que usarmos 192.168.1.1/24. O número depois da /, é o número de bits que usamos para o prefixo. No exemplo acima, significa que 2001:1111:2222:3333 é o prefixo de 64 bits, e todo ‘o espaço’ que vier depois dele pode ser usado para o endereçamento de hosts.

Vamos aprender como calcular o prefixo no IPv6.

Abaixo, um endereço IPv6 que pode ser atribuído a um host:

2001:1234:5678:1234:5678:ABCD:EF12:1234/64

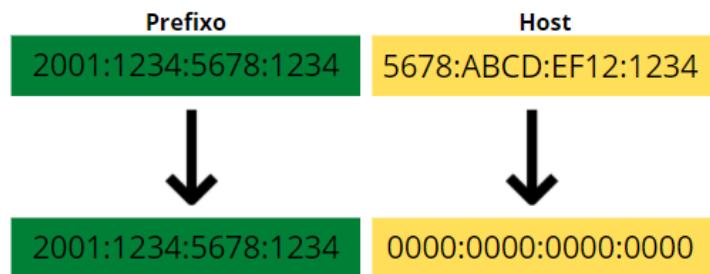
Qual parte desse endereço IPv6 é o prefixo e qual parte identifica o host?

Prefixo	Host
2001:1234:5678:1234	5678:ABCD:EF12:1234

Como usamos um **/64**, significa que os primeiros 64 bits formam o prefixo. Cada caractere hexadecimal representa 4 bits binários, o que significa que o prefixo é: **2001:1234:5678:1234**, esta parte possui 16 caracteres hexadecimais: $16 \times 4 = 64$ bits.

O restante do endereço IPv6 identifica o host: **5678:ABCD:EF12:1234**

Já sabemos que o “**2001:1234:5678:1234**” é a parte do prefixo, mas precisamos escrever o prefixo corretamente, adicionando zeros no final deste prefixo para que ele se torne um endereço de 128 bits novamente:



Agora nós temos um prefixo válido: **2001:1234:5678:1234:0000:0000:0000:0000/64**, e como visto anteriormente é possível encurta-lo para que fique mais amigável. Vamos remover essa sequência de zeros:

2001:1234:5678:1234::/64

Essa é a maneira mais curta de escrever esse prefixo.

Um velho e sábio ditado nos diz que: “A repetição é a chave do aprendizado”, seguindo esse princípio, nada melhor que fazer mais um exemplo para fixação:

Vamos trabalhar com o endereço: **3211::1234:ABCD:5678:1010:CAFE/64**

Antes de começarmos os cálculos para descobrir o prefixo, perceba que o endereço está abreviado (observe o ::). Precisamos adicionar os zeros até completarmos um endereço com 128 bits:

3211:0000:0000:1234:ABCD:5678:1010:CAFE/64

Sabemos que o comprimento do prefixo é de 64 bits, então vamos para os cálculos: Um único caractere hexadecimal representa 4 bits binários, portanto, os primeiros 16 caracteres hexadecimais são o prefixo:

3211:0000:0000:1234

Agora vamos adicionar zeros ao final do endereço para torná-lo um endereço de 128 bits, lembrando de escrever no final o tamanho do prefixo:

3211:0000:0000:1234::/64

Vamos encurtar um pouco mais esse prefixo suprimindo as sequências de zeros:

3211:0:0:1234::/64

Lembre-se sempre da regrinha: 04 zeros em sequência podem ser substituídos por um único zero.

Infelizmente, dependendo do tamanho do prefixo, os cálculos podem ser bem mais difíceis que os exemplos acima. Até agora tralhamos com prefixos /64, porém, se o prefixo for quebrado, como /53, o cálculo se torna um pouco mais chato.

Cada caractere hexadecimal representa 4 bits binários, por isso é fácil realizar esses cálculos quando o comprimento do prefixo é um múltiplo de 16, afinal, 16 bits binários representam 4 caracteres hexadecimais.

Eu sei, é melhor desenhar:



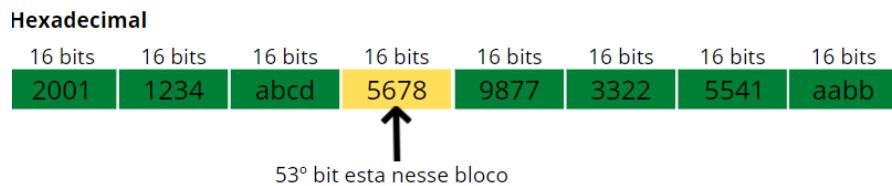
Portanto, em um prefixo com 64 bits, temos 4 “blocos” com 4 caracteres hexadecimais cada, o que facilita o cálculo. Quando o comprimento do prefixo é um múltiplo de 4, o cálculo ainda é simples, pois o limite será um único caractere hexadecimal.

Porém, quando o comprimento do prefixo não é um múltiplo de 16 ou 4, as coisas se tornam um pouco mais complexa, e precisamos fazer algumas conversões a mais para números binários. Vamos para um exemplo, utilizando o prefixo /53!

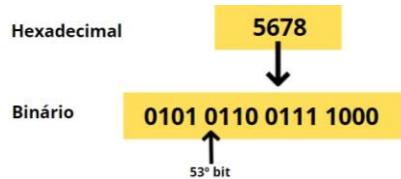
2001:1234:abcd:5678:9877:3322:5541:aabb/53

Através desse endereço IPv6 vamos fazer os cálculos e descobrir qual endereço é o prefixo desta rede.

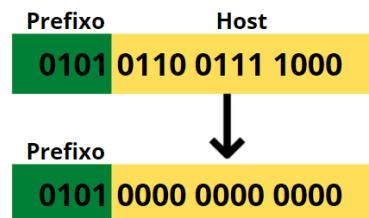
O primeiro passo, é determinar em qual “bloco” está localizado o 53º bit:



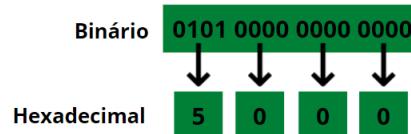
Em algum lugar do bloco mais claro está o 53º bit. Para encontrarmos o prefixo, teremos que transformar esses caracteres hexadecimais para binários:



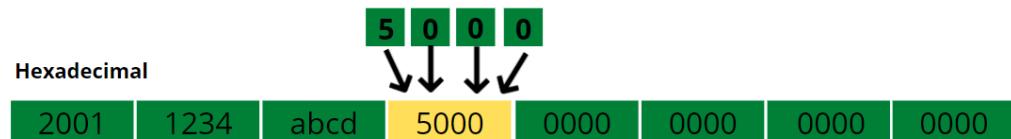
Já sabemos qual é o 53º bit, ele é a fronteira que define o que é “prefixo” e o que é “host”:



O próximo passo é definirmos os bits do host como 0, para que apenas o prefixo permaneça. Por fim, convertemos o binário de volta ao hexadecimal:



Por último, vamos colocar o bloco de volta no seu lugar de origem e definir todos os outros bits de host para 0:



Depois de uma série de idas e vindas, encontramos o prefixo: **2001:1234:abcd:5000::/53**. Na verdade, não é difícil, mas é trabalhoso.

Configurando endereços IPv6 em dispositivos Cisco

Roteadores Cisco não vem com roteamento IPv6 habilitado. Isso significa que para trabalharmos com roteamento IPv6, precisamos antes de tudo habilitá-lo no roteador. Para isso, precisamos entrar no modo de configuração global, e digitarmos o comando: **Ipv6 unicast-routing global**, esse comando habilita o IPv6 no roteador.

Vamos para um exemplo:

```
R1 (config) # ipv6 unicast-routing
R1 (config) #int Gi0/0
R1 (config-if) # ipv6 address 2001:0BB9:AABB:1234::/64 eui-64
```

Observe que no endereço, colocamos o parâmetro **eui-64**, mais para frente, na parte **1.9.f**, há uma explicação específica desse parâmetro.

Para verificarmos o endereço IPv6, usamos o comando **show ipv6 interface Gi0/0**:

```
R1#show ipv6 interface Gi0/0
GigabitEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::201:42FF:FE65:3E01
  No Virtual link-local address(es):
  Global unicast address(es):
    2001:BB9:AABB:1234:201:42FF:FE65:3E01, subnet is 2001:BB9:AABB:1234::/64[EUI]
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF65:3E01
  MTU is 1500 bytes
  ....
```

Aprendemos como configurar e verificar a configuração básica, agora, é hora de aprendermos os diferentes tipos de endereçamento no IPv6.

1.9 Compare IPv6 address types

Há vários tipos diferentes de endereçamento IPv6, cada um com uma função diferente, alguns possuem semelhanças com os endereçamentos do IPv4, como endereços públicos e privados, e alguns são totalmente novos como o anycast.

A partir de agora, vamos ver todos que são cobrados no CCNA.

1.9.a Global unicast

Os endereços unicast IPv6 global, são semelhantes aos endereços públicos do IPv4. Ou seja, são endereços que podem ser usados na Internet.

A grande diferença para o IPv4, é que a quantidade possível de endereços IPv6 é tão grande, que podemos usar endereços unicast globais em qualquer dispositivo, sem necessidade de NAT\PAT, etc.

A **IANA** (Internet Assigned Numbers Authority, ou, "autoridade para atribuição de números de internet", é a organização mundial que supervisiona a atribuição global dos números na Internet - entre os quais estão os números das portas, os endereços IP, sistemas autônomos, servidores-raiz de domínio DNS e outros recursos relativos aos protocolos de Internet), atribuiu o range **2000::/3** para distribuição de endereços globais.

O endereço IPv6 global, consiste em duas partes:

- **Subnet ID** – Possui 64 bits de comprimento. Contém o prefixo da rede, e a ID da sub-rede.
- **Interface ID** - Também possui 64 bits de comprimento, normalmente é composto pelo endereço MAC da interface.

Eis um resumo gráfico:

3 bits	45 bits	16 bits	64 bits
001	Prefixo de roteamento global	Identificação da Sub-rede	Identificação da Interface

1.9.b Unique local

Os endereços locais exclusivos (unique local), exercem a mesma função que os endereços privados no IPv4. Podemos utilizar esses endereços livremente nas LANs, porém, não são podem ser usados na Internet.

Um endereço IPv6 local é construído anexando uma string hexadecimal de 40 bits que é gerada aleatoriamente ao prefixo **FD00::/8**. O campo da sub-rede e o ID da interface são criados da mesma forma que os endereços IPv6 globais.

Eis um resumo gráfico:

8 bits	40 bits	16 bits	64 bits
FD	Identificação global	Identificação da Sub-rede	Identificação da Interface

1.9.c Link local

Os endereços de link local são uma novidade implementada pelo IPv6. Esses endereços funcionam apenas no **link local**, ou seja, nunca são encaminhamos para outras sub-redes, eles são usados apenas para enviar e receber pacotes IPv6 naquela sub-rede específica.

Quando ativamos o IPv6 em uma interface, o dispositivo automaticamente cria um endereço de link local. Esse endereço é utilizado para descoberta de vizinhos (substituindo o protocolo ARP) e também como o próximo salto na tabela de roteamento.

Os endereços de link local têm o prefixo **FE80::/10**. Eis um resumo gráfico:

64 bits	64 bits
FE80:0000:0000:0000	Identificação da Interface

1.9.d Anycast

O endereço anycast é outro recurso novo trazido pelo IPv6. Com ele, o mesmo endereço IP pode ser atribuído a vários dispositivos e anunciado em um protocolo de roteamento.

Quando um pacote é enviado para um endereço anycast, ele é entregue na interface mais próxima. Não há prefixo específico para endereços anycast, portanto, qualquer endereço unicast que escolhermos usar em mais de um dispositivo se torna automaticamente um endereço anycast. Só precisamos ‘informar’ ao dispositivo que o endereço será usado para anycast.

1.9.e Multicast

No IPv6, o multicast é usado por protocolos de roteamento e para tráfego de usuário, quando é necessário a comunicação com alguns hosts, mas não todos. O prefixo escolhido para os endereços multicast é o **FF::/8**:

Eis uma representação gráfica:

8 bits	4 bits	4 bits	112 bits
FF	Flags	Escopo	Identificação do Grupo

Os primeiros 8 bits indicam que temos um endereço multicast. Os próximos 4 bits, são usados para definir sinalizadores (flags), estes são usados para sinalizar elementos do multicast. Os bits do escopo são usados para informar o “escopo” desse tráfego multicast, por exemplo, ele pode indicar que o tráfego multicast deve ser restrito ao link-local, unique local ou global.

Abaixo, uma tabela com alguns dos endereços IPv6 multicast mais comuns:

Endereço	Função
FF02::1	Destinado para todos os hosts no segmento de rede local.
FF02::2	Destinado para todos os roteadores no segmento de rede local.
FF02::5	Destinado para os roteadores que estão rodando OSPFv3.
FF02::6	Destinado para roteadores OSPFv3 DR.
FF02::9	Destinado para roteadores RIPng.
FF02::A	Destinado para roteadores EIGRP

Alguns desses endereços são semelhantes aos seus correspondentes multicast no IPv4. Por exemplo, no IPv4 usamos 224.0.0.5 e 224.0.0.6 para OSPF, já no IPV6 usamos FF02::5 e FF02::6 para OSPFv3, no RIPv2 usamos 224.0.0.9 e no IPv6 FF02::9 para RIPng.

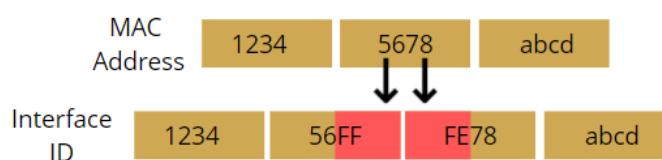
1.9.f Modified EUI 64

EUI-64 (Extended Unique Identifier) é um método que podemos usar para configurar automaticamente endereços IPv6, nesse método, o dispositivo IPv6 usará o endereço MAC de sua interface para gerar um endereço de interface exclusivo de 64 bits. Porém, um endereço MAC possui 48 bits e o ID da interface (endereço ou identificação da interface) é de 64 bits. Por isso, os bits que faltam precisam ser preenchidos, eis o método que o dispositivo utiliza:

1. O endereço MAC é dividido em duas partes.
 2. É inserido “FFFE” entre as duas partes do endereço MAC, assim chega-se ao valor de 64 bits.
 3. Inverte-se o 7º bit da interface ID.

Confuso? Vamos desenhar um exemplo passo a passo para que as coisas fiquem claras:

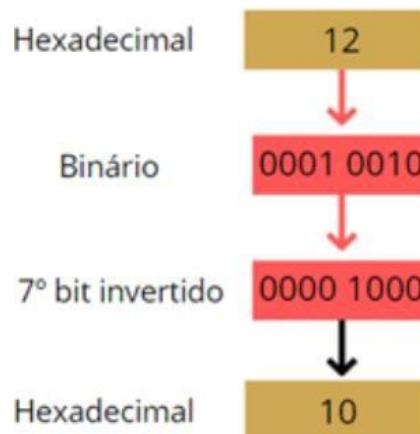
Vamos usar o endereço MAC 1234.5678.ABCD, e transformá-lo em uma interface ID:



Rpare como dividimos o endereço MAC e colocamos o FFFE no meio.

Ainda falta uma etapa, que é inverter o 7º bit. Primeiro, devemos converter os dois primeiros caracteres hexadecimais do primeiro byte em binário, depois encontrar o 7º bit e invertê-lo. Isso significa que se o 7º bit for o número 0 ele se tornará 1 e vice e versa.

O 7º bit representa o **universal unique bit (bit único universal)**. Um endereço MAC sempre terá o 7º bit definido como 0, o que significa que na maioria das vezes o EUI-64 mudará o 7º bit de 0 para 1 (salvo raras exceções quando o MAC address é alterado de forma manual, sim, é possível).



Pegamos os dois primeiros caracteres hexadecimais do primeiro byte, que era o número “12”, e convertemos para binário. Depois, invertemos o 7º bit de 1 para 0, e convertemos para hexadecimal novamente. A EUI-64 interface ID será este:

Antes	1234	56FF	FE78	abcd
Depois	1034	56FF	FE78	abcd

Agora que já aprendemos como o EUI-64 funciona, vamos rever como configuramos um endereço IPv6 em um roteador. Vamos usar o endereço **2001:1234:5678:abcd::/64**

```

Router(config)#interface fastEthernet 0/0
Router(config-if)#ipv6 address 2001:1234:5678:abcd::/64 eui-64
  
```

A interface está configurada com o endereço IPv6 e utilizamos o EUI-64 no final. Essa é a sintaxe correta para configura-lo. Observe abaixo, o endereço IPv6 que foi criado (Preste sempre atenção no comando):

```

Router#show interfaces fastEthernet 0/0 | include Hardware
Hardware is Gt96k FE, address is c200.185c.0000 (bia c200.185c.0000)
  
```

```

Router#show ipv6 interface fa0/0
FastEthernet0/0 is up, line protocol is up
IPv6 is enabled, link-local address is FE80::C000:18FF:FE5C:0
No Virtual link-local address(es):
Global unicast address(es):
2001:1234:5678:ABCD:C000:18FF:FE5C:0, subnet is 2001:1234:5678:ABCD::/64 [EUI]
  
```

Observe o **C000:18FF:FE5C:0**, esse é o endereço MAC que compõe a interface ID após aplicação do EUI-64. Bem no meio do endereço, temos o **FFFE**. Observe também, que o 7º bit estava em “**C200**” do endereço MAC, e para compor o endereço IPv6, esse sétimo bit foi invertido, e é por isso que ele aparece como “**C000**”.

Quando utilizamos EUI-64 em uma interface que não possui endereço MAC, o roteador seleciona o endereço MAC da menor interface do roteador.

IPv6 é um tanto complexo para quem não gosta de números, cálculos, etc. Mas é um tópico extremamente importante. Caso tenha ficado alguma dúvida, releia todo o capítulo e faça os exercícios.

1.10 Verify IP parameters for Client OS (Windows, Mac OS, Linux)

Embora faça parte do blueprint da prova, esse tópico ainda não foi cobrado (e sinceramente, duvido que será).

Cada Sistema Operacional possui comandos específicos para verificarmos os parâmetros de rede, abordaremos os principais de cada um deles.

Windows

No Windows, basta digitar ‘prompt de comando’ na barra de pesquisa, e na tela do DOS digitar o comando **ipconfig/all**:

```
C:\WINDOWS\system32>ipconfig/all

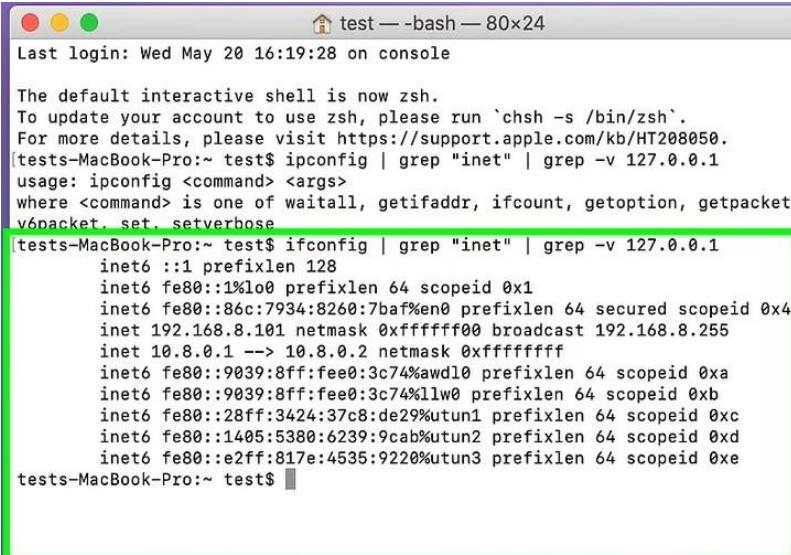
Windows IP Configuration

Wireless LAN adapter Wi-Fi:

  Connection-specific DNS Suffix . :
  Description . . . . . : Intel(R) Dual Band Wireless-AC 8265
  Physical Address . . . . . : 00-50-56-C0-00-08
  DHCP Enabled. . . . . : Yes
  Autoconfiguration Enabled . . . . . : Yes
  IPv6 Address. . . . . : 2001:470:c090:11:a07c:1a25:fbd7:83c0(PREFERRED)
  Temporary IPv6 Address. . . . . : 2001:470:c090:11:15a8:22e6:f5b5:e9bd(PREFERRED)
  Link-local IPv6 Address . . . . . : fe80::a07c:1a25:fbd7:83c0%20(PREFERRED)
  IPv4 Address. . . . . : 10.0.11.20(PREFERRED)
  Subnet Mask . . . . . : 255.255.255.0
  Lease Obtained. . . . . : Thursday, July 11, 2019 6:57:44 PM
  Lease Expires . . . . . : Saturday, July 13, 2019 1:10:40 AM
  Default Gateway . . . . . : fe80::def7:19ff:fe1d:6149%20
                               10.0.11.1
  DHCP Server . . . . . : 10.0.11.1
  DHCPv6 IAID . . . . . : 158626045
  DHCPv6 Client DUID. . . . . : 00-01-00-01-22-BB-AC-E5-8C-16-45-91-4D-D5
  DNS Servers . . . . . : 10.0.3.4
                           208.67.220.220
                           208.67.222.222
  NetBIOS over Tcpip. . . . . : Enabled
```

Mac OS

No Mac, abra o ‘Utilitários’ na pasta de Aplicativos e utilize o comando **ifconfig**:



```
test — bash — 80x24
Last login: Wed May 20 16:19:28 on console

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
tests-MacBook-Pro:~ test$ ipconfig | grep "inet" | grep -v 127.0.0.1
usage: ipconfig <command> <args>
where <command> is one of waitall, getifaddr, ifcount, getoptoption, getoptpacket,
v6packet, set, setverbose
tests-MacBook-Pro:~ test$ ifconfig | grep "inet" | grep -v 127.0.0.1
inet6 ::1 prefixlen 128
inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
inet6 fe80::86c:7934:8260:7baf%en0 prefixlen 64 secured scopeid 0x4
inet 192.168.8.101 netmask 0xffffffff broadcast 192.168.8.255
inet 10.8.0.1 --> 10.8.0.2 netmask 0xffffffff
inet6 fe80::9039:8ff:fee0:3c74%wd10 prefixlen 64 scopeid 0xa
inet6 fe80::9039:8ff:fee0:3c74%llw0 prefixlen 64 scopeid 0xb
inet6 fe80::28ff:3424:37c8:de29%utun1 prefixlen 64 scopeid 0xc
inet6 fe80::1405:5380:6239:9cab%utun2 prefixlen 64 scopeid 0xd
inet6 fe80::e2ff:817e:4535:9220%utun3 prefixlen 64 scopeid 0xe
tests-MacBook-Pro:~ test$
```

Linux

No Linux, basta pesquisar por ‘terminal’ e digitar o comando **ifconfig**:

```
[root@srv-lnx-01 network-scripts]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 172.27.1.107 netmask 255.255.0.0 broadcast 172.27.255.255
        inet6 fe80::71e2:58d6:f81e:63a0 prefixlen 64 scopeid 0x20<link>
          ether 00:15:5d:33:c1:81 txqueuelen 1000 (Ethernet)
            RX packets 136716 bytes 27952091 (26.6 MiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 1249 bytes 86560 (84.5 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
          loop txqueuelen 1000 (Loopback Local)
            RX packets 456 bytes 39608 (38.6 KiB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 456 bytes 39608 (38.6 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

1.11 Describe wireless principles

Redes sem fio estão em todos os lugares, e a tendência, é que redes sem fio expandam cada vez mais. Com a popularização dos notebooks e celulares nos últimos anos, a necessidade do wireless, seja em casa ou em empresas, aumentou muito.

Wireless traz consigo uma série de desafios para os administradores de redes, seja em termos de segurança (um ponto crítico) ou mesmo na qualidade da navegação para o usuário final, entre esses desafios os mais comuns são:

- **Cobertura:** É necessário analisar a localização dos pontos de acesso, as frequências que serão usadas, levar em consideração no projeto que diferentes materiais afetam a qualidade do sinal de forma singular.
- **Interferência:** Vários dispositivos e aparelhos domésticos utilizam frequências de 2,4 e 5 GHz (as mesmas utilizadas por dispositivos wireless), dessa forma, interferência se torna praticamente inevitável. O principal malefício da interferência é o enfraquecimento da qualidade do sinal, o que provoca lentidão.
- **Privacidade:** Os dados “voam no ar”, o que significa que não temos como proteger a camada física, por isso, é imperioso a utilização de autenticação e criptografia.
- **Regulamentação:** Cada país tem seus próprios regulamentos, ex: força do sinal, frequências permitidas, etc.

Além disso, por definição, há muitos outros fatores que podem ocasionar erros com o sinal wireless:

- **Reflexão:** Ocorre quando o sinal sem fio é refletido em algum material. O metal, é um bom exemplo. É muito difícil obter o sinal wireless através de um teto de metal ou elevador, pois o sinal simplesmente não passa, ele reflete.
- **Dispersão:** Ocorre quando o sinal sem fio atinge alguma superfície e “se quebra” em várias partes, deixando o sinal original fraco.
- **Absorção:** Ocorre quando o sinal sem fio encontra algum material que o absorve, água e o corpo humano são bons exemplos disso, pois, eles absorvem fortemente o sinal wireless. A absorção é terrível, pois faz com que o receptor receba o sinal com baixíssima intensidade.

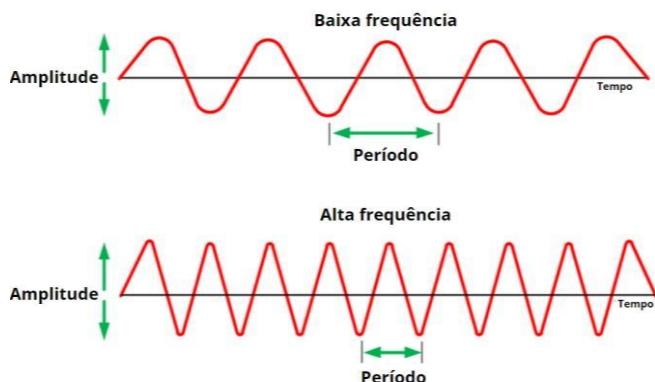
1.11.a Nonoverlapping Wi-Fi channels

Se você reparar o roteador que você tem em casa, verá que ele possui dois padrões diferentes de comunicação. O primeiro é o mais comum: **2,4 GHz**. Já o segundo, mesmo que seja tão antigo quanto o primeiro, está se tornando mais popular somente agora: **5 GHz**. Vamos entender o que esses números significam!

Geralmente o termo GHz (giga-hertz) está associado a desempenho. Um processador de 5 GHz, por exemplo, é bem mais potente que um de 2,4 GHz. Nesse caso podemos dizer que quanto maior o número, melhor será o produto.

No caso do WiFi, existe uma pequena diferença: Esses números não expressam a quantidade de ciclos por segundo como no caso dos processadores, por exemplo. Aqui, estamos falando das frequências de operação do sinal de rádio transmitido pelo roteador. Assim, não existe um valor necessariamente melhor, já que são dois padrões de transmissão diferentes.

Frequências mais altas proporcionam taxas de dados mais altas; quanto mais alta a frequência, mais "ondas" haverá em um determinado ciclo de tempo:



O IEEE definiu para o wireless a nomenclatura 802.11, e nele, nós temos diversos padrões: 802.11a, 802.11b, 802.11ge e 802.11n. Abaixo, uma tabela comparativa entre os diversos padrões:

	802.11a	802.11b	802.11ge	802.11n
Frequência	5 GHz	2,4 GHz	2,4 GHz	2,4 e 5 GHz
Canais	23	3	3	Depende
Taxa de dados	Até 54Mbit	Até 11Mbit	Até 54Mbit	Até 300 – 600 Mbit

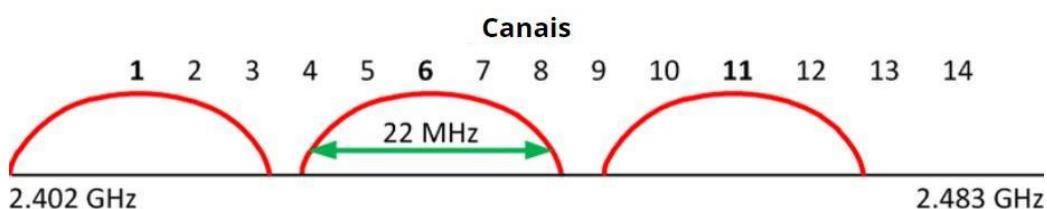
Vamos dissecar a tabela acima.

O padrão 802.11b é o mais lento, a velocidade máxima que ele alcança é 11 Mbits. Já o 802.11a, é o mais rápido, ele opera na banda de 5GHz. Com ele, é possível alcançar velocidade de até 54Mbit. A mesma técnica que foi usada para 802.11a é utilizada para 802.11g, só que utilizando a banda de 2,4 GHz. Isso torna 802.11b e 802.11g compatíveis, uma vez que operam na mesma frequência.

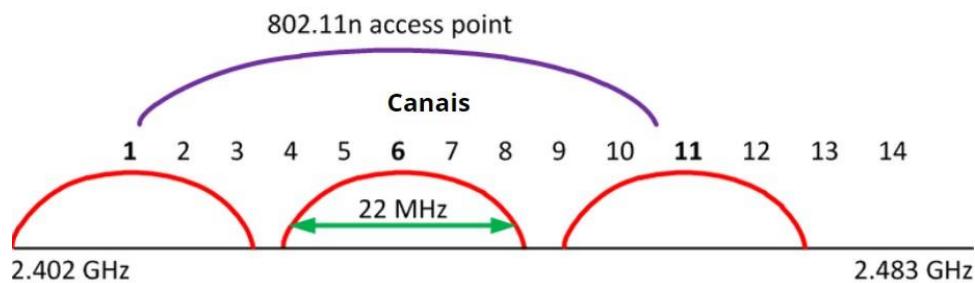
O padrão 802.11n é um pouco diferente de todos os outros, ele foi completamente alterado para propiciar um desempenho melhor, com ele, é possível obter velocidades que variam de 300 a 600Mbit, sendo que ele consegue operar em ambas as frequências.

No padrão 802.11b e 802.11g, há apenas 3 canais. Se você já configurou um roteador wireless, mesmo que seja o roteador da sua casa, deve ter notado que havia a possibilidade de configurar 11 canais. Sim, há essa possibilidade, mas na realidade, existem apenas 3 canais que não se sobrepõem.

Os canais **1, 6 e 11** podem ser usados sem interferência, os outros canais podem ser usados, mas precisamos ter certeza de que há **5 canais** de 'distância' para os outros, senão, haverá interferência. Esta é uma das razões pelas quais em algumas situações a faixa 5 GHz é mais indicada, pois há muito mais espaço. A desvantagem é que a cobertura na banda de 2,4 GHz é muito maior e melhor que na de 5 GHz.



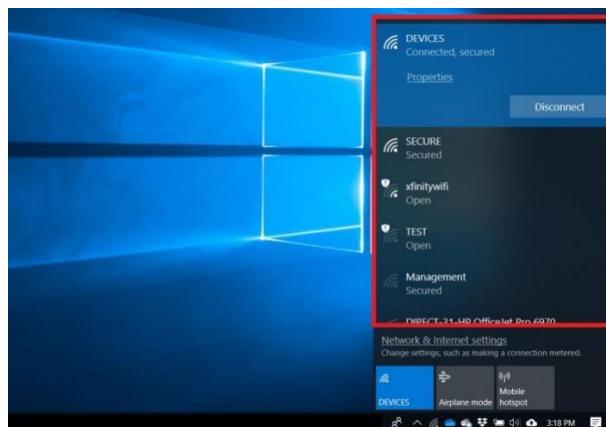
O padrão 802.11n é sui generis, pois pode operar tanto 2,4 GHz quanto em 5 GHz, além disso, o número de canais que ele abrange pode variar. Isso ocorre porque o 802.11n pode fazer o “channel-bonding”. Basicamente, com o “channel-bonding” é possível obter o dobro do desempenho, porém, utilizando o dobro da largura de banda. O inconveniente dessa tecnologia é que sobra pouco espaço para outros dispositivos sem fio. Se utilizarmos o “channel-bonding” na frequência de 2,4 GHz, ele ocuparia tanto espaço que não haveria mais canais sem sobreposição!



1.11.b SSID

Assim como as redes cabeadas, as redes sem fio possuem diferentes topologias físicas e lógicas. O padrão **802.11** relata esses diferentes **Service Set (conjuntos de serviços)**. Um **Service Set** descreve como determinado grupo de dispositivos sem fio deve se comunicar entre si.

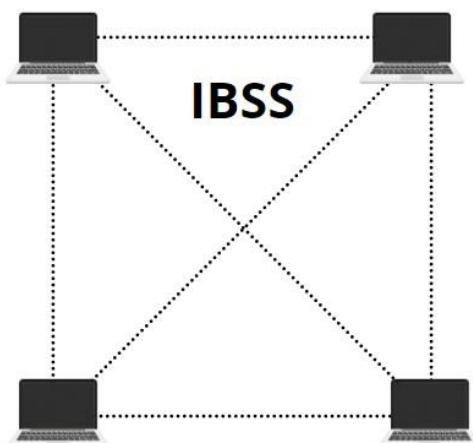
Cada conjunto de serviços utiliza o “mesmo” ‘**Same Service Set Identifier (SSID)**’. O **SSID** é o nome “amigável” da rede sem fio, literalmente é o nome das redes que você vê em seu computador.



IBSS

O ‘**Independent Basic Service Set (IBSS)**’, permite que dois ou mais dispositivos sem fio se conectem **diretamente**, sem a necessidade de um **Access Point** (ponto de acesso - **AP**), outra nomenclatura utilizada para o **IBSS** é **rede ad hoc**.

Nesse tipo de rede, um dos hosts anuncia o SSID como se fosse um AP, e os demais hosts se conectam nele. Geralmente o **IBSS** é usado para transferência de arquivos entre dois ou mais laptops, smartphones ou tablets, sem que estes precisem se conectar à rede sem fio por meio de um Access Point.



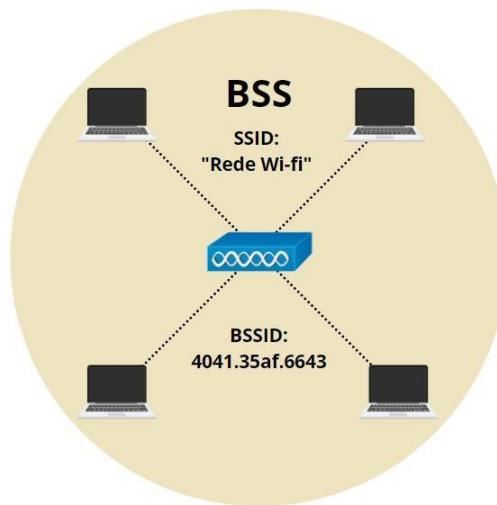
Modo de infraestrutura

Em um modo de infraestrutura, todos os dispositivos sem fio são conectados a um dispositivo central, ou seja, a um ponto de acesso (Access Point). Neste modo, todos os dados passam pelo Access Point. O padrão 802.11 descreve diferentes Service Sets. Vamos conhecê-los:

Basic Service Set (BSS)

Em um Basic Service Set (**BSS**), os clientes se conectam a uma rede sem fio **por meio de um Access Point**. A infraestrutura **BSS** é utilizada na maioria das redes sem fio, justamente pela sua simplicidade. A ideia por trás de um BSS é que o AP seja responsável pela gerência da rede, seja negando ou concedendo permissão para os clientes wireless ingressarem na rede.

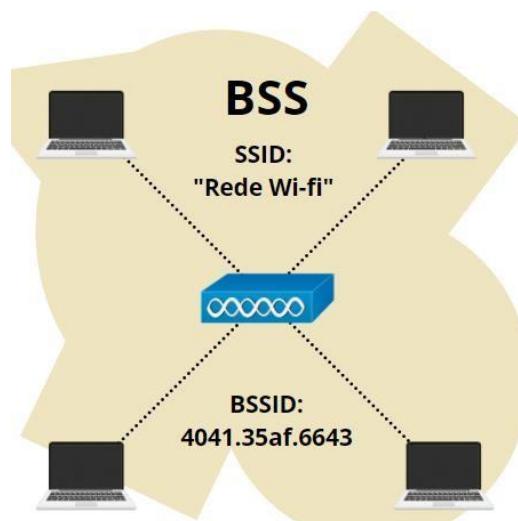
O **BSS** utiliza um único canal para todas as comunicações, sendo assim, os clientes e os APs usam o mesmo canal para transmitir e receber dados.



SSID é o nome “amigável” da rede sem fio, ele sequer **precisa ser exclusivo**.

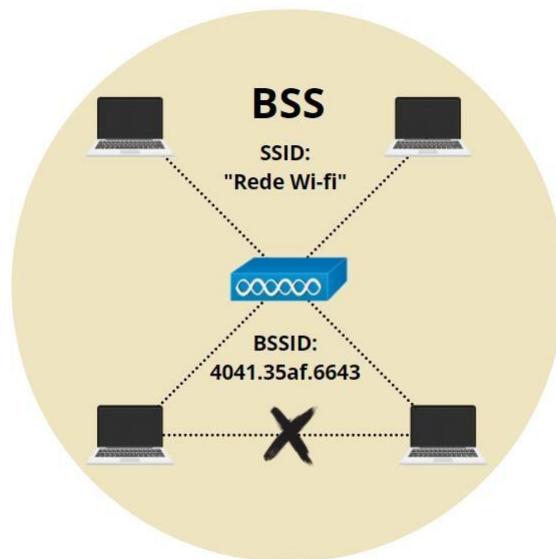
Poucas pessoas sabem, mas o AP também anuncia o ‘**Basic Service Set Identifier (BSSID)**’. O **BSSID** é o endereço MAC do AP, como sabemos o endereço MAC é único, portanto, esse é um **endereço único que identifica o Access Point**. O range de sinal do AP define o tamanho do BSS, essa área é chamada de **Basic Service Area (Área de Serviço Básico (BSA))**.

No diagrama anterior, BSA forma um lindo círculo. Esse cenário aconteceria somente se instalássemos o Access Point no meio do deserto, sem nada em volta. Mas, no mundo real, onde Access Point são instalado em prédios e casas, o BSA fica mais ou menos com a forma abaixo:



Um host para ingressar no BSS, deve enviar uma ‘association request’ (solicitação de associação) ao Access Point, que pode negar ou aceitar a solicitação. Se o ingresso for permitido, esse host poderá se juntar ao BSS, então ele será chamado de **wireless cliente** (cliente sem fio) ou **802.11 station (STA)**.

Todo o tráfego de um cliente wireless obrigatoriamente passa pelo Access Point, mesmo que esse tráfego seja destinado a outro cliente wireless.



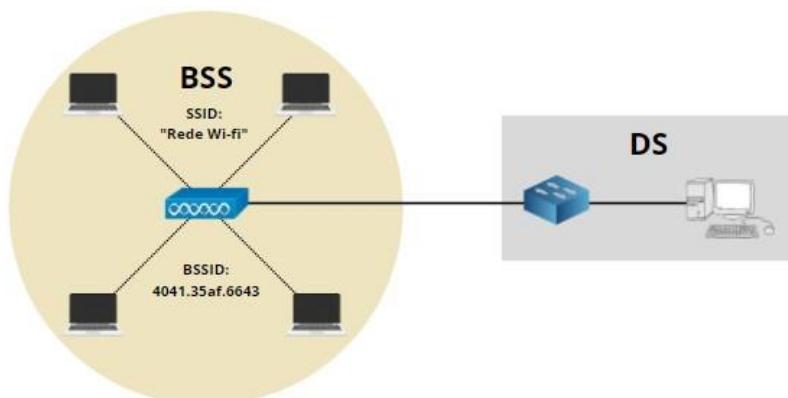
Tudo tem que passar pelo AP, afinal o AP é o ponto central de gerenciamento, e isso limita o tamanho do BSS. O alcance do sinal do AP define o **limite do BSS**.

Distribution System (DS)

Um **BSS** é uma rede independente com um único AP. Observe que nos diagramas anteriores não há conexão com uma rede cabeadas.

A maioria das redes sem fio, entretanto, é uma extensão de uma rede cabeadas. Um Access Point suporta conexões com e sem fio. O padrão 802.11 chama a rede cabeadas de ‘**distribution system**’ (sistema de distribuição (DS)).

O Access Point faz a ponte entre os quadros Ethernet da rede cabeadas com a rede sem fio, permitindo que os dados trafeguem pelas duas redes.



Cada rede sem fio possui um **BSSID** exclusivo. O **BSSID** é baseado no endereço MAC, a maioria dos fornecedores (incluindo Cisco) incrementa o último dígito do endereço MAC para criar um **BSSID** exclusivo.

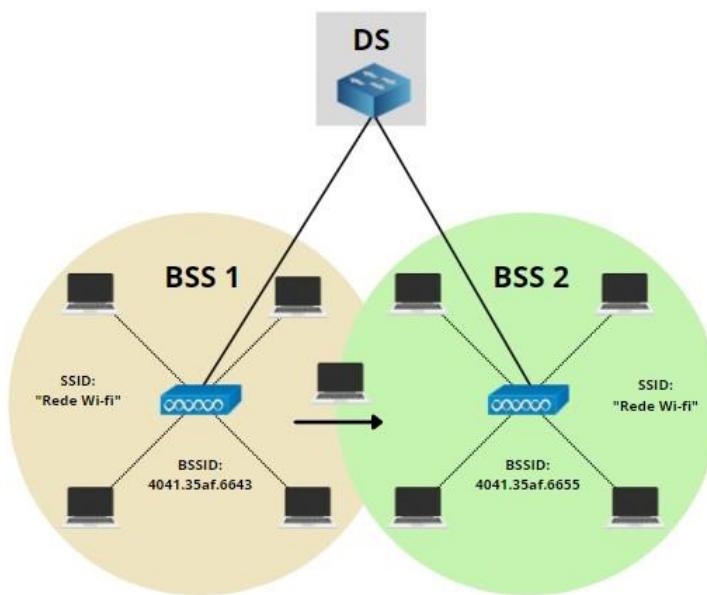
Extended Service Set (ESS)

O BSS usa um único AP, por dois motivos em especial um único Access Point pode não ser suficiente:

- **Cobertura:** O sinal de um único AP geralmente não é suficiente para cobrir toda área de um prédio ou mesmo um andar. Nesse caso, precisamos de vários APs para termos conexão sem fio em todos os lugares.
- **Largura de banda:** Um AP usa um único canal e a conexão wireless é half-duplex. Quanto mais clientes wireless ativos, menor será a taxa de transferência. Isso também depende das taxas de dados suportada. Um host que esteja no limite do sinal do BSA, ainda poderá alcançar o AP, mas as taxas de dados que ele obterá serão baixas. Um host que esteja próximo ao AP alcançará altas taxas de transferência de dados. Assim, um cliente wireless distante acabará exigindo mais “tempo de antena” (airtime), reduzindo a largura de banda para todos os outros clientes.

Por isso, a necessidade de criar uma rede sem fio maior.

Essa rede sem fio maior é formada quando conectamos vários APs à rede cabeada. Esses APs trabalharão juntos para criar uma grande rede wireless que pode se estender por vários andares ou mesmo por um prédio inteiro. O usuário verá apenas um **único SSID**, embora estejamos usando vários Access Points. **Cada AP usa um BSSID diferente**, portanto, nos bastidores, o host vê vários APs disponíveis para conexão. Chamamos essa topologia com vários APs de **Extended Service Set (ESS)**.



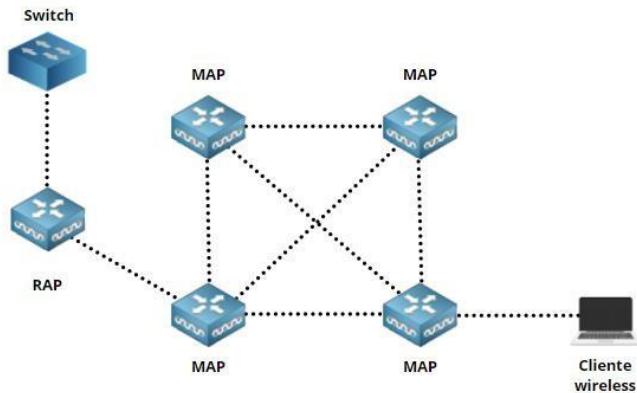
Como dito anteriormente, todos esses APs trabalham em conjunto. Por exemplo, o usuário ao associar seu celular a um AP poderá andar tranquilamente pelo ambiente e mesmo assim não se desconectará. O celular irá “pular” automaticamente de um AP para outro AP. Chamamos isso de **roaming**. Para tornar essa experiência perfeita, precisamos de uma sobreposição entre os APs. Nessa sobreposição, cada AP deverá usar um canal diferente para evitar interferência.

Mesh Basic Service Set (MBSS)

Aqui o desafio é um pouco maior. Imagine que você precisa fornecer rede sem fio para uma área muito grande, como uma cidade! Não é viável simplesmente sair conectando cada AP a uma rede cabeada.

A solução mais recomendada é construir uma rede mesh (em malha), essa rede é conhecida como **Mesh Basic Service Set (MBSS)**. Com uma rede em malha, **conectamos um access point diretamente a outro access point**. Os APs de malha geralmente trabalham com várias frequências. Uma dessas frequências será para tráfego de **backhaul** (ligação entre um ap e outro); e outra frequência será destinada para manter os clientes wireless em outro canal.

Pelo menos um desses APs tem que estar conectado à rede cabeada; esse access point é chamado de **Root AP (RAP)**. Os demais APs são chamados de **APs de malha (MAP)** e são conectados por meio do **backhaul** wireless.



Existem vários caminhos possíveis para um **MAP** chegar à rede cabeada através do **RAP**, portanto, precisamos de um protocolo que encontre o melhor caminho e elimine a possibilidade de ocorrerem loops na rede. O IEEE possui o padrão **802.11s** para redes mesh, porém, grande parte dos fabricantes possuem soluções proprietárias. A Cisco, por exemplo, possui o **Adaptive Wireless Path Protocol (AWPP)**.

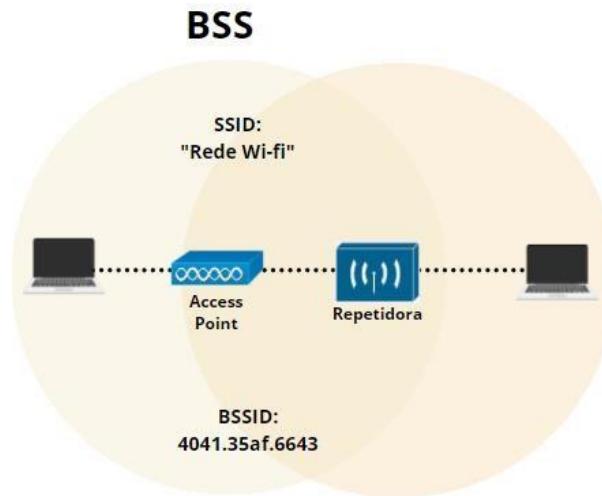
AP Modes

Até agora, falamos apenas sobre **conjuntos de serviços**. Alguns APs também oferecem suporte a diferentes modos que não estão diretamente relacionados à infraestrutura. Vamos estudar os modos de AP mais comuns.

Repetidor

Quando existe a necessidade de cobrir uma grande área, geralmente utiliza-se um ESS. Um ESS, no entanto, requer conexões cabeadas. Se for impossível conectar o AP a uma rede cabeada, há a possibilidade de configurarmos em **modo repetidor (repeater mode)**.

Um repetidor wireless recebe o sinal e retransmite. Isso permite que dispositivos sem fio que não estejam próximos suficientes ao AP se conectem à rede.

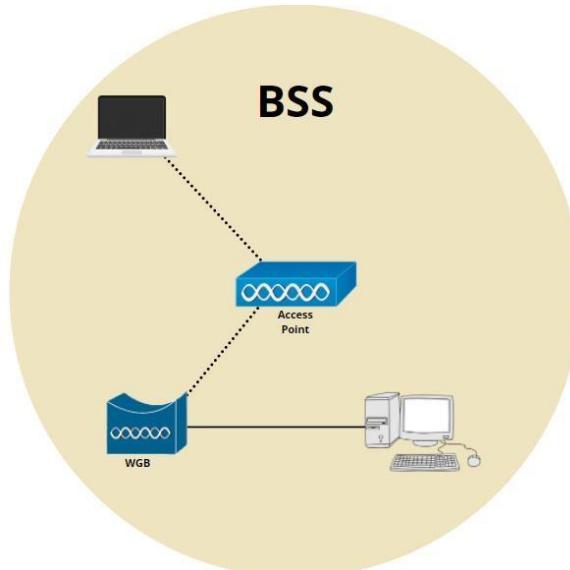


Deve haver uma sobreposição (overlap) entre o tamanho da célula do AP e da repetidora. Para um desempenho ideal, deve ser cerca de 50%. Se a repetidora operar em uma única frequência, ela receberá e transmitirá no mesmo sinal do AP. Nesse caso, o AP também receberá o sinal retransmitido. Como o wireless é half-duplex, adicionar uma repetidora reduzirá a taxa de transferência disponível em cerca de 50%.

Para contornar isso, alguns repetidores trabalham com duas frequências. Eles recebem em um canal (o mesmo do AP) e retransmitem em outro.

Workgroup Bridge

Caso haja um dispositivo com fio que precisa se conectar a uma rede wireless, porém, esse dispositivo não possui tecnologia wireless? Por exemplo, impressoras, sistemas de ponto de venda (PoS) antigos, etc. Nesse caso, temos a opção **Work group bridge (ponte do grupo de trabalho (WGB))**. O **WGB** possui uma conexão com fio para conectar o dispositivo que só opera de forma cabeada e uma conexão sem fio para conectá-lo a rede wireless.



Existem dois tipos de WGBs:

- **Universal workgroup bridge (uWGB)**: WGB universal suporta apenas um único cliente com fio.
- **Workgroup Bridge (WGB) ou Workgroup Bridge Mode** é uma tecnologia proprietária da Cisco que oferece suporte a vários clientes com fio.

Resumo:

Essa foi uma parte bem longa, o melhor a fazer é um pequeno resumo para facilitar os estudos:

- Um conjunto de serviços (**Service Set**) descreve como um grupo de dispositivos wireless se comunicam.
- Cada Service Set usa um **SSID**. Literalmente, o **SSID** é o nome da rede wireless que aparece para os usuários.
- Existem diferentes conjuntos de serviços:
 - **IBSS**: É uma rede ad-hoc para comunicação direta entre dispositivos wireless, aqui não existe um ponto central de conexão.
 - **BSS**: Rede que utiliza um AP. Toda a comunicação passa pelo AP.
 - **DS**: A rede cabeada que conectamos à rede wireless.
 - **ESS**: Rede sem fio com vários APs. Frequentemente é exigida por dois motivos:
 - Um único AP não pode cobrir um andar ou prédio inteiro.
 - A conexão sem fio é half-duplex, portanto a largura de banda disponível depende do número e da localização dos dispositivos sem fio.
 - **MBSS**: Uma rede em malha, útil quando não se pode conectar todos os APs a uma rede com fio. Os APs criam um backbone sem fio.
- Diferentes modos de AP:
 - **Repetidor**: Repete o sinal de um AP, útil se você não pode usar um **ESS**, mas deve ser evitado.
 - **Workgroup Bridge**: Permite que você conecte um dispositivo com fio como se fosse um cliente wireless.

1.11.c RF

Agora, falaremos sobre as frequências utilizadas pelos dispositivos wireless, por motivos didáticos, tudo que você precisa saber sobre essa parte foi explicado no tópico '**1.11.a Nonoverlapping Wi-Fi channels**'.

Para não ficarmos repetitivos vou colar a tabela comparativa dos padrões definidos pela IEEE:

	802.11a	802.11b	802.11ge	802.11n
Frequência	5 GHz	2,4 GHz	2,4 GHz	2,4 e 5 GHz
Canais	23	3	3	Depende
Taxa de dados	Até 54Mbit	Até 11Mbit	Até 54Mbit	Até 300 – 600 Mbit

Caso tenha ficado alguma dúvida nessa tabela, retorne para o tópico citado acima.

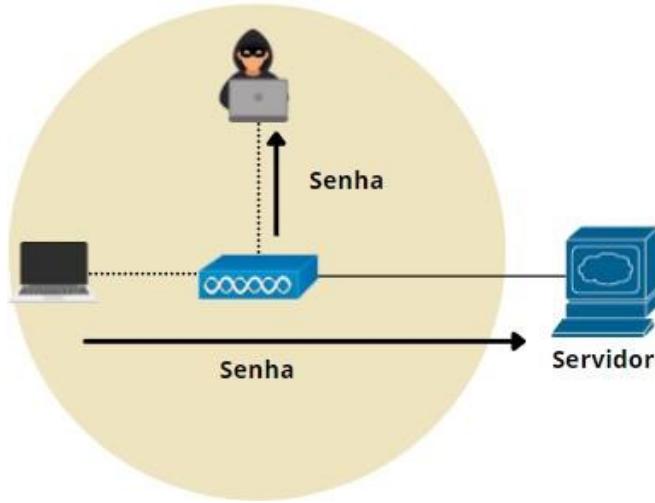
1.11.d Encryption

Neste tópico falaremos sobre os mecanismos de segurança de uma rede sem fio, mas antes de entrarmos no tópico, preciso fazer uma pequena introdução.

Introdução a segurança de rede sem fio

Quando transmitimos dados pelo ar, deixamos de ter uma barreira física, os cabos como proteção extra. Os dados ‘voando’ pelo ar estão ao alcance de qualquer um que esteja dentro da área de abrangência com um dispositivo wireless. Em uma rede cabeada, caso alguém não autorizado consiga acesso a um computador, só terá acesso aos dados direcionados para aquela porta do switch, não verá tráfego multicast ou unicast direcionado a outros usuários. Já na rede wireless, ele terá acesso a todo o tipo de tráfego.

No tópico sobre Service Sets, vimos como os clientes wireless se associam aos APs, nele, aprendemos que todo o tráfego sem fio deve passar pelo AP em vez de trafegar diretamente entre o remetente e o destinatário. Agora, vimos que qualquer pessoa dentro do alcance do AP pode receber esse sinal. Isso é um problema. Por exemplo, imagine que temos um usuário que esteja enviando uma senha para acessar a sua conta bancária:

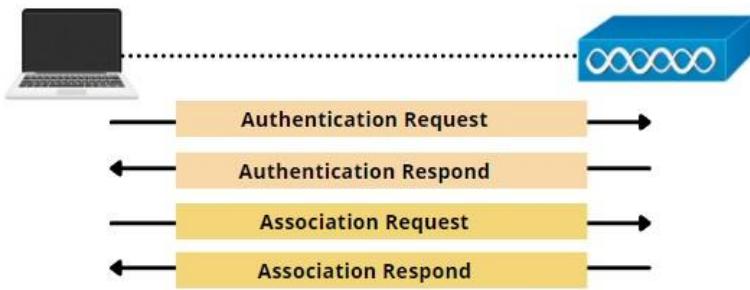


O usuário transmite sua senha para o banco, e como o invasor está ao alcance de nossa rede sem fio, ele poderá capturar essa senha.

Como podemos transmitir dados com segurança pelo ar e garantir que eles permaneçam privados e não sejam adulterados? O padrão 802.11 oferece mecanismos de segurança que fornecem **autenticação, criptografia e integridade**. Neste tópico, teremos uma visão geral desses três itens.

Authentication

Para usar uma rede sem fio, o cliente precisa descobrir o **BSS**. Os APs anunciam **beacons** (*O Beacon é um quadro (frame) de sincronização enviado em períodos constantes pelo roteador wifi. A sua principal função é a de avisar todos os clientes de que a rede está ativa, e também sincronizar a transmissão dos dados trafegados no wireless*) com seu **SSID**, o cliente wireless seleciona a rede que deseja se conectar e então se associa ao AP. Por padrão, a autenticação é aberta, o que significa que todos são bem-vindos.

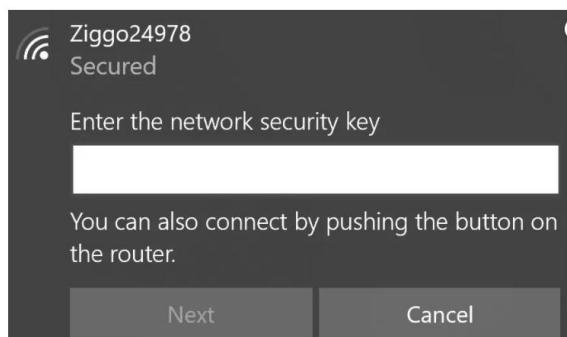


Com pouquíssimas exceções, todas as redes vão solicitar autenticação para permitir que os usuários se conectem à rede wireless, ainda mais em se tratando de uma rede corporativa. Apenas usuários legítimos devem ser capazes acessar os recursos da rede.

Mas, e se houver usuários convidados? Se você deseja oferecer uma rede sem fio para convidados, uma boa solução seria configurar um **segundo SSID, vinculado a uma VLAN com acesso restrito**.

Os APs podem autenticar clientes sem fio **antes que eles se associem ao AP**. Isso mantém pessoas não autorizadas fora da rede sem fio.

Existem muitas opções para autenticação wireless. Provavelmente, você está familiarizado com algumas delas, um exemplo é a **pre-shared key (chave pré-compartilhada)**. A pre-shared key é configurada no AP e qualquer usuário que deseja entrar na rede deverá inserir a **chave pré-compartilhada**.



Essa solução é interessante, porém há um problema: E se alguém roubar um dos dispositivos wireless?

- O invasor terá acesso à chave pré-compartilhada e poderá se conectar à rede wireless a partir de qualquer dispositivo.
- Precisaremos configurar uma nova chave pré-compartilhada no AP, e depois configurá-la em todos os clientes sem fio.

Existem opções de autenticação mais fortes, em que é solicitado nome de usuário e senha. Isso resolve parte do problema caso um dispositivo seja roubado ou uma senha seja vazada para pessoas não autorizadas, pois assim que for identificado qual nome de usuário foi comprometido bastará redefinir a senha.

E quanto ao AP? Imagine que você esteja hospedado em um hotel e há uma rede sem fio com o nome “hóspedes”; é natural presumirmos que se trata da rede sem fio do hotel. Porém, qualquer pessoa pode configurar um AP e usar o SSID “hóspedes”. Como ter certeza que este é um AP legítimo, de propriedade do hotel?

Os clientes wireless salvam um perfil para todas as redes sem fio em que ele já se conectou. Quando ele vir a rede “hóspedes” novamente, ele tentará se autenticar e se associar a ela.

Alguns ataques sem fio usam um AP falso, chamado **rogue ap** (AP vampiro). O **rogue ap** age como um AP normal: transmite **beacons**, responde **probes** e permite associação de clientes. Quando um cliente se associa ao **rogue ap**, o invasor fica

entre o tráfego da rede wireless e a rede cabeadas, podendo assim interceptar todo o tráfego, agindo exatamente da mesma forma que age um AP real. Este tipo de ataque é conhecido como **man-in-the-middle**.

Para evitar o ataque **man-in-the-middle**, o cliente deve autenticar o AP antes mesmo que ele mesmo se autentique no AP.

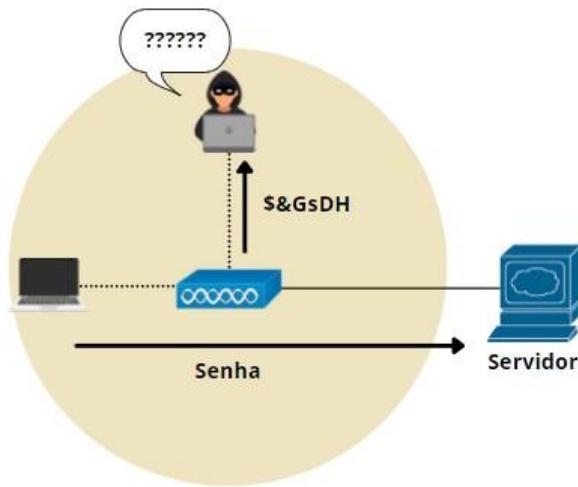


Encryption

Temos um cliente wireless e um AP. Ambos os dispositivos autenticaram um ao outro, portanto, ambos sabem que são dispositivos legítimos. Até agora, tudo bem.

O cliente e o AP autenticaram um ao outro, mas mesmo assim, ainda é possível que esse tráfego seja visto por pessoas que não deveriam.

Se quisermos privacidade, precisamos **criptografar o tráfego**. Para fazer isso, criptografamos a carga útil (**payload**) de todos os quadros que transmitimos e descriptografamos o **payload** dos quadros que recebemos.



Cada SSID suporta apenas **um método de autenticação e criptografia**. Isso significa que todos os clientes sem fio precisam usar o **mesmo modo de autenticação e de criptografia**. Se houver necessidade de utilizar outro modo, deveremos configurar um segundo SSID.

Agora vamos imaginar que temos dois clientes wireless conectados a um AP. Usamos criptografia para que pessoas não autorizadas não consigam decifrar nosso tráfego. Isso significa que o cliente wireless **A** pode descriptografar qualquer dado que o cliente **B** transmite?

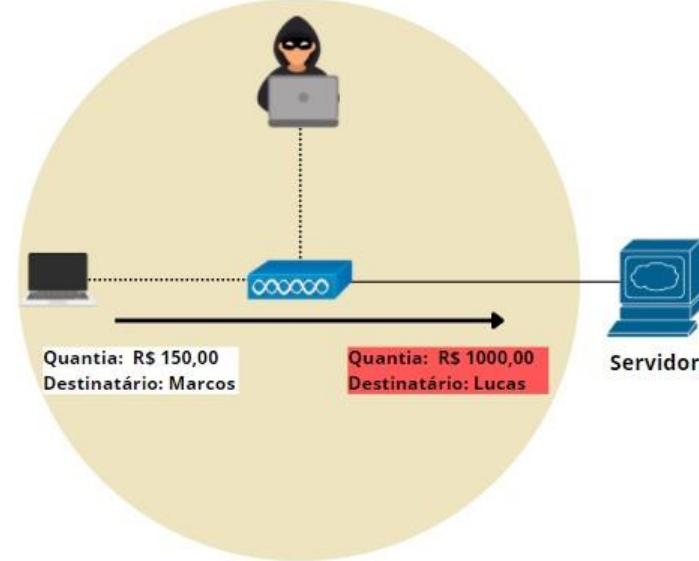
Na verdade, depende! Depende do algoritmo de criptografia que usamos. Ainda assim, idealmente, o AP **deve usar uma chave de criptografia exclusiva para cada cliente wireless**. Com dois clientes wireless, o AP terá duas chaves de criptografia, uma para cada cliente.

E quanto ao tráfego de **broadcast** e **multicast**? Esse tipo de tráfego é enviado para mais de um cliente. O AP também tem uma **group encryption key** (**chave de criptografia de grupo**) que usa quando precisa transmitir tráfego de broadcast ou multicast. Cada cliente sem fio usa a mesma chave de grupo para descriptografar o tráfego.

Integrity

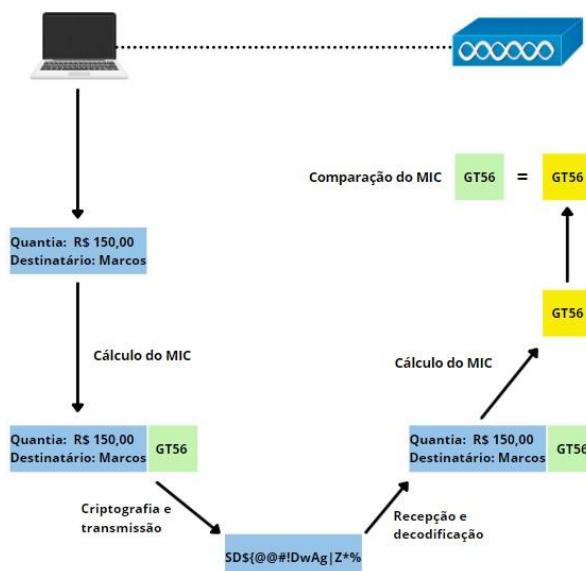
Até agora aprendemos que podemos usar criptografia para evitar que outras pessoas “leiam” o tráfego wireless. O remetente criptografa o tráfego e o receptor o descriptografa.

Mas, e se alguém alterar a mensagem ao longo do caminho? Não há como saber se o que recebemos é descriptografamos é realmente a mensagem original do remetente.



Podemos resolver esse problema com o **Message Integrity Check (MIC - verificação de integridade de mensagem)**. O **MIC** é como um carimbo dentro do quadro de dados criptografado.

O remetente cria um carimbo com base nos dados que deseja transmitir. Quando o receptor descriptografa o quadro, ele também cria um carimbo. Quando os dois carimbos são idênticos, o receptor sabe que os dados não foram adulterados.



Resumo:

- Como transmitimos dados pelo ar, qualquer pessoa dentro do alcance da rede wireless pode ver e capturar nosso tráfego.
- O padrão 802.11 oferece mecanismos de segurança que fornecem:
 - Autenticação
 - Criptografia
 - Integridade
- Existem vários modos de autenticação wireless, onde autenticamos o cliente e o AP.
- O método de autenticação mais comum é a **pre-shared key**, geralmente usada para redes domésticas. Este método não é adequado para redes wireless maiores:
 - Qualquer pessoa com a **pre-shared key** pode ingressar na rede.
 - Quando a **pre-shared key** está comprometida, ela deve ser substituída em todos os dispositivos.
- A autenticação do AP é essencial para evitar que clientes wireless se conectem a um AP não autorizado que esteja se passando pelo **SSID** verdadeiro.
- Para evitar que pessoas não autorizadas espionem o tráfego, usa-se criptografia.
- A verificação de integridade da mensagem ajuda a detectar se alguém alterou a mensagem.
- Cada **SSID** suporta apenas um modo de autenticação e criptografia. Se necessitarmos usar outros modos, devemos configurar um segundo SSID.

Algoritmos de criptografia

Vamos estudar os algoritmos de criptografia, algumas siglas só farão sentido na parte **5.9 Describe wireless security protocols (WPA, WPA2, and WPA3)**.

TKIP

WEP (sigla de "Privacidade Equivalente à de Redes com Fios", foi o pioneiro no assunto de proteção de redes wireless, tendo sido lançado como um padrão de segurança neste tipo de rede em 1997) usa o algoritmo **RC4** para criptografia. O WEP não é seguro, portanto, para garantir que os hardwares mais antigos pudessem usar um método de criptografia confiável, o IEEE desenvolveu o **Temporal Key Integrity Protocol (TKIP)**.

O **TKIP** adiciona os seguintes recursos de segurança:

- **MIC**: A mensagem ganha uma verificação extra de integridade chamada MIC. Essa verificação adiciona um valor de hash a cada quadro. É utilizada para detectar se houve alteração no quadro.
- **TKIP sequence counter**: Este contador fornece um registro de quadros enviados por cada endereço MAC. Usado para evitar que um invasor execute um ataque de repetição em que fique retransmitindo quadros infinitamente.
- **Key mixing algorithm**: Este algoritmo calcula uma chave WEP de 128 bits exclusiva para cada quadro.
- **A longer initialization vector (IV)**: A utilização dessa ferramenta dobra o número de bits para 48 bits, contra os 24 bits normalmente utilizados pelo WEP. Isso dificulta para um invasor utilizar a técnica de força bruta nas chaves WEP.
- **Timestamp**: Adiciona um “timestamp” ao MIC para evitar ataques de repetição. Um ataque de repetição tenta retransmitir um quadro que foi enviado anteriormente.
- **Endereço MAC do remetente**: O MIC inclui o endereço MAC do remetente, este endereço é usado para provar quem é o verdadeiro remetente do quadro.

O TKIP foi uma solução temporária adotada enquanto o IEEE trabalhava no padrão 802.11i. Hoje em dia, o TKIP também tem vulnerabilidades e não deve mais ser usado.

WPA (versão 1) também usa TKIP.

CCMP

CCMP significa ‘**Counter Mode with Cipher Block Chaining Message Authentication Code Protocol**’. Nome gigantesco que consiste em dois algoritmos:

- AES counter mode encryption
- CBC-MAC

O Advanced Encryption Standard (**AES**) é um algoritmo de criptografia amplamente usado, sendo considerado o **método de criptografia mais seguro no momento**.

O Instituto Nacional de Padrões e Tecnologia (NIST) definiu cinco modos de operação para AES. Sendo eles:

- Electronic Code Book (BCE)
- Cipher Block Chaining (CBC)
- Cipher Feedback (CFB)
- Output Feedback (OFB)
- Counter (CTR)

Cipher Block Chaining Message Authentication Code (CBC-MAC) é uma técnica que constrói um código de autenticação de mensagens a partir de uma cifra de bloco. Os dados são criptografados com AES criando assim uma cadeia de blocos. Cada bloco depende da criptografia do bloco anterior.

Antes de usar o CCMP, você precisa verificar se seu dispositivo wireless é compatível com AES e CBC-MAC. O CCMP não oferece suporte para hardwares mais antigos.

O WPA2 usa CCMP, portanto, é possível verificar se o hardware oferece suporte a CCMP, através do símbolo WPA2. O WPA2 está disponível desde 2006, então a maioria dos hardwares atuais oferece suporte a CCMP.

GCMP

802.11ad oferece taxas de dados ainda mais altas do que 802.11ac, por isso, necessita de um método de criptografia mais rápida que o CCMP pode oferecer. O **Galois/Counter Mode Protocol (GCMP)** pode ser executado em paralelo, por isso é mais eficiente e oferece melhor desempenho do que o **CCMP**.

O CCMP usa AES de 128 bits. Com CGMP podemos usar AES de 128 ou 256 bits. Se você usar 802.11ac, poderá usar GCMP opcionalmente.

O GCMP usa dois algoritmos:

- AES counter mode encryption
- Galois Message Authentication Code (GMAC).

AES counter mode é um dos cinco modos AES. Este modo é amplamente utilizado porque oferece alto desempenho. GMAC é uma variante somente de autenticação do modo Galois-Counter e usado para a verificação de integridade da mensagem.

WPA3 usa GCMP.

Resumo:

- O padrão 802.11 original suportava apenas autenticação aberta e WEP.
 - WEP utiliza criptografia RC4.
 - O WEP é inseguro, portanto, outros algoritmos de criptografia e integridade são necessários.
 - Hardware mais antigo suportavam apenas RC4, não conseguindo trabalhar com algoritmos de criptografia mais fortes como AES.
- O TKIP foi uma solução temporária utilizada em substituição ao WEP.
 - Assim como o WEP, o TKIP usa o RC4, mas foram realizadas alterações para contornar problemas que afligiam o WEP.
 - WPA (versão 1) usa TKIP.
- CCMP usa AES com counter mode encryption e CBC-MAC.
 - Se você deseja usar CCMP, seu dispositivo wireless tem que oferecer suporte para criptografia AES.

- Os dispositivos mais antigos não conseguem utilizar o CCMP, pois há incompatibilidade com hardware.
- WPA versão 2 usa CCMP.
- GCMP oferece melhor desempenho que CCMP.
 - Precisamos de maior desempenho porquê 802.11ad oferece taxas de dados maiores que 802.11.ac
 - O GCMP oferece AES de 128 ou 256 bits.
 - WPA versão 3 usa GCMP.

1.12 Explain virtualization fundamentals (virtual machines)

Entraremos agora em um dos melhores tópicos para estudar, virtualização e máquinas virtuais.

Virtual Machines

Antigamente, um servidor físico executava um único sistema operacional com algumas poucas aplicações. Os processadores tinham apenas um núcleo, não era possível expandir a memória, etc. Hoje, os servidores possuem vários núcleos, grande poder de processamento e quantidade significativa de memória. Basicamente, esse era o esquema de organização de um servidor:



Esquema de organização de um servidor

Um dos motivos pelos quais a virtualização se tornou popular é que os servidores eram subutilizados. Com o aumento do poder computacional, percebeu-se que era possível executar vários sistemas operacionais e várias aplicações em um único servidor físico, eliminando a ociosidade que os servidores ficavam na maior parte do tempo. A máquina virtual foi criada justamente para ocupar esse tempo ocioso e utilizar todo potencial computacional do hardware. Uma máquina virtual é idêntica no seu funcionamento a um servidor físico, possuindo seu próprio hardware virtual (CPUs, memória, armazenamento) que roda em um **hypervisor**.

O **hypervisor** é o software de virtualização que é executado no servidor físico. É aqui que criamos máquinas virtuais e configuramos quantos núcleos de CPU, memória, armazenamento cada máquina virtual terá.

Existem dois tipos de hypervisor:

Tipo 1: Este hypervisor é executado diretamente no hardware, o que significa que você pode atribuir mais recursos às máquinas virtuais pois não há sistema operacional para consumir recursos. Os exemplos são VMware ESXi, Citrix Xen, KVM e Microsoft Hyper-V.

Tipo 2: Este hypervisor é executado em um sistema operacional como Microsoft Windows, Apple MacOS ou Linux. Normalmente usamos um hypervisor tipo 2 em desktops ou laptops. Os hypervisores mais populares são Virtualbox da Oracle e VMWare Workstation. Caso você nunca tenha utilizado uma máquina virtual, experimente o Virtualbox que é gratuito e fácil de utilizar.



Tipo 1



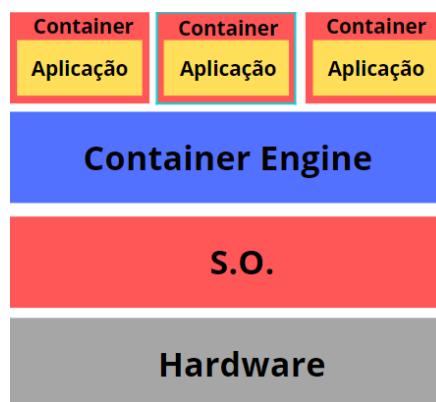
Tipo 2

Até mesmo um hypervisor tipo 1 tem uma espécie de sistema operacional. Por exemplo, VMWare usa um kernel proprietário chamado VMkernel, esse S.O. é muito mais leve que um sistema operacional “completo” como o Windows ou qualquer distribuição Linux.

Containers

Os contêineres são conhecidos como *máquinas virtuais leves* (*light-weight virtual machines*), mas na verdade, não se parecem em nada com uma máquina virtual.

Um contêiner é um “pacote” que contém apenas a aplicação e suas dependências, nada mais. Este pacote é armazenado como uma imagem de contêiner. Os contêineres são executados em cima de um **container engine**, este sim, executado em um sistema operacional. Iniciar um contêiner é muito rápido, pois o sistema operacional já está em execução. Os contêineres são isolados uns dos outros pelo **container engine**.



As principais características da conteinerização são:

- **Pequeno:** A imagem do contêiner contém apenas a aplicação e suas dependências, nada mais. Isso resulta em uma imagem pequena. Por exemplo, se criarmos uma imagem de contêiner do Freeradius (servidor RADIUS de código livre), essa imagem terá apenas 5 MB, e nela, conterá tudo o que precisamos para rodar o Freeradius.
- **Rápido:** Não é necessário iniciar um servidor virtual com sistema operacional virtual. Ativar um contêiner é tão rápido quanto iniciar uma aplicação qualquer, leva apenas alguns milissegundos.
- **Portabilidade:** A imagem do contêiner contém tudo que a aplicação precisa. É possível criar uma imagem de contêiner em uma máquina e enviar para outra.
- **Isolamento:** Os contêineres são executados no mesmo sistema operacional, mas são isolados uns dos outros. Se um contêiner falhar, não afetará outros contêineres.
- **Escalabilidade:** É possível adicionar mais contêineres para escalar horizontalmente.
- **Imutabilidade:** As imagens de contêiner são construídas com base em um “blueprint”. A imagem Freeradius que mencionei anteriormente é um contêiner do Docker; o blueprint é chamado de Dockerfile. Se esse código for alterado, será criada uma nova imagem de contêiner.

Existem algumas desvantagens no uso de contêineres:

Os contêineres são isolados uns dos outros no nível de processo, o que o torna menos seguro que uma máquina virtual, que é totalmente isolada.

Outro problema de segurança é que os contêineres são baseados em snapshot. Depois de criada, não é possível modificar a imagem do contêiner. Se por algum motivo, o conteúdo do contêiner precisar ser atualizado, será necessário reconstruir uma nova imagem. Em um servidor, mesmo que virtual, instalações e atualizações são fáceis de serem realizadas, sem a necessidade de reinstalar todo o servidor.

Para resumir, uma pequena tabela comparativa com os principais pontos estudados até agora:

	Máquina Virtual	Container
Overhead	Depende da capacidade do hardware e de quantas máquinas virtuais estão instaladas no servidor.	Dificilmente ocorre pelas próprias características do container e por ser executado diretamente no hospedeiro.
Performance	Pequena queda no desempenho por conta do hardware virtual.	Quase nenhuma queda em comparação a execução nativa da aplicação.
Sistema Operacional	Cada máquina virtual possui seu próprio S.O.	Utiliza o S.O. do host.
Tempo de inicialização	Leva um certo tempo, pois tem que iniciar o S.O e a aplicação.	Inicialização quase instantânea.
Armazenamento	Desperdiça uma grande quantidade de espaço, já que precisa armazenar o S.O.	As imagens são pequenas, pois temos apenas a aplicação e suas dependências.
Isolamento	Totalmente isolado.	Isolado apenas no nível de processo, o que causa uma certa vulnerabilidade.

Introdução a Cloud Computer

A abordagem da computação em nuvem é diferente do conceito tradicional de computação, na computação em nuvem tudo é entregue como um serviço, seja por provedores externos ou pelo próprio departamento de TI da própria empresa.

Antes de entrarmos em computação em nuvem, vamos fazer uma breve viagem no tempo para entendermos como tudo começou e como chegamos até aqui.

Servidores Bare Metal

Antes de existir virtualização de servidores, havia apenas servidores **bare metal**. Servidores Bare metal são servidores físicos que executam apenas um sistema operacional como o Windows Server. Nessa época, as CPUs possuíam poucos núcleos e havia pouca capacidade para memória RAM.

Servidores são semelhantes aos computadores que temos em casa, a grande diferença é que eles foram projetados para trabalharem o tempo todo, 24 horas por dia durante 07 dias na semana, o popular 24/7. Por isso, possuem um hardware mais confiável, o que inclui, discos rígidos e fonte de alimentação redundantes.

Servidores podem ter formatos de torre, semelhante aos computadores que temos em casa, ou em formato de switches, sendo estes, projetados especialmente para Data Center.



Servidor em formato de torre



Servidor projetado para racks



Gabinetes de servidores



Servidor de rack HP Proliant

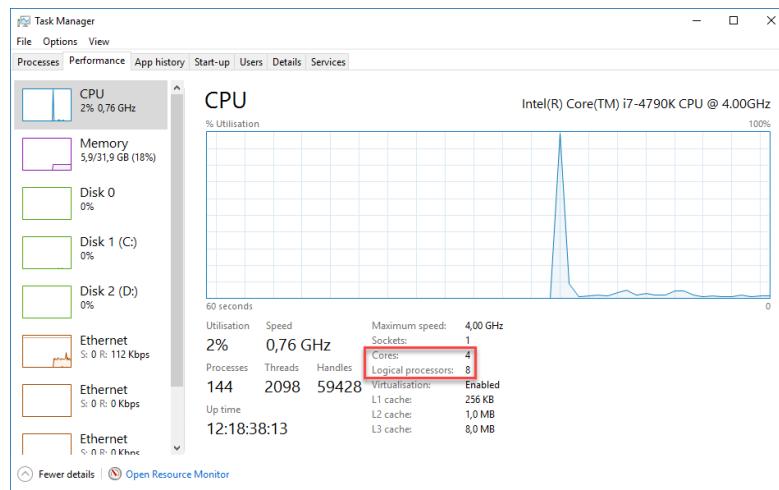
Servidores Bare Metal ainda são muito utilizados em escritórios e pequenas empresas, pois servem como:

- Servidor de email,
- Servidor de arquivos,
- Controlador de domínio

Em ambientes como um data center, em que precisamos economizar espaço, os servidores de gabinete são os mais utilizados.

Virtualização de servidores

Nos últimos anos, o número de núcleos em um único processador cresceu rapidamente, a 15 anos atrás, os custos para ter um processador com mais de um núcleo era gigantesco. Hoje temos processadores de 20 núcleos com preço totalmente acessível. Soma-se a isso uma técnica chamada hyper threading, onde para cada núcleo físico é possível ter 02 núcleos virtuais.



Exemplo de servidor com 4 núcleos físicos e 8 virtuais

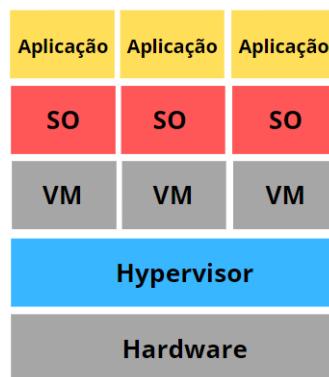
A quantidade de memória RAM que os hardwares aceitam também aumentou muito. Hoje, um servidor físico tem muito mais recurso de CPU e RAM do que é necessário para rodar um único sistema operacional. Por esse motivo, como dito anteriormente, a virtualização se tornou tão popular, pois podemos ‘colocar’ várias máquinas virtuais em um único servidor físico.

Todo o hardware da VM (virtual machine) é virtualizado, o que inclui: CPU, HD, placa de rede, memória RAM, etc. Cada máquina virtual terá a quantidade de memória ram, HD, CPU que configurarmos.

Abaixo, o esquema de um servidor Bare Metal que executa um único Sistema Operacional:



Observe que temos o hardware que executa o sistema operacional e as aplicações. Abaixo um esquema de um servidor que utiliza máquinas virtuais:

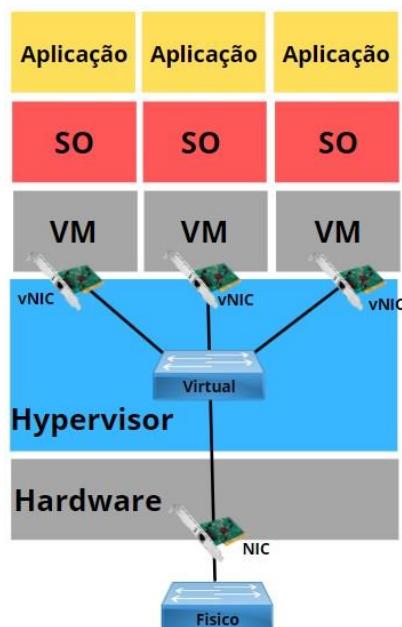


Acima do hardware há o hypervisor que executa as máquinas virtuais e os sistemas operacionais.

Caso haja um problema irrecuperável no hardware, todas as máquinas virtuais desaparecerão, por isso é importante adotar soluções de backup.

Rede virtual

Os servidores físicos possuem pelo menos uma **NICs** (Network interface cards, ou simplesmente placa de rede) conectada a um switch. Já as máquinas virtuais, que possuem hardware virtualizado (o que inclui NIC, chamadas de vNIC) se conectam a rede através de switches virtuais. Observe o diagrama abaixo:



Cada máquina virtual está conectada através de sua vNIC a um switch virtual. O Switch virtual é conectado a um switch físico através da interface de rede física do servidor que o Hypervisor está conectado.

OBS: Na imagem acima temos apenas uma NIC ligando o servidor físico ao switch, porém, em ambiente de produção é normal que esses servidores sejam conectados a mais de um switch para garantir redundância e aumento da largura de banda.

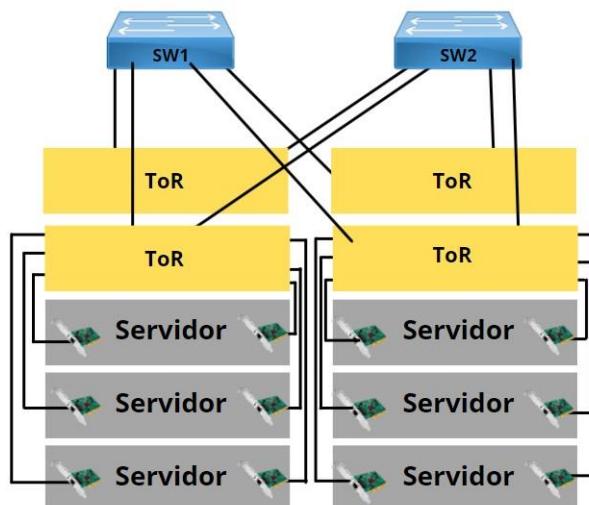
Temos a opção de não usar o switch virtual fornecido pelo Hypervisor, a Cisco por exemplo, trabalha com switches virtuais como o Nexus 1000v.

Redes de Data Center

Vamos ver como funciona uma rede de um data center físico, onde há racks cheio de servidores, switches... Os dois designers mais utilizados são:

TOR (Top of Rack)

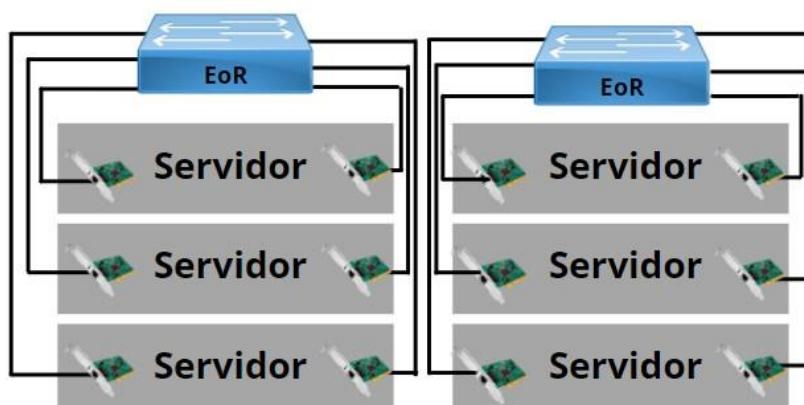
Top of rack ou topo do rack, nesse designer os switches são colocados em cima de cada rack. Os servidores ficam localizados abaixo dos switches ToR. Eles são conectados a mais se um switch ToR por questão de redundância.



Os switches ToR são ligados aos switches da camada de distribuição. Uma das vantagens dessa configuração é que a maior parte do cabeamento permanece dentro do Rack, os únicos cabos que saem do rack são os que saem dos switches ToR e se conectam aos switches da camada de distribuição. A desvantagem desse designer é que necessitamos de muitos switches, e dependendo da quantidade de servidores no rack, algumas portas desses switches podem ficar vazias.

EoR (End of Row)

No designer EoR ou ‘fim da linha’, os switches não ficam nos mesmos racks que servidores, nessa topologia os switches ficam em um rack separado, e todos os servidores se conectam a eles.



A principal vantagem desse designer é que ele não necessita de muitos switches, e a possibilidade de ficarem portas vazias e ociosas diminui muito. A desvantagem seria o cabeamento, pois a quantidade de cabos utilizados para conectar os racks dos servidores até o rack onde estão os switches EoR pode ser gigantesca.

Computação em nuvem

A ideia por trás da computação em nuvem é que um cliente seja capaz de solicitar um serviço e recebe-lo imediatamente, e para isso, não deve haver ninguém para verificar a solicitação, tudo deve ser automatizado.

Essa não é a única vantagem, O NIST (Instituto Nacional de padrões e Tecnologia) define que a computação em nuvem deve seguir as seguintes características:

- **On-demand self-service:** O consumidor pode provisionar por conta própria os recursos de computação, como tempo de uso do servidor e armazenamento em rede, tudo isso de forma automática e conforme a necessidade, sem necessitar de intervenção humana para tais solicitações.
- **Amplio acesso:** O serviço deve estar disponível em uma variedade de plataformas diferentes, incluindo computadores, tablets, smartphones. Também deve proporcionar acesso aos serviços através das mais diferentes conexões, incluindo Internet e conexões VPNs.
- **Pooling de recursos:** O provedor da nuvem não deve atribuir recursos fixos ao serviço, ele deve ser dinâmico. Por exemplo, se um site receber repentinamente um aumento expressivo de tráfego, o provedor deve ser capaz de criar vários servidores web automaticamente para lidar de forma satisfatória com esse tráfego.
- **Escalabilidade:** Para o cliente, os recursos devem parecer ilimitados. Por exemplo, existem alguns provedores de backup em nuvem. O cliente paga pelo serviço e, nos bastidores, o provedor cuidará para que haja espaço de suficiente de armazenamento.
- **Mensuração dos serviços:** Os sistemas na nuvem automaticamente controlam e otimizam o uso dos recursos através de medições de serviços como armazenamento, processamento, utilização da rede e contas de usuário. A utilização de recursos tem de ser monitorada, controlada e informada ao cliente, gerando transparência tanto para o fornecedor como para o consumidor do serviço utilizado.

Modelos de serviço

No modelo atual, **não se compra mais um produto, hoje, paga-se por um serviço**. Essa é a melhor definição para o que é Computação em Nuvem. Na Computação em Nuvem usamos muito o termo ‘como serviço’, abaixo os 03 modelos mais comuns:

- **(IaaS)** Infraestrutura como serviço
- **(PaaS)** Plataforma como serviço
- **(SaaS)** Software como serviço

IaaS (Infraestrutura como serviço)

A sigla **IaaS**, em inglês, significa: *infrastructure as a service*, em português: infraestrutura como serviço.

Este é o modelo onde os recursos de nuvem computacional são totalmente configuráveis. É possível dimensionar servidores, armazenamento, processamento e demais itens de acordo com a demanda.

Neste modelo existe autonomia total e flexibilidade para aumentar e reduzir recursos, realizar configurações de infraestrutura, configurações de firewall, gerenciamento da rede e diversas configurações. Isso quer dizer que para operar um ambiente IaaS é imprescindível ter conhecimento ou mesmo o acompanhamento de uma equipe de TI especializada para realizar a manutenção e gestão do ambiente.

Podemos fazer uma analogia com uma viagem para um camping. Utilizando o IaaS é como se o cliente escolhesse fazer toda a estrutura. Nesse modelo o cliente precisa providenciar a casa (barraca), itens de higiene pessoal, alimentação e qualquer outro item que necessitar para ter uma estadia confortável. Quanto maior o nível de autonomia (dentro do terreno que foi disponibilizado) maior a responsabilidade.

PaaS (plataforma como serviço)

A sigla PaaS, em inglês, significa: *platform as a service*, em português: infraestrutura como serviço.

Como o nome sugere, neste modelo são disponibilizadas plataformas para desenvolvimento e implantação de soluções para tecnologia em nuvem.

Desta forma, no PaaS, as aplicações já possuem uma finalidade de utilização e o cliente não tem necessidade de se preocupar com o que está na camada de infraestrutura, uma vez que o provedor da tecnologia em nuvem, fica responsável por essa parte. Ou seja, a solução PaaS permite ao cliente concentrar-se somente na aplicação que precisa disponibilizar.

SaaS (software como serviço)

A sigla SaaS em inglês, significa: *software as a service*, em português: software como serviço.

Semelhante ao PaaS, o SaaS também possui um propósito claro com foco em atender uma ou mais necessidades de quem irá utilizar o recurso.

Entretanto, o SaaS está mais próximo das regras de negócio e processos das empresas. O SaaS é a camada onde a maioria dos usuários estão em contato com a nuvem, seja alimentando, editando e visualizando informações.

Graças ao SaaS muitas empresas vêm mudando a forma como disponibilizam seus sistemas, conseguindo assim fornecer softwares com preço mais acessíveis para seus clientes.

Diferente do IaaS e PaaS a contratação de SaaS torna-se mais prática e barata para o usuário final, uma vez que demanda por menos conhecimento técnico e suporte de TI. Clientes SaaS não precisam se preocupar com a manutenção da estrutura do sistema, atualizações de versão e uma série de processos da área de tecnologia necessários para manter um software rodando dentro da estrutura física da empresa.

Nuvem Pública

A nuvem pública é o modelo mais utilizado nas empresas, por ser a mais adequada à utilização de softwares como serviço (SaaS) e permitir a ampliação da capacidade de armazenamento. Assim, os serviços são fornecidos em um ambiente virtualizado, acessível por meio da internet. Onde cada cliente tem seu nível de acesso aos recursos bem definidos.

É o tipo de nuvem mais barato, pois os custos de hardware, aplicações e largura de banda são cobertos pela provedora. A empresa paga somente pelos recursos utilizados.

Principais benefícios:

- Escalabilidade ilimitada;
- Disponibilidade;
- Recursos sob demanda;
- Custos controláveis e menores do que a infraestrutura interna (ou o modelo privado);
- Confiabilidade devido à quantidade de servidores disponíveis.

Uma nuvem pública é indicada para empresas que querem ganhar poder tecnológico sem dispor de grandes investimentos em TI. Ela também é útil para quem tem pressa em utilizar recursos virtualizados, por tratar-se de uma nuvem que já está pronta.

Nuvem Privada

A nuvem privada foi criada para atender às necessidades de um único negócio. Ela pode ser implementada internamente para atender diversas filiais, por exemplo, ou ser fornecida por um provedor. É uma arquitetura de data center própria e exclusiva de uma empresa. Ela oferece todos os benefícios da nuvem pública, como flexibilidade, escalabilidade, provisionamento, automação, monitoramento, entre outros, com a grande diferença de não ser dividida com outras empresas.

Vale lembrar que nesse modelo, os recursos “as-a-service” não são vendidos a diferentes clientes pelo provedor, mas ofertados a uma única empresa, podendo servir, por exemplo, diferentes filiais e parceiros de negócios.

Principais benefícios:

- Maior nível de confiabilidade;
- Controle totalmente interno dos servidores e outros recursos;
- Possibilidade de utilizar os recursos legados para manter a própria nuvem;

Uma nuvem privada é indicada para empresas que gerenciam dados muito sensíveis, como transações financeiras, por exemplo. Ela também serve bem para negócios nos quais a cultura de controle interno é bem rígida.

Nuvem híbrida

A nuvem híbrida mescla os dois modelos anteriores, visando extrair o melhor de ambos e desempenhar funções distintas dentro de uma mesma organização. Se por um lado as nuvens públicas oferecem mais escalabilidade do que as privadas, estas por sua vez são mais recomendadas para armazenamento de dados críticos. Logo, é possível maximizar as eficiências por meio dessa mescla, conforme as necessidades da empresa.

Principais benefícios:

- Flexibilidade e escalabilidade;
- Controle de custos;
- Controles técnicos (especialmente do modelo privado);
- Possibilidade de alternar entre o modelo público e o privado conforme a necessidade do negócio.

Na maioria das vezes, o modelo é escolhido por empresas que já possuem uma boa infraestrutura interna e também querem aproveitar os benefícios do modelo público, especialmente no que diz respeito a softwares como serviço (SaaS).

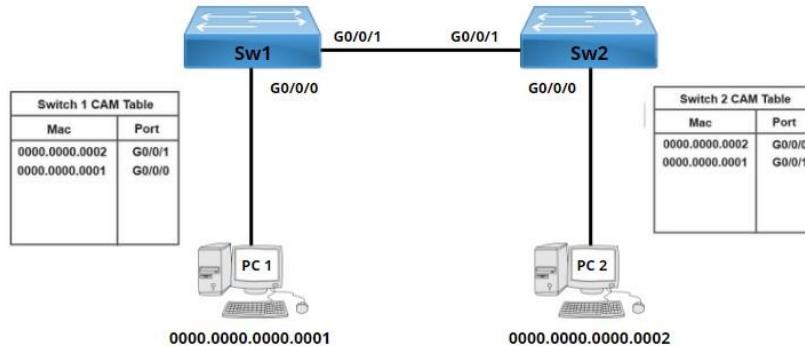
1.13 Describe switching concepts

Chegou a hora de entrarmos no conceito de switch, esse é um tema gostoso de estudar e fácil de ser replicado em emuladores como o packet tracer.

Switch é um dispositivo inteligente que opera na camada dois do modelo OSI. Ele toma decisões a partir do endereço **MAC (Media Access Control)** ou controle de acesso ao meio), o endereço MAC nada mais é que o endereço físico dos dispositivos (endereço da placa de rede de um computador, por exemplo).

Sempre que um endpoint é plugado na porta de um switch, ele verifica qual o endereço MAC daquele dispositivo e armazena em uma tabela chamada de **CAM Table**. Assim que o switch aprende o MAC Address de todos os dispositivos daquela rede, ele começa a encaminhar frames ethernet.

Cada porta do switch é um domínio de colisão, por isso ele é mais rápido (e seguro) que os hubs (dispositivo que opera na camada 1 do modelo OSI).

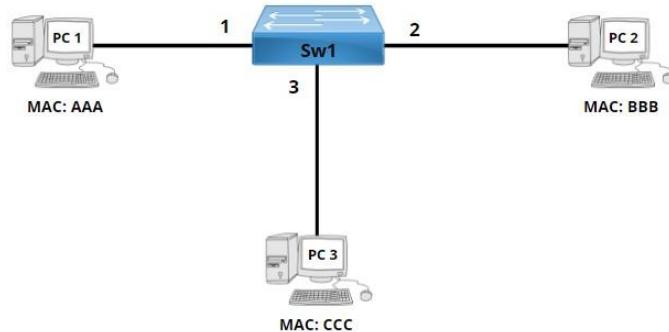


Exemplo de tabela MAC e como o switch armazena.

Nos próximos tópicos, vamos aprender como um switch utiliza essa tabela pra encaminhar frames.

1.13.a MAC learning and aging

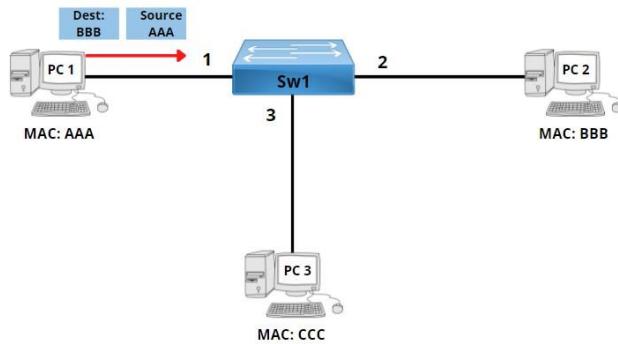
Observe a topologia abaixo:



Processo de aprendizagem de endereços MAC

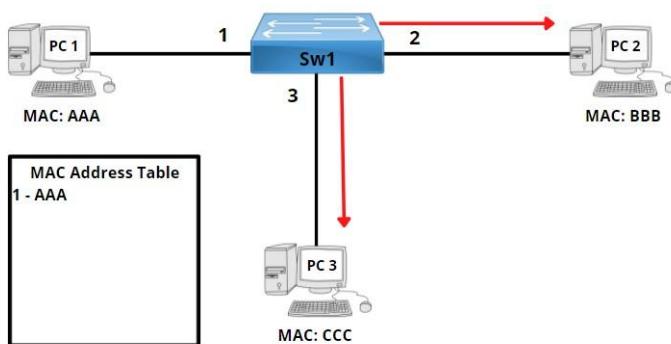
Na figura acima, temos um switch cercado por 03 computadores. Todos os computadores têm um endereço MAC (no exemplo acima, esses endereços MACs estão simplificados). Antes de enviar frames por essa rede, o switch precisa aprender o endereço MAC de cada um desses dispositivos.

Vamos enviar dados do PC 1 para o PC 2.



Processo de aprendizagem de endereços MAC

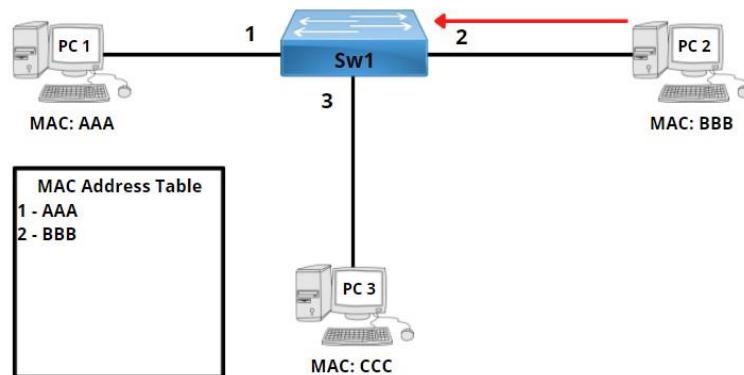
Source é a origem do quadro e destination é o destino. No exemplo acima, PC 1 deseja enviar dados para o PC 2, então, ele criará um quadro Ethernet com seu endereço MAC de origem (AAA) e destino (BBB). O switch possui uma tabela de endereçamento que está vazia no momento, porém, ela começará a ser populada assim que o quadro chegar:



Processo de aprendizagem de endereços MAC

O switch começou a construir a tabela com endereços MAC, porém, ele só **aprenderá os endereços MAC de origem**. Até agora, ele só aprendeu o endereço MAC do PC 1 e sabe que este computador está conectado na interface 1, essa informação é adicionada na tabela.

O switch não sabe onde o PC 2 está conectado, e para descobrir, ele só tem uma opção, essa opção é chamada de **flood (inundação)**. Nesse processo, ele enviará frames para todas as interfaces, exceto para interface 1, que é a interface de onde o frame veio.



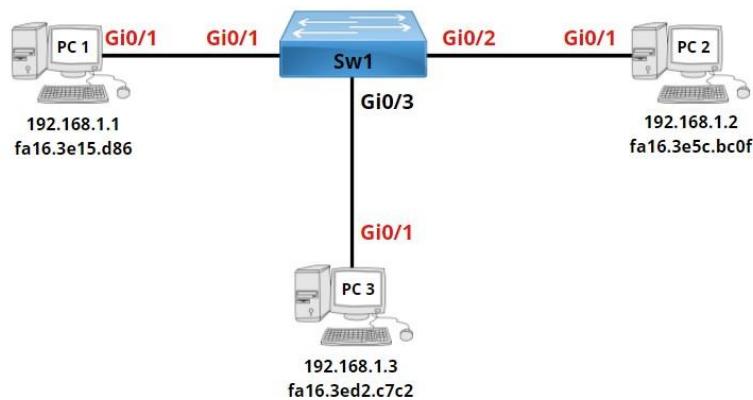
Processo de aprendizagem de endereços MAC

Quando o PC 2 ver seu endereço MAC como destino do quadro Ethernet, ele responderá o PC 1. Já o PC 3 quando receber o quadro irá ignorar, pois sabe que não é pra ele. Assim que a resposta do PC 2 chegar ao switch, ele colocará o endereço na sua tabela MAC e encaminhará o frame para o PC 1 através da interface 1.

Na próxima vez que o PC 1 encaminhar um quadro para o PC 2, o switch enviará direto para ele, sem a necessidade de ‘floodar’, e o PC 3, sequer saberá que essa mensagem foi enviada.

Comandos de verificação

Hora de vermos como esse processo funciona com dispositivos reais. A topologia que usaremos será basicamente a mesma, porém, agora adicionaremos endereços IP e endereços MAC reais.



Processo de aprendizagem de endereços MAC

Primeiro, vamos verificar se todas as interfaces estão conectadas e UP através do comando ‘show interface status’:

```
SW1#show interfaces status
```

Port	Name	Status	Vlan	Duplex	Speed	Type
Gi0/1		connected	1	auto	auto	unknown
Gi0/2		connected	1	auto	auto	unknown
Gi0/3		connected	1	auto	auto	unknown

Todas as interfaces estão conectadas, hora de verificarmos a tabela MAC do dispositivo:

```
SW1#show mac address-table dynamic
```

Mac Address Table

Vlan	Mac Address	Type	Ports
-----	-----	-----	-----
1	fa16.3e15.d86d	DYNAMIC	Gi0/1
1	fa16.3e5c.bc0f	DYNAMIC	Gi0/2
1	fa16.3ed2.c7c2	DYNAMIC	Gi0/3

Total Mac Addresses for this criterion: 3

O comando **show mac address-table dynamic** mostra todos os endereços que o switch aprendeu dinamicamente. Observe que ele aprendeu 03 endereços MAC.

Existe alguns parâmetros que podemos usar com esse comando, por exemplo:

```
SW1#show mac address-table dynamic address fa16.3e15.d86d
```

Mac Address Table

Vlan	Mac Address	Type	Ports
-----	-----	-----	-----
1	fa16.3e15.d86d	DYNAMIC	Gi0/1

Este comando mostra informações apenas do dispositivo que contém o endereço MAC que colocamos. Também é possível obter uma visão geral de todos os endereços MAC que foram aprendidos em uma determinada interface.

```
SW1#show mac address-table dynamic interface Gi0/2
```

Mac Address Table

Vlan	Mac Address	Type	Ports
-----	-----	-----	-----
1	fa16.3e5c.bc0f	DYNAMIC	Gi0/2

Total Mac Addresses for this criterion: 1

Aging

Switches trabalham com o conceito de aging-time (envelhecimento), observe o comando abaixo:

```
SW1#show mac address-table aging-time
```

```
Global Aging Time: 300
```

Vlan	Aging Time
-----	-----

Esse comando mostra o ‘**aging time**’ do switch, no caso 300 segundos. Isso quer dizer que, se o switch não vir um endereço MAC específico por 300 segundos, esse endereço MAC será removido da tabela de endereços MAC. O aging-time é necessário porque o dispositivo pode ter sido desligado ou desconectado do switch, e caso não houvesse esse mecanismo, ele ficaria para sempre poluindo a tabela MAC.

Caso queira remover manualmente algum endereço MAC da tabela, basta digitar o comando **clear mac address-table**:

```
SW1#clear mac address-table dynamic ?
```

address	address keyword
interface	interface keyword
vlan	vlan keyword

```
<cr>
```

Podemos escolher se queremos remover um único endereço MAC, todos os endereços MAC em uma interface específica, ou os que pertençam a uma determinada VLAN. Se não for adicionado nenhum parâmetro, todos os endereços MAC serão removidos.

```
SW1#clear mac address-table dynamic
```

No comando acima, limpamos toda a tabela MAC, observe que agora ela está vazia.

```
SW1#show mac address-table dynamic
```

```
Mac Address Table
```

Vlan	Mac Address	Type	Ports
-----	-----	-----	-----

Vamos enviar um ping do computador 1 para o 2 e verificar a tabela MAC sendo preenchida novamente:

```
PC1#ping 192.168.1.2
```

```
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
```

```
!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 9/9/10 ms
```

Ping bem sucedido, vamos verificar a tabela:

```
SW1#show mac address-table dynamic

      Mac Address Table

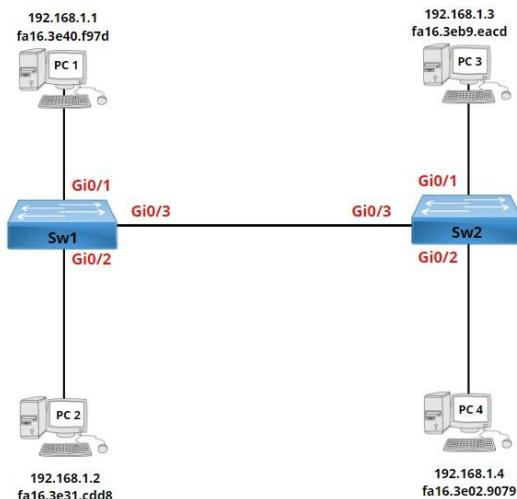
-----

Vlan      Mac Address          Type      Ports
-----  -----
  1        fa16.3e15.d86d    DYNAMIC   Gi0/1
  1        fa16.3e5c.bc0f    DYNAMIC   Gi0/2

Total Mac Addresses for this criterion: 2
```

A tabela foi populada novamente, sendo possível ver os endereços MAC do PC1 e PC2.

Hora de mudarmos um pouco a topologia para entendermos como funciona quando temos mais switches e hosts envolvidos



Processo de aprendizagem de endereços MAC

Vamos verificar a tabela MAC dos dois switches, preste atenção em especial a interface Gi0/3:

```
SW1#show mac address-table dynamic

      Mac Address Table

-----

Vlan      Mac Address          Type      Ports
-----  -----
  1        fa16.3e02.9079    DYNAMIC   Gi0/3
  1        fa16.3e31.cdd8    DYNAMIC   Gi0/2
```

```

1    fa16.3e40.f97d    DYNAMIC    Gi0/1
1    fa16.3eb9.eacd    DYNAMIC    Gi0/3

Total Mac Addresses for this criterion: 4

```

Primeiro, aplicamos o comando no SW1, observe que ele aprendeu os endereços MAC do PC3 e PC4 através da interface GigabitEthernet 0/3. Já o SW2 aprendeu os endereços do PC1 e 2 também através da interface Gi0/3:

```

SW2#show mac address-table dynamic

Mac Address Table

-----
Vlan   Mac Address      Type      Ports
---   -----
1     fa16.3e02.9079    DYNAMIC   Gi0/2
1     fa16.3e31.cdd8    DYNAMIC   Gi0/3
1     fa16.3e40.f97d    DYNAMIC   Gi0/3
1     fa16.3eb9.eacd    DYNAMIC   Gi0/1

Total Mac Addresses for this criterion: 4

```

Este exemplo nos mostra que um switch é capaz de aprender diversos endereços MAC pela mesma interface.

1.13.b Frame switching

Quando o switch recebe um frame, ele olha na tabela MAC para verificar se o endereço destino daquele frame já está mapeado na MAC Table. Quando ele já conhece o destino, simplesmente encaminha o frame para a porta que contém aquele MAC.

Existe 03 métodos de switching (ou comutação):

- **Store and forward** switching method
- **Cut through** switching method
- **Fragment free** switching method

O ‘método de comutação’ (switching), define como um switch processa um frame (quadro). O frame é uma parte do fluxo de dados que é transferido entre dois dispositivos na rede. Ele consiste em quatro partes: o endereço de hardware do dispositivo de origem, o endereço de hardware do dispositivo de destino, opções de controle e os dados.

Os endereços de hardware dos dispositivos de origem e de destino são usados respectivamente para identificar o dispositivo remetente e receptor do quadro. Como dito anteriormente, o endereço de hardware também é conhecido como endereço MAC. As opções de controle são usadas principalmente para dois propósitos: verificar a integridade do quadro e identificar o protocolo da camada superior que deve ser usado para processar o quadro no dispositivo de destino. Um quadro também é conhecido como frame Ethernet.

A imagem a seguir mostra a estrutura básica de um quadro Ethernet.

Preâmbulo	SFD	Destino	Origem	Tipo	Dados	FCS
7	1	6	6	2	46 - 1500	4

Frame Ethernet

Para processar um quadro, um switch usa o endereço MAC de origem, o endereço MAC de destino e a opção de controle (FCS) do quadro.

Vamos relembrar: Um switch possui várias portas. Essas portas são usadas para conectar vários dispositivos. Para saber qual dispositivo está conectado a qual porta, o switch armazena endereços MAC de todos os dispositivos conectados as suas portas em uma tabela conhecida como **tabela CAM**.

Uma entrada da tabela CAM consiste em um número de porta local e o endereço MAC do dispositivo que está conectado à porta. Se vários dispositivos estiverem conectados a uma única porta, os endereços MAC de todos os dispositivos conectados serão mapeados para a mesma porta.

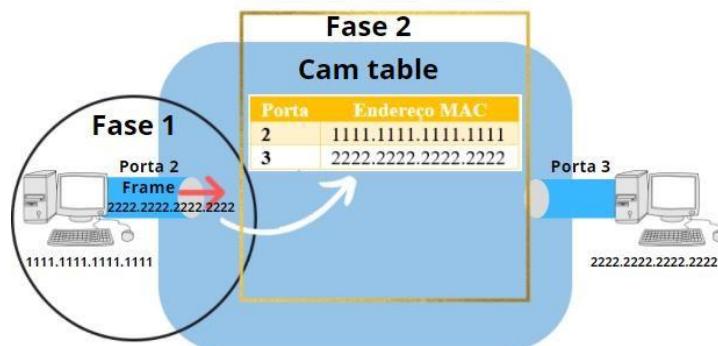
O processamento de quadro possui três fases: **1^a recebimento, 2^a processamento e 3^a encaminhamento** (receiving, processing, and forwarding).

Na fase de recebimento, o switch recebe o quadro ethernet em uma de suas portas.

Na fase de processamento, o switch lê o endereço MAC de destino do quadro e verifica na tabela CAM se há uma entrada para o endereço MAC de destino.

Se a tabela CAM tiver uma entrada para o endereço MAC de destino, o switch selecionara a porta que está associada ao endereço MAC de destino e começará a terceira fase. Caso o endereço MAC de destino não exista na tabela CAM, o switch selecionará todas as portas para a terceira fase, exceto a porta na qual o quadro foi recebido. Depois de selecionar a porta ou portas para a terceira fase, o switch validará a integridade do quadro.

A imagem a seguir mostra a primeira e a segunda fase do processamento interno de quadros.



As 3 fases do processamento de um frame

Um método de comutação se aplica a forma como um switch inicia a fase de encaminhamento e processamento do quadro ethernet. Existem três tipos de métodos de comutação: o método store-and-forward, o método fragment-free e o método cut-through.

Nos três métodos, a primeira e a segunda fases são iguais.

Na primeira fase, o switch recebe o quadro em uma de suas interfaces e começa a segunda fase antes mesmo de receber o quadro por inteiro, tão logo receba os bits do quadro que contém o endereço de destino. Esse mecanismo permite que o switch selecione a porta ou portas de encaminhamento mesmo antes do quadro entrar completamente no switch.

Depois de tomar a decisão de encaminhamento, o switch usa uma abordagem diferente para iniciar a terceira fase em cada um dos métodos. Vamos ver abaixo quais as diferenças entre os três métodos.

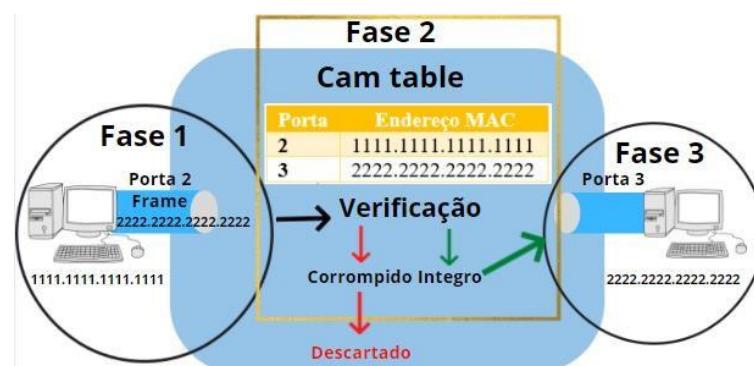
Método store-and-forward

Nesse método, o switch espera até que todos os bits do quadro sejam recebidos. Depois de receber todos os bits, o switch verifica se o quadro está íntegro (livre de erros). Se o quadro estiver íntegro, o switch encaminhará para a porta selecionada. Se o quadro recebido contiver erros, o switch descartará o quadro.

Para saber a condição de um quadro, o switch usa o campo FCS (sequência de verificação). O campo FCS contém um valor conhecido como CRC. O valor do CRC permite que um dispositivo receptor saiba se o quadro está exatamente no mesmo estado em que a fonte o empacotou, se foi danificado ou modificado durante a transmissão.

Depois de criar um quadro, o remetente ou o dispositivo de origem executa o algoritmo CRC (Cyclic Redundancy Check). O valor produzido por este algoritmo é conhecido como valor CRC, que é armazenado no campo FCS do quadro. Depois de armazenar o valor CRC, o dispositivo emissor envia o quadro pelo meio físico.

Ao receber o quadro, o receptor ou dispositivo de destino executa o algoritmo CRC e compara o resultado com o valor CRC armazenado no campo FCS. Se o resultado e o valor do CRC forem iguais, o quadro é considerado sem erros. Se não forem iguais, o quadro é considerado danificado e será descartado.



Método store-and-forward

Nesse método, o switch apenas encaminhará quadros que estiverem íntegros. Por isso, esse método fornece o mais alto nível de precisão, mas ao custo de velocidade. Se compararmos os três métodos de comutação, esse método fica respectivamente na primeira e na última posição em termos de precisão e velocidade.

Método cut-through

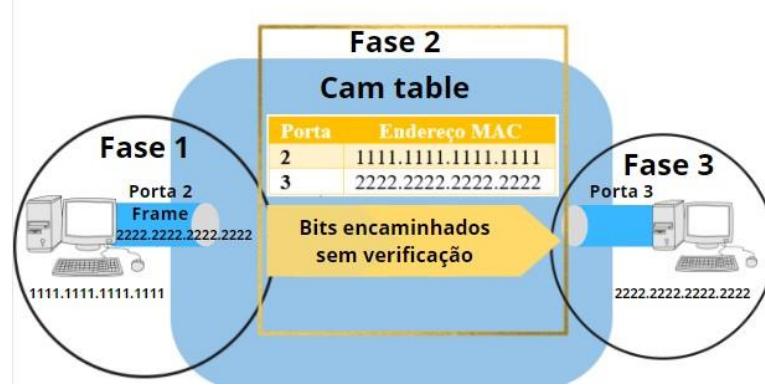
Nesse método, o switch inicia a terceira fase assim que a porta de encaminhamento é determinada. O quadro Ethernet armazena o endereço MAC de destino no terceiro campo. Para encaminhar o quadro, o switch precisa apenas do endereço MAC de destino. Como o endereço MAC de destino está localizado no início do quadro Ethernet, o switch pode começar a encaminhar os bits recebidos antes de receber todos os bits do quadro.

Bytes que o método cut-through precisa antes de tomar decisão para onde encaminhar o frame.



Método cut-through

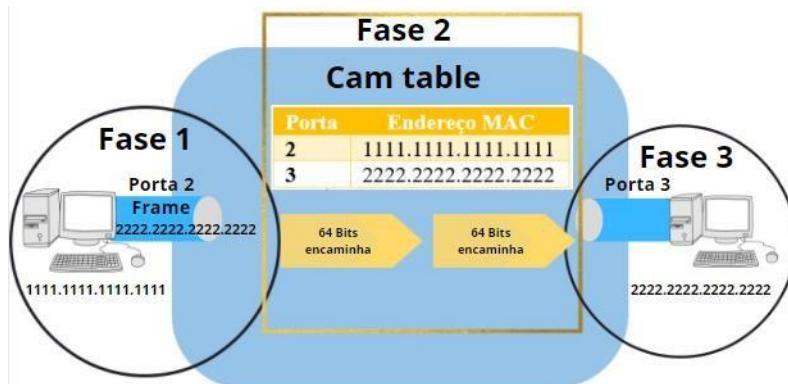
No cut-through, o switch não verifica a condição do quadro antes de encaminhá-lo. Isso reduz a latência, mas também propaga erros. De todos os três métodos de comutação, este é o método mais rápido. Mas fornece velocidade ao custo de encaminhar alguns quadros danificados.



Método cut-through

Método fragment-free

Nesse método, após determinar a porta de encaminhamento, o switch espera até que os primeiros 64 bytes do quadro sejam recebidos. Os 64 bytes são o tamanho mínimo de um quadro Ethernet. Um quadro Ethernet menor que 64 bytes é conhecido como quadro **runt**, e por definição, um quadro **runt** é um quadro corrompido.



Método fragment-free

O fragment-free é a versão modificada do método cut-through. Ele reduz o número de quadros runt que estão sendo enviados. Esse método também é conhecido como *cut-through modificado* ou método de *comutação sem execução* (runtless switching method).

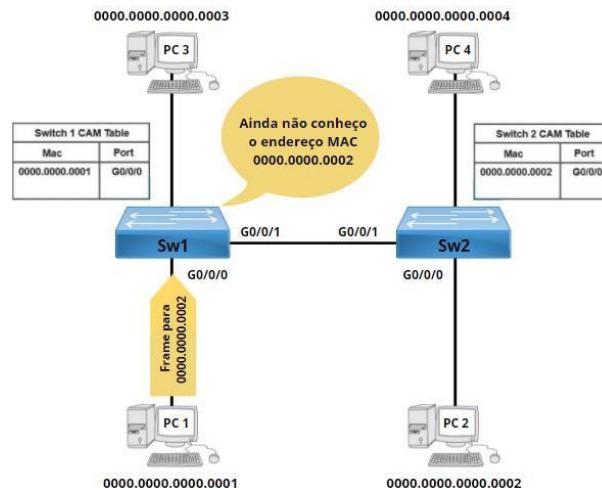
Tabela comparativa dos métodos de comutação:

Método	Store-and-Foward	Fragment-free	Cut-and-through
Posição de início do envio do quadro:	Após armazenar o quadro inteiro e executar o algoritmo CRC.	Após receber os primeiros 64 bytes.	Depois de receber os primeiros 8 bytes.
Posição em termos de velocidade:	Terceiro	Segundo	Primeiro
Posição em termos de precisão:	Primeiro	Primeiro	Terceiro

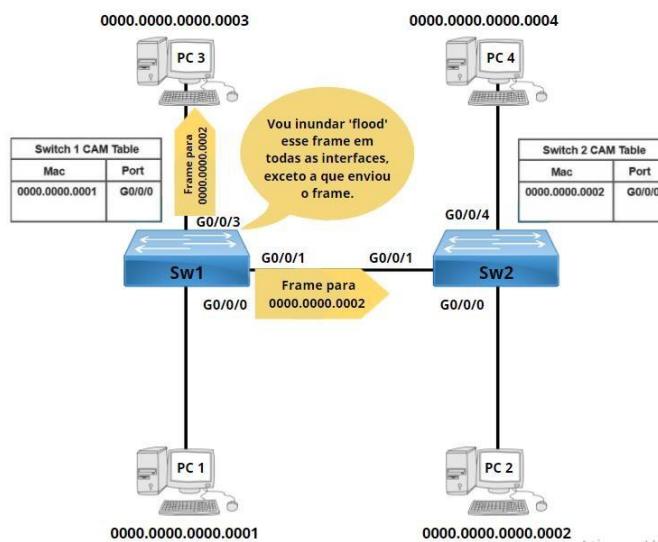
1.13.c Frame flooding

Já vimos um pouco desse processo anteriormente, mas agora trataremos mais profundamente.

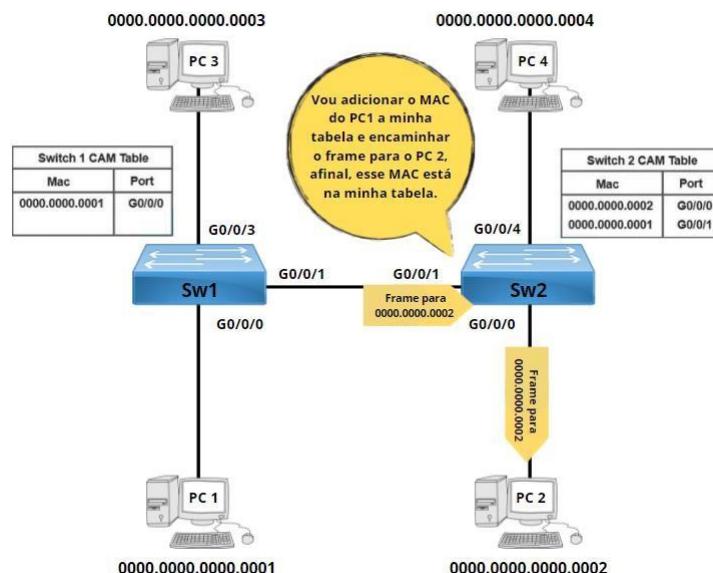
Se o endereço de destino do frame não estiver na tabela MAC do switch, este frame será inundado/encaminhado (flooded/forwarded) para todas as interfaces do switch, exceto a porta em que o frame foi recebido. Este processo é conhecido como ‘unicast flooding’, neste caso o endereço MAC de destino será um endereço broadcast ffff.ffff.ffff.



Processo de Frame flooding



Processo de Frame flooding



Processo de Frame flooding

1.13.d MAC address table

A tabela mac talvez seja o elemento mais importante do switch, pois é através dela que o switch toma decisões para onde encaminhar um quadro.

A tabela MAC é composta pela vlan que o dispositivo pertence, o endereço MAC desse dispositivo, se esse endereço foi aprendido dinamicamente ou de forma manual e em qual porta ele está. Podemos verificar a tabela MAC do switch com o comando **show mac address-table**:

Mac Address Table			

Vlan	Mac Address	Type	Ports
-----	-----	-----	-----
1	fa16.3e15.d86d	DYNAMIC	Gi0/1
1	fa16.3e5c.bc0f	DYNAMIC	Gi0/2
1	fa16.3ed2.c7c2	DYNAMIC	Gi0/3

Total Mac Addresses for this criterion: 3

Exercícios:

1. Qual dispositivo filtra o tráfego olhando para o endereço de destino do quadro e depois encaminha o quadro para a porta em que reside o sistema de destino?
 - A. Hub
 - B. Roteador
 - C. Repetidor
 - D. Switch

2. Um dispositivo que pode enviar e receber informações, mas não ao mesmo tempo, é chamado de:
 - A. Simplex
 - B. Full duplex
 - C. Multicast
 - D. Half-duplex

3. Uma mensagem enviada pela rede que é destinada a todos os dispositivos é conhecida como:
 - A. Unicast
 - B. Multicast
 - C. Full Duplex
 - D. Broadcast

4. Um grupo de hosts que podem receber mensagens de broadcast uns dos outros é conhecido como:
 - A. Domínio de colisão
 - B. Dominio de Active directory
 - C. Fully Qualified domain name
 - D. Broadcast Domain

5. Qual endereço abaixo é um endereço MAC de camada 02?
 - A. 192.168.0.2
 - B. PC 1
 - C. 00-AB-0F-2B-3C-4E
 - D. www.google.com

6. Qual padrão Gigabit Ethernet usa cabeamento UTP para alcançar 1000 Mbps?
 - A. 1000BaseTX
 - B. 1000BaseSX
 - C. 1000BaseCX
 - D. 1000BaseLX

7. Que tipo de cabo você é recomendado para conectar um host com conector RJ 45 a um switch?
 - A. Fibra
 - B. Cabo Cruzado
 - C. Cabo Direto
 - D. Cabo coaxial

8. Quais das opções a seguir são consideradas endereços de classe A? (Selecione todas as corretas)
 - A. 129.45.10.15
 - B. 10.35.87.5
 - C. 131.15.10.12
 - D. 192.156.8.34
 - E. 121.59.87.32
 - F. 210.45.10.112

9. Qual endereço abaixo é um endereço privado de classe A:
 - A. 24.56.10.12
 - B. 192.168.0.5
 - C. 172.16.45.10

- D. 10.55.67.99
10. Qual a máscara de rede padrão para o endereço 189.34.5.67?
- A. 255.0.0.0
 - B. 255.255.0.0
 - C. 255.255.255.0
 - D. 255.255.255.255
11. Qual equivalente em binário do número 137?
- A. 10001001
 - B. 10101001
 - C. 11001001
 - D. 10000101
12. Quais são as três fases do TCP three-way handshake?
- A. ACK/SYN, SYN, ACK
 - B. SYN, ACK/SYN, ACK
 - C. ACK/SYN, ACK, SYN
 - D. SYN, ACK, SYN/ACK
13. Qual protocolo da camada de transporte é responsável pela entrega não confiável?
- A. TCP
 - B. IP
 - C. ICMP
 - D. UDP
14. Quais dos seguintes campos são encontrados no cabeçalho IP? (Selecione todas as opções corretas.)
- A. Sequence number
 - B. Destination port
 - C. Source IP address
 - D. Type
 - E. Time to Live
 - F. SYN flag
15. Qual das alternativas a seguir é o equivalente IPv6 a 127.0.0.1?
- A. ::127
 - B. 127::1
 - C. ::1
 - D. FE80::
16. Se dividíssemos a rede 129.65.0.0 em seis redes diferentes, qual seria a nova máscara de sub-rede?
- A. 255.224.0.0
 - B. 255.255.192.0
 - C. 255.255.224.0
 - D. 255.255.255.224
17. Qual a máscara de sub-rede do endereço 135.44.33.22/20?
- A. 255.255.192.0
 - B. 255.255.240.0
 - C. 255.255.224.0
 - D. 255.255.248.0
18. Onde fica armazenada a running config?
- A. NVRAM
 - B. VRAM
 - C. ROM
 - D. Flash
19. Você deseja obter ajuda sobre quais parâmetros existem no comando ping. Qual comando poderia ser usado para mostrar uma lista de parâmetros?
- A. help ping
 - B. ping ?
 - C. ? ping
 - D. ?? ping

20. Qual dos prompts a seguir representa o modo de configuração global?
- A. router#
 - B. router(config)#
 - C. router>
 - D. router(global)#
21. Que comando você usaria para navegar para o modo de configuração global?
- A. >config terminal
 - B. #setup
 - C. >enable
 - D. #config terminal
22. Qual padrão wireless opera em ambas as frequências 2.4Ghz e 5 Ghz?
- A. 802.11a
 - B. 802.11b
 - C. 802.11n
 - D. 802.11i
23. Quais padrões Wireless oferecem a maior velocidade de transmissão?
- A. 802.11a
 - B. 802.11b
 - C. 802.11n
 - D. 802.11i
24. Qual item abaixo não é usado em uma conexão ad hoc?
- A. WEP
 - B. Wireless Cliente
 - C. Access Point
 - D. SSID

1 – d, 2 – d, 3 – d, 4 – d, 5 – c, 6 – a, 7 – c, 8 – b, e, 9 – d, 10 – B, 11- a, 12 – b, 13 – d, 14 c, e, 15 – c, 16 – c, 17 – b, 18 – b
19 – b, 20 – b, 21 – d, 22 – c, 23 – a, c, 24 – c

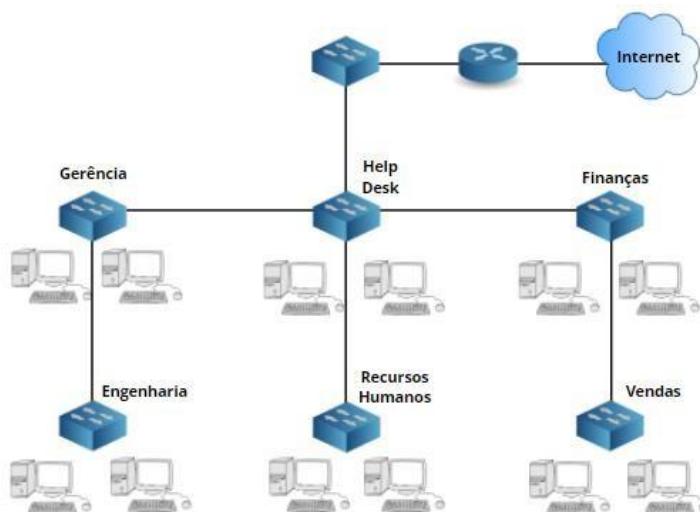
2.0 Network Access

Estamos entrando, finalmente, na parte mais prática. Aqui veremos como os dispositivos finais se conectam a rede e como eles obtém acesso a ela através dos dispositivos de camada 2, como o switch por exemplo.

2.1 Configure and verify VLANs (normal range) spanning multiple switches

Vamos aprender o que é uma VLAN (Lan Virtual), para que serve e como é configurada.

Observe a imagem abaixo:



Temos a topologia de uma empresa com vários departamentos, repare os usuários de cada setor estão agrupados fisicamente e conectados ao mesmo switch. Vamos refletir um pouco sobre isso. Esse designer faz sentido? Parece um bom design de rede? Vamos aprofundar um pouco mais, realizando mais perguntas para acharmos boas respostas!

- O que acontece quando um computador conectado ao switch do setor de ‘Vendas’ envia uma solicitação ARP?
- O que acontece quando o switch que está conectado ao pessoal do Helpdesk cai?
- Os usuários do setor de Recursos Humanos obterão conectividade com boa velocidade?
- Como podemos implementar segurança nesta rede?

Vamos as respostas: Esse é um péssimo designer de rede. Se qualquer computador nessa rede enviar um ‘broadcast’ os switches inundarão (flood) toda a rede com esse pacote! O ‘flood’ em toda rede acontecerá também quando um switch necessitar aprender sobre um endereço MAC que ele ainda não conhece. Isso causará lentidão.

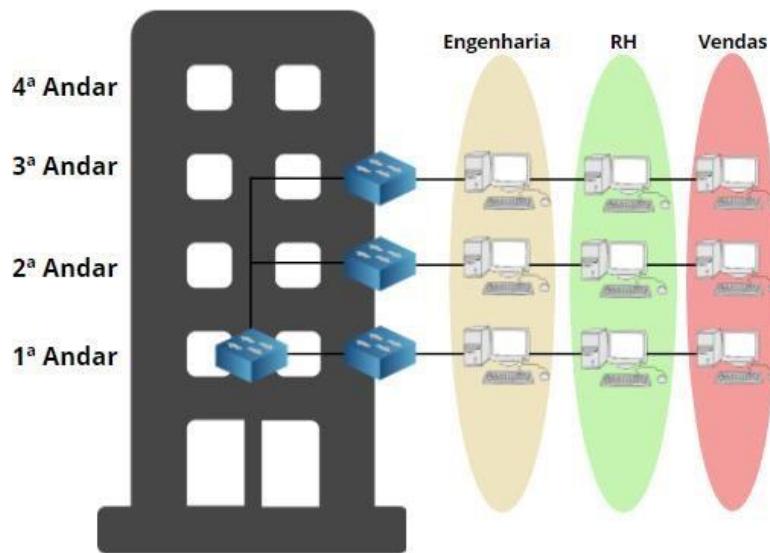
Se o switch do helpdesk falhar, os usuários do setor de Recursos Humanos ficarão “isolados”, e não conseguirão acessar outros departamentos ou a Internet. Na verdade, todos os usuários que precisam navegar na Internet precisam passar pelo switch do setor de Helpdesk, o que causa gargalos na rede e um único ponto de falha.

Por último, mas não menos importante, e quanto à segurança? Poderíamos implementar port-security com bloqueios baseados nos endereços MAC, mas esse não é um método muito seguro, pois os endereços MAC são fáceis de falsificar. Uma solução para todos esses problemas é a criação de VLANs.

Uma pergunta importante para assegurar que podemos ir em frente: Você sabe quantos domínios de broadcast temos aqui?

Se um computador da Central de Vendas enviar um quadro de broadcast, sabemos que todas os switches o encaminharão. Olhe no topo da topologia, lá nós temos um roteador, você acha que o roteador encaminhará o quadro de broadcast?

A resposta é: Roteadores não encaminham broadcast, portanto, efetivamente eles “limitam” o domínio de broadcast. Perceba que na outra ponta do roteador temos uma conexão com a Internet, esta conexão seria outro domínio de broadcast, logo, temos 2 domínios de broadcast nessa topologia. Vamos aprofundar um pouco mais:



Quando estamos lidando com switches devemos ter em mente que há uma grande diferença entre a topologia física e a topologia lógica. A topologia física refere-se a forma como os cabos são conectados, enquanto a topologia lógica refere-se a como configuramos as coisas “virtualmente”. No exemplo acima, temos 4 switches e 3 VLANs chamadas RH, Engenharia e Vendas. Uma VLAN é uma LAN virtual, então é como ter um “switch dentro de um switch”.

As principais vantagens de usar VLANs são:

- Uma VLAN é um domínio de broadcast, o que significa que se um usuário na VLAN de ‘RH’ enviar um frame de broadcast, apenas os usuários da mesma VLAN irão recebê-lo.
- Os usuários só podem se comunicar na mesma VLAN, a menos, que tenhamos um roteador.
- Os usuários não precisam estar agrupados fisicamente, observe, temos usuários na VLAN de Engenharia no 1^a, 2^a e 3^a andar.

Já aprendemos o que são Vlans, agora é hora de aprendermos a configura-las.

Configuração de Vlans em Switches Cisco

Hora de aprendermos a configurar VLANs em switches Cisco da linha Catalyst. Vamos aprender também como atribuir interfaces a essas VLANs.

Vamos começar com uma topologia de rede simples:



O comando para vermos as vlans configuradas em um switch é: **Show vlan**

```
SW1#show vlan
```

VLAN Name	Status	Ports
-	-	-

```

1    default           active   Fa0/1, Fa0/2, Fa0/3, Fa0/4
                           Fa0/5, Fa0/6, Fa0/7, Fa0/8
                           Fa0/9, Fa0/10, Fa0/12
                           Fa0/13, Fa0/14, Fa0/22
                           Fa0/23, Fa0/24, Gi0/1, Gi0/2

1002 fddi-default      act/unsup
1003 token-ring-default act/unsup
1004 fddinet-default   act/unsup
1005 trnet-default     act/unsup

```

Observe na saída do comando que a VLAN 1 é a VLan default do switch, e todas as interfaces ativas estão atribuídas à ela.

As informações das VLANs não são salvas na *running-config* ou na *startup-config*, mas em um arquivo separado chamado **vlan.dat**, esse arquivo fica armazenado na memória flash do dispositivo. Para excluir as informações de VLANs, deve-se excluir este arquivo, digitando **delete flash: vlan.dat**.

Os computadores estão na mesma subrede 192.168.1.0/24 com finais 1 para o PC1 e 2 para o PC2. Vamos ver se há conectividade entre eles enviando um ping:

```

C:\Documents and Settings\PC1>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

Reply from 192.168.1.2: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms

```

Mesmo com a configurações padrão do switch, o PC1 é capaz de pingar o PC2. Vamos criar uma nova VLAN e incluir o PC1 e PC2:

```

SW1(config)#vlan 50
SW1(config-vlan)#name Computadores
SW1(config-vlan)#exit

```

Acabamos de criar uma vlan de número 50, demos também o nome de ‘Computadores’, o nome é opcional e serve apenas para facilitar o entendimento para nós, humanos. O que conta mesmo para o switch é o número! Vamos ver como estão as vlans:

SW1#show vlan			
VLAN Name	Status	Ports	
<hr/>			
1 default	active	Fa0/1, Fa0/2, Fa0/3, Fa0/4 Fa0/5, Fa0/6, Fa0/7, Fa0/8 Fa0/9, Fa0/10, Fa0/11, Fa0/12 Fa0/13, Fa0/14, Fa0/15, Fa0/23, Fa0/24, Gi0/1, Gi0/2	
50 Computers	active		

A VLAN 50 foi criada no SW1, observe que ela está ativa. No entanto, nenhuma porta está atribuída a VLAN 50. Vamos atribuir as interfaces que os computadores estão plugados a ela:

```
SW1(config)interface fa0/1
SW1(config-if)#switchport mode access
SW1(config-if)#switchport access vlan 50
SW1(config)interface fa0/2
SW1(config-if)#switchport mode access
SW1(config-if)#switchport access vlan 50
```

Primeiro, configuramos o **modo de acesso (access mode)** com o comando ‘switchport mode access’. Depois, colocamos a interface na vlan escolhida com o comando ‘switchport access vlan XX’.

Vamos verificar como estão as vlans nesse momento:

SW1#show vlan			
VLAN Name	Status	Ports	
<hr/>			
1 default	active	Fa0/3, Fa0/4 Fa0/5, Fa0/6, Fa0/7, Fa0/8 Fa0/9, Fa0/10, Fa0/12 Fa0/13, Fa0/14, Fa0/15, Fa0/23, Fa0/24, Gi0/2	
50 Computers	active	Fa0/1, Fa0/2	

Ambos os computadores estão na vlan 50. Vamos testar a conectividade através do ping do PC1 para o PC2 e ver se tudo está funcionando como deveria:

```
C:\Documents and Settings\PC1>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:
Reply from 192.168.1.2: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Conforme o teste acima, há conectividade entre os computadores. Hora de aprendemos outro comando para verificação das interfaces em um switch Cisco:

```
SW1#show interfaces fa0/1 switchport

Name: Fa0/1

Switchport: Enabled

Administrative Mode: static access

Operational Mode: static access

Administrative Trunking Encapsulation: negotiate

Operational Trunking Encapsulation: native

Negotiation of Trunking: Off

Access Mode VLAN: 50 (Computers)

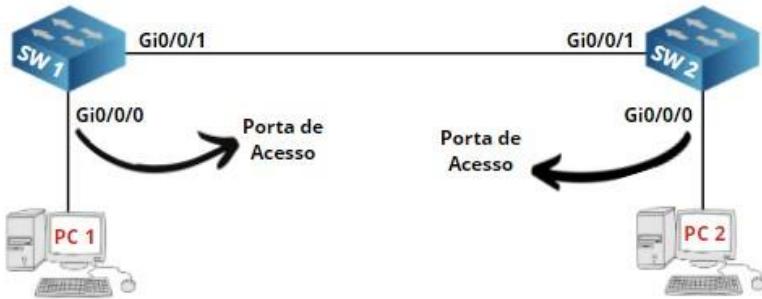
Trunking Native Mode VLAN: 1 (default)
```

Através do comando “show interfaces switchport” é possível ver que o **modo operacional**, no caso em tela o modo é “acesso estático (static access)”, o que significa que a interface está no *modo de acesso*. Também nos é mostrado que a interface está atribuída à VLAN 50.

2.1.a Access ports (data and voice)

Vlan de dados

Portas de acesso também são conhecidas como ‘portas de fronteira’, são as portas que conectam os switches aos dispositivos finais como computador, telefone, etc.



Portas de acesso não contém tags de vlans (falaremos sobre essas tags (etiquetas) mais a frente), portando os dispositivos conectados a essas portas encaminham frames sem essas etiquetas (untagged), na verdade, o dispositivo não sabe se ele está ou não em determinada vlan, para os dispositivos esse processo é transparente.

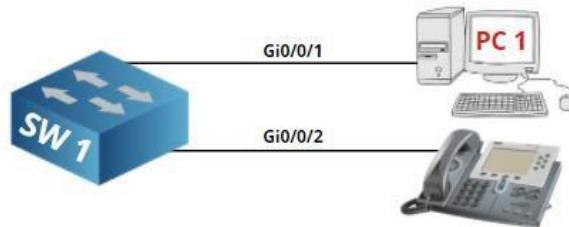
Para colocarmos uma interface em modo de acesso (access mode) usamos o comando ‘**switchport mode access**’, dentro da interface que queremos colocar nesse estado:

```
SW1(config)#interface fa0/1
SW1(config-if)#switchport mode access
```

Vlan de Voz

Normalmente os telefones IPs ficam próximos a um computador. Eles trabalham com os mesmos cabos UTP que os computadores e também usam a suíte de protocolos Ethernet. Se quisermos conectá-los a um switch, temos duas opções.

Podemos conectar o computador e o telefone IP usando dois cabos diferentes:



Essa conexão funcionará, porém, tem algumas desvantagens:

- Precisaremos instalar um novo cabo da porta do switch até o telefone IP.
- Perderemos uma porta do switch para conectar somente o telefone IP.

Por essas e outras questões que a maioria dos telefones IPs (incluindo Cisco), vem com um mini switch de três portas embutido:

- Uma porta se conecta ao switch.
- Uma porta se conecta ao computador.
- Uma porta (interna) se conecta ao telefone.

Isso permite conectar o telefone IP e o computador desta forma:



VLAN de voz também é conhecida como VLAN Auxiliar (AUX VLAN)

O computador ficará em uma **VLAN de dados** (data vlan) e o telefone fica em uma **VLAN de voz** (voice vlan), conforme a topologia abaixo:



Nos bastidores, há um trunk entre o switch e o telefone IP. A porta do telefone IP que se conecta ao computador é uma porta de acesso. O telefone IP encaminhará todo o tráfego do computador para o switch com frames **não marcados (untagged)**, já o tráfego do próprio telefone IP será **marcado (tagged)**. As únicas duas VLANs permitidas, porém, são as VLANs de dados e de voz.

Configuração

Vamos configurar a porta do switch da seguinte forma: VLAN 100 para o computador e VLAN 101 para o telefone IP. Primeiro passo é criar as duas VLANs:

```
SW1(config)#vlan 100
SW1(config-vlan)#name COMPUTADOR
SW1(config-vlan)#exit
SW1(config)#vlan 101
SW1(config-vlan)#name VOIP
SW1(config-vlan)#exit
```

Agora, podemos configurar a interface:

```
SW1(config)#interface GigabitEthernet 0/1
SW1(config-if)#switchport mode access
SW1(config-if)#switchport access vlan 100
SW1(config-if)#switchport voice vlan 101
SW1(config-if)#exit
```

Observe que configuramos a interface em modo de acesso (access mode) e usamos a vlan 100 para o computador e vlan 101 para o VoIP. O comando ‘**switchport voice vlan 101**’ informa ao switch para usar a vlan 101 como vlan de voz.

Configuramos o switch, mas como o telefone IP sabe quais VLANs usar? Os telefones IPs da Cisco usam o CDP (Cisco Discovery Protocol) para realizar essa descoberta. O Telefone IP aprenderá através do CDP quais VLANs ele deve usar. Telefones IPs de outros fabricantes usam LLDP (Link Layer Discovery Protocol) para essa função. Ainda veremos quais as funções dos protocolos CDP e LLDP.

Verificação

Vamos utilizar o comando ‘**show interfaces**’ para verificarmos o que fizemos até aqui:

```
SW1#show interfaces GigabitEthernet 0/1 switchport
```

```

Name: Gi0/1
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
Access Mode VLAN: 100 (COMPUTADOR)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: 101 (VOIP)
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk Native VLAN tagging: enabled
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk associations: none
Administrative private-vlan trunk mappings: none
Operational private-vlan: none
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
Capture VLANs Allowed: ALL

Protected: false
Unknown unicast blocked: disabled
Unknown multicast blocked: disabled
Appliance trust: none

```

Observe que estamos usando VLAN 100 para os computadores e VLAN 101 para os telefones IP.

Também podemos dar uma olhada no status da interface trunk (interface que conecta um switch ao outro), embora, a interface não esteja em modo trunk ela nos dirá quais são as duas VLANs usadas (essa interface na verdade é um trunk, lembre-se que os telefones VoIP possuem um pequeno switch de três portas).

```
SW1#show interfaces GigabitEthernet 0/1 trunk
```

Port	Mode	Encapsulation	Status	Native vlan
Gi0/1	off	negotiate	not-trunking	1
Port Vlans allowed on trunk				
Gi0/1	100-101			
Port Vlans allowed and active in management domain				
Gi0/1	100-101			
Port Vlans in spanning tree forwarding state and not pruned				
Gi0/1	100-101			

2.1.b Default VLAN

Por default, todas as interfaces do switch estão na vlan 1, logo, está é a vlan default dos switches Cisco.

Ela não pode ser editada ou excluída, a grosso modo, é assim e pronto. As melhores práticas recomendam que não deixemos nenhuma interface nessa vlan.

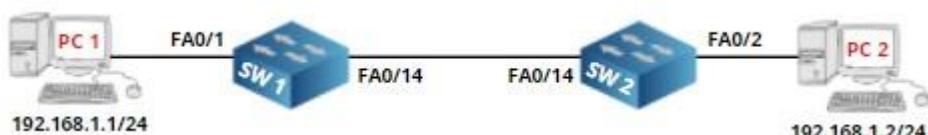
2.2 Configure and verify interswitch connectivity

Nesse tópico abordaremos a conectividade entre switches, incluindo a interface trunk que falamos tanto, mas ainda não explicamos.

2.2.a Trunk ports

Aprendemos que interfaces de acesso são as interfaces utilizadas para conectar os dispositivos finais aos switches. Mas, e se quisermos conectar um switch ao outro, qual interface devemos utilizar? Para casos como esse, utilizamos as interfaces trunk. Vamos aprender o que são e como configura-las nos switches da Cisco.

Observe a topologia abaixo:



Temos dois computadores conectados a dois switches. Colocaremos os computadores na mesma VLAN e criaremos uma interface trunk entre os dois switches. Vamos começar criando as vlans:

```
SW1(config)#vlan 50
SW1(config-vlan)#name Computadores
SW1(config-vlan)#exit
```

```
SW2(config)#vlan 50
SW2(config-vlan)#name Computadores
SW2(config-vlan)#exit
```

Vamos colocar as interfaces conectadas aos computadores na interface correta:

```
SW1(config)#interface fa0/1  
SW1(config-if)#switchport access vlan 50
```

```
SW2(config)#interface fa0/2  
SW2(config-if)#switchport access vlan 50
```

A próxima etapa é criar uma interface trunk entre os dois switches. Tecnicamente, nesse momento não haveria problema de deixar as interfaces entre os dois switches em modo de acesso porque só temos uma única VLAN.

```
SW1(config)#interface fa0/14  
SW1(config-if)#switchport mode trunk  
Command rejected: An interface whose trunk encapsulation is "Auto" can not be  
configured to "trunk" mode.
```

```
SW2(config)#interface fa0/14  
SW2(config-if)#switchport mode trunk  
Command rejected: An interface whose trunk encapsulation is "Auto" can not be  
configured to "trunk" mode.
```

Tivemos um erro na hora de mover as interfaces para o “modo trunk”. O comando para realizar essa troca está correto ‘**switchport mode trunk**’, porém, dependendo do modelo do switch, esse erro aparecerá. Quando isso acontece, é porque antes de tudo, precisamos mudar o tipo de encapsulamento. Vamos ver quais opções temos:

```
SW1(config-if)#switchport trunk encapsulation ?  
dot1q      Interface uses only 802.1q trunking encapsulation when trunking  
isl        Interface uses only ISL trunking encapsulation when trunking  
negotiate  Device will negotiate trunking encapsulation with peer on interface
```

Aqui podemos escolher entre o encapsulamento 802.1Q ou ISL, aprenderemos no próximo tópico o que é o protocolo 802.1q, por enquanto, vamos apenas escolhe-lo através do comando ‘**switchport trunk encapsulation**’:

```
SW1(config-if)#switchport trunk encapsulation dot1q
```

```
SW2(config-if)#switchport trunk encapsulation dot1q
```

Comando aplicado, hora de verificarmos:

```
SW1#show interfaces fa0/14 switchport  
Name: Fa0/14  
Switchport: Enabled
```

```
Administrative Mode: dynamic auto  
Operational Mode: static access  
Administrative Trunking Encapsulation: dot1q
```

```
SW2#show interfaces fa0/14 switchport  
Name: Fa0/14  
Switchport: Enabled  
Administrative Mode: dynamic auto  
Operational Mode: static access  
Administrative Trunking Encapsulation: dot1q
```

O encapsulamento agora é o 802.1Q, hora de mudarmos as interfaces para o modo trunk:

```
SW1 (config) # interface fa0/14  
SW1 (config-if) # switchport mode trunk
```

```
SW2 (config) # interface fa0/14  
SW2 (config-if) # switchport mode trunk
```

E verificarmos:

```
SW1#show interfaces fa0/14 switchport  
Name: Fa0/14  
Switchport: Enabled Administrative Mode: trunk Operational Mode: trunk  
Administrative Trunking Encapsulation: dot1q  
Operational Trunking Encapsulation: dot1q
```

```
SW2#show interfaces fa0/14 switchport  
Name: Fa0/14  
Switchport: Enabled Administrative Mode: trunk Operational Mode: trunk  
Administrative Trunking Encapsulation: dot1q  
Operational Trunking Encapsulation: dot1q
```

Como o modo operacional é o dot1q, podemos afirmar que temos um trunk formado. Vamos verificar se o PC1 consegue pingar o PC2:

```
C: \ Documents and Settings \ PC1> ping 192.168.1.2  
Ping em 192.168.1.2 com 32 bytes de dados:
```

```

Resposta de 192.168.1.2: bytes = 32 tempo <1ms TTL = 128
Resposta de 192.168.1.2: bytes = 32 tempo <1ms TTL = 128
Resposta de 192.168.1.2: bytes = 32 tempo <1ms TTL = 128
Resposta de 192.168.1.2: bytes = 32 tempo <1ms TTL = 128
Estatísticas de ping para 192.168.1.2:

Pacotes: Enviados = 4, Recebidos = 4, Perdidos = 0 (0% de perda),
Tempos aproximados de ida e volta em milissegundos:

Mínimo = 0ms, Máximo = 0ms, Média = 0ms

```

Vamos analisar alguns comandos e descobrir qual o comportamento quando há uma interface trunk no meio. O primeiro comando é o ‘**show vlan**’

```

SW2#show vlan

VLAN Name                               Status      Ports
-----+
1      default                           active     Fa0/1, Fa0/3, Fa0/4, Fa0/5
                                         Fa0/6, Fa0/7, Fa0/8, Fa0/9
                                         Fa0/10, Fa0/11, Fa0/12, Fa0/13
                                         Fa0/15, Fa0/22, Fa0/23, Fa0/24
                                         Gi0/1, Gi0/2
50    Computadores                      active     Fa0/2

```

Repare que não estamos vendo a interface Fa0/14. Isso é normal, o comando **show vlan** **mostra apenas interfaces no modo de acesso**, por isso as interfaces em modo trunk não são mostradas.

```

SW2#show interface fa0/14 trunk

Port        Mode         Encapsulation  Status        Native vlan
Fa0/14      on          802.1q        trunking     1
Port        Vlans allowed on trunk
Fa0/14      1-4094
Port        Vlans allowed and active in management domain
Fa0/14      1,50
Port        Vlans in spanning tree forwarding state and not pruned
Fa0/14      50

```

O comando **show interface trunk** é muito útil nesses casos. Com ele é possível ver se uma interface está no modo trunk (trunk mode), qual protocolo de encapsulamento está sendo usado (802.1Q ou ISL) e qual a VLAN nativa. Também podemos ver que VLAN 1 - 4094 estão permitidas nessa interface trunk.

Ele também nos informa quais as vlans estão ativas, nesse momento apenas a VLAN 1 (VLAN nativa) e a VLAN 50 estão ativas. Por último, mas não menos importante, ele também mostra quais VLANs estão no estado de encaminhamento (forwarding state) do spanning tree (assunto que veremos mais adiante).

Vamos ver mais um detalhe sobre interfaces de acesso e trunk:

```
SW2#show interface fa0/2 switchport  
  
Name: Fa0/2  
  
Switchport: Enabled  
  
Administrative Mode: static access  
  
Operational Mode: static access
```

Uma interface pode estar no modo de acesso ou no modo de trunk. A interface acima está conectada ao PC2 e por isso o modo operacional é “acesso estático (static access)”, o que significa que está em modo de acesso (access mode).

```
SW2#show interfaces fa0/14 switchport  
  
Name: Fa0/14  
  
Switchport: Enabled  
  
Administrative Mode: trunk  
  
Operational Mode: trunk
```

Esta é a interface do Sw2 que está conectada ao Sw1, o modo operacional esta setado para trunk mode.

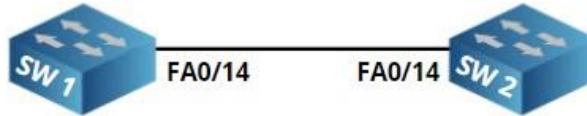
Se entrarmos na configuração da interface é possível verificar que temos mais opções que simplesmente o access mode ou trunk mode.

```
SW2(config-if)#switchport mode ?  
  
access      Set trunking mode to ACCESS unconditionally  
  
dot1q-tunnel  set trunking mode to TUNNEL unconditionally  
  
dynamic      Set trunking mode to dynamically negotiate access or trunk  
  
private-vlan  Set private-vlan mode  
  
trunk        Set trunking mode to TRUNK unconditionally
```

Além da access mode e trunk mode temos mais três métodos. Mas vamos focar, por ora, no método **dinâmico**.

```
SW2(config-if)#switchport mode dynamic ?  
  
auto        Set trunking mode dynamic negotiation parameter to AUTO  
  
desirable   Set trunking mode dynamic negotiation parameter to DESIRABLE
```

Podemos escolher entre **dynamic auto** e **dynamic desirable**. Habilitando essas opções, o switch descobrirá automaticamente se a interface deve se tornar uma porta de acesso ou trunk. Então, qual é a diferença entre automático **dynamic auto** e **dynamic desirable**? Vamos descobrir!



Vamos fazer algumas mudanças no SW1 e SW2 para ver qual será o resultado:

Primeiro, vamos mudar ambas as interfaces para ‘dynamic auto’:

```
SW1 (config)#interface fa0/14
SW1 (config-if)#switchport mode dynamic auto
```

```
SW2 (config)#interface fa0/14
SW2 (config-if)#switchport mode dynamic auto
```

Observe o resultado da aplicação desse comando:

```
SW1(config-if)#do show interface f0/14 switchport
Name: Fa0/14
Switchport: Enabled
Administrative Mode: dynamic auto
Operational Mode: static access
```

```
SW2(config-if)#do show interface f0/14 switchport
Name: Fa0/14
Switchport: Enabled
Administrative Mode: dynamic auto
Operational Mode: static access
```

O modo administrativo (administrative mode) passou a ser dynamic auto e, como resultado, agora temos uma access port.

E se mudarmos para dynamic desirable? Vamos testar o que acontecerá:

```
SW1(config)#interface fa0/14
SW1(config-if)#switchport mode dynamic desirable
```

```
SW2(config)#interface fa0/14
SW2(config-if)#switchport mode dynamic desirable
```

```
SW1#show interfaces fa0/14 switchport
```

```
Name: Fa0/14  
Switchport: Enabled  
Administrative Mode: dynamic desirable  
Operational Mode: trunk
```

```
SW2#show interfaces fa0/14 switchport  
Name: Fa0/14  
Switchport: Enabled  
Administrative Mode: dynamic desirable  
Operational Mode: trunk
```

Uma vez que mudamos ambas as interfaces para o dynamic desirable, terminamos com uma interface trunk.

O que você acha que acontecerá se misturarmos os tipos de switchport? Dynamic desirable de um lado, e dynamic auto do outro? Vamos descobrir!

```
SW1(config)#interface fa0/14  
SW1(config-if)#switchport mode dynamic desirable
```

```
SW2(config)#interface fa0/14  
SW2(config-if)#switchport mode dynamic auto
```

```
SW1#show interfaces f0/14 switchport  
Name: Fa0/14  
Switchport: Enabled  
Administrative Mode: dynamic desirable  
Operational Mode: trunk
```

```
SW2#show interfaces fa0/14 switchport  
Name: Fa0/14  
Switchport: Enabled  
Administrative Mode: dynamic auto  
Operational Mode: trunk
```

Observe que nessa combinação o trunk foi formado, vamos testar outras combinações:

```
SW1(config)#interface fa0/14
```

```
SW1(config-if)#switchport mode dynamic auto
```

```
SW2(config)#interface fa0/14
```

```
SW2(config-if)#switchport mode trunk
```

```
SW1#show interfaces f0/14 switchport
```

```
Name: Fa0/14
```

```
Switchport: Enabled
```

```
Administrative Mode: dynamic auto
```

```
Operational Mode: trunk
```

```
SW2#show interfaces fa0/14 switchport
```

```
Name: Fa0/14
```

```
Switchport: Enabled
```

```
Administrative Mode: trunk
```

```
Operational Mode: trunk
```

No Dynamic auto o switch prefere que aquela interface seja uma porta de acesso, mas se a interface do outro lado for um trunk ele aceita formar um.

```
SW1(config)#interface fa0/14
```

```
SW1(config-if)#switchport mode dynamic auto
```

```
SW2(config)#interface fa0/14
```

```
SW2(config-if)#switchport mode access
```

```
SW1#show interfaces f0/14 switchport
```

```
Name: Fa0/14
```

```
Switchport: Enabled
```

```
Administrative Mode: dynamic auto
```

```
Operational Mode: static access
```

```
SW2#show interfaces fa0/14 switchport
```

```
Name: Fa0/14
```

```
Switchport: Enabled  
Administrative Mode: static access  
Operational Mode: static access
```

Configurando um lado como dynamic auto e o outro lado como access port, o resultado será uma porta de acesso.

```
SW1(config)#interface fa0/14  
SW1(config-if)#switchport mode dynamic desirable
```

```
SW2(config)#interface fa0/14  
SW2(config-if)#switchport mode trunk
```

```
SW1#show interfaces f0/14 switchport  
Name: Fa0/14  
Switchport: Enabled  
Administrative Mode: dynamic desirable  
Operational Mode: trunk
```

```
SW2#show interfaces fa0/14 switchport  
Name: Fa0/14  
Switchport: Enabled  
Administrative Mode: trunk  
Operational Mode: trunk
```

Dynamic desirable e trunk formam uma interface trunk.

O que acontecerá se colocarmos uma interface em modo de acesso e outra como Trunk? Não parece ser uma ideia muito inteligente, mas, vamos testar!

```
SW1(config)#interface fa0/14  
SW1(config-if)#switchport mode access
```

```
SW2(config)#interface fa0/14  
SW2(config-if)#switchport mode trunk
```

```
SW1#show interfaces f0/14 switchport  
Name: Fa0/14
```

```

Switchport: Enabled
Administrative Mode: static access
Operational Mode: trunk

```

```

SW2#show interfaces fa0/14 switchport
Name: Fa0/14
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk

```

```

SW1#
%SPANTREE-7-RECV_1Q_NON_TRUNK: Received 802.1Q BPDU on non trunk FastEthernet0/14
VLAN1.
%SPANTREE-7-BLOCK_PORT_TYPE: Blocking FastEthernet0/14 on VLAN0001. Inconsistent
port type.
%SPANTREE-2-UNBLOCK_CONSIST_PORT: Unblocking FastEthernet0/14 on VLAN0001. Port
consistency restored.

```

Tão logo realizamos as mudanças, apareceram essas mensagens de erro do spanning tree no SW1. Spanning-tree é um protocolo executado em switches para evitar loops na rede.

DTP – Dynamic Trunking Protocol

Todo esse processo de mudança se deu através do protocolo **DTP (Dynamic Trunking Protocol)**. O DTP é um protocolo proprietário desenvolvido pela Cisco, e sua finalidade é gerenciar e configurar automaticamente as interfaces trunk.

Vamos a um quadro comparativo para resumir tudo que aprendemos até agora sobre formação da interface trunk através do DTP:

MODE:	Trunk	Access	Dynamic Auto	Dynamic Desirable
Trunk	Trunk	Conectividade limitada	Trunk	Trunk
Access	Conectividade limitada	Acesso	Acesso	Acesso
Dynamic Auto	Trunk	Acesso	Acesso	Trunk
Dynamic Desirable	Trunk	Acesso	Trunk	Trunk

Pratique essas combinações nos simuladores\emuladores, elas serão cobradas no seu exame e também durante sua vida profissional.

2.2.b 802.1Q

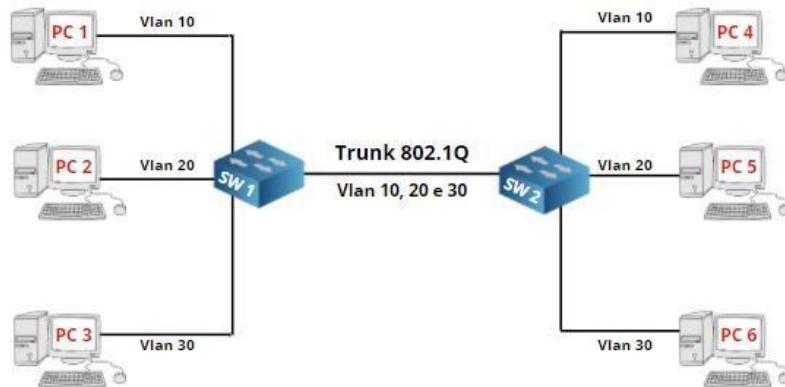
Quando o tráfego de uma VLAN necessita ‘passar’ de um switch para outro, encontramos um problema! Dê uma olhada na imagem abaixo:



Frame Ethernet

Este é um quadro Ethernet, observe que não há nenhum campo para especificar a qual VLAN o frame pertence! Então, como um switch identifica a VLAN daquele frame? Para responder essa pergunta, precisamos utilizar outro protocolo.

Para que o tráfego entre VLANs em diferentes switches aconteça, temos que usar uma interface **trunk**. Uma conexão trunk é simplesmente um link normal, porém esse link é capaz de passar tráfego de diferentes VLANs, pois, possui um método para separar o tráfego entre VLANs. Eis um exemplo:

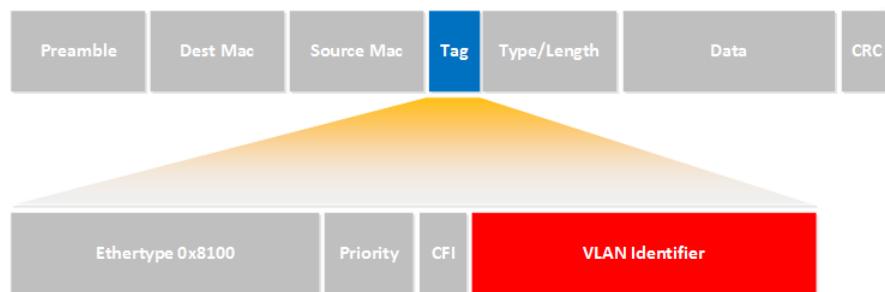


Topologia com trunk

Observe, temos computadores em ambos os lados em VLANs diferentes. Utilizando a interface Trunk, podemos garantir que o tráfego das diferentes VLANs possa ser enviado entre os switches. Como nossos frames Ethernet não têm nada para indicar a qual VLAN aquele frame pertence, utilizamos um ‘protocolo de entroncamento’ (trunking protocol). Existem dois protocolos de entroncamento:

- **802.1Q**: Protocolo trunk mais comum. É um padrão aberto suportado por quase todos os fabricantes.
- **ISL**: Protocolo trunk de propriedade da Cisco, por isso, apenas os switches da Cisco oferecem suporte a ele.

Vamos dar uma olhada melhor no protocolo 802.1Q



Frame 802.1Q

Eis o exemplo de um quadro Ethernet 802.1Q. Perceba, ele é semelhante a um quadro Ethernet normal, mas com um campo a mais. Foi adicionado um campo de tag no meio (representado pelo campo mais escuro). Dentro do campo ‘tag’, encontraremos quatro campos, sendo os principais:

- Vlan identifier: Identificador de VLAN – Indica a qual VLAN o frame Ethernet pertence.
- Priority: Prioridade - Este campo mostra quais frames devem ter prioridade no tráfego. Campo útil para dar prioridade ao tráfego VoIP (por exemplo) em detrimento do tráfego de dados.

2.2.c Native VLAN

O protocolo IEEE 802.1Q descreve a *VLAN nativa*, este padrão nos diz que todo o **tráfego da VLAN nativa é desmarcado**, ou seja, ele não tem uma tag 802.1Q no quadro Ethernet.

Quando um switch recebe um quadro Ethernet sem uma tag através de uma interface trunk, ele assume que esse frame pertence à VLAN nativa. Por esse motivo, é necessário se certificar de que a VLAN nativa é a mesma em ambos os switches.

Por padrão, **VLAN 1 é a VLAN nativa**. Essa vlan pode ser mudada, inclusive, boas práticas de segurança aconselham que ela seja modificada. Vejamos um exemplo.



Topologia com dois switches conectados por uma interface trunk

Vamos configurar uma interface trunk em ambos os switches e em seguida verificar qual vlan está configurada como nativa.

```
SW1(config)#interface Fastethernet 0/24
SW1(config-if)#switchport trunk encapsulation dot1q
SW1(config-if)#switchport mode trunk
```

```
SW2(config)#interface Fastethernet 0/24
SW2(config-if)#switchport trunk encapsulation dot1q
SW2(config-if)#switchport mode trunk
```

Vamos verificar a configuração com o comando ‘**show interface fastethernet 0/24 trunk**’

```
SW1#show interface fastEthernet 0/24 trunk

Port        Mode          Encapsulation  Status      Native vlan
Fa0/24      on           802.1q        trunking    1

Port        Vlans allowed on trunk
Fa0/24      1-4094

Port        Vlans allowed and active in management domain
Fa0/24      1,10,12-13,20,23,34,100,123

Port        Vlans in spanning tree forwarding state and not pruned
Fa0/24      1,10,12-13,20,23,34,100,123
```

```
SW2#show interfaces fastEthernet 0/24 trunk
```

Port	Mode	Encapsulation	Status	Native vlan
Fa0/24	on	802.1q	trunking	1
Port Vlans allowed on trunk				
Fa0/24	1-4094			
Port Vlans allowed and active in management domain				
Fa0/24	1,10,12-13,20,23-24,30			
Port Vlans in spanning tree forwarding state and not pruned				
Fa0/24	1,10,12-13,20,23-24,30			

Observe que o trunk está operacional, estamos usando encapsulamento 802.1Q e a VLAN nativa é a de número 1.

Vamos alterar a vlan nativa para vlan 10:

```
SW1(config)#interface fastEthernet 0/24
SW1(config-if)#switchport trunk native vlan 10
```

```
SW2(config)#interface fastEthernet 0/24
SW2(config-if)#switchport trunk native vlan 10
```

Mais uma vez, usaremos o comando ‘**show interface fastethernet 0/24 trunk**’ para verificarmos:

```
SW1#show interfaces fastEthernet 0/24 trunk
```

Port	Mode	Encapsulation	Status	Native vlan
Fa0/24	on	802.1q	trunking	10

```
SW2#show interfaces fastEthernet 0/24 trunk
```

Port	Mode	Encapsulation	Status	Native vlan
Fa0/24	on	802.1q	trunking	10

Antes de encerrarmos esse assunto, há mais uma coisa a ensinar. É possível configurar o switch para marcar (tag) a vlan nativa como qualquer outra vlan:

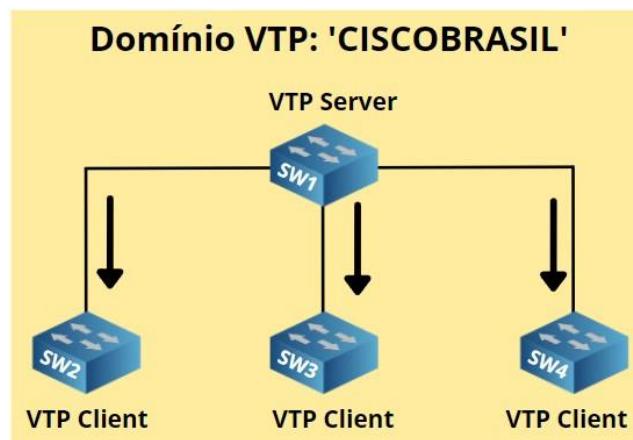
```
SW1(config)#vlan dot1q tag native
```

```
SW2(config)#vlan dot1q tag native
```

VTP (VLAN Trunking Protocol)

Embora não esteja no escopo do CCNA, esse assunto é importantíssimo, não me sentiria bem em ignorá-lo.

Vamos supor que temos uma rede com 20 switches e 50 VLANs. Normalmente, teríamos que configurar cada switch separadamente e criar essas VLANs em cada um desses switches. Essa é uma tarefa demorada e sujeita a erros, porém para facilitar nossa vida, o protocolo VTP (VLAN Trunking Protocol) foi criado. O VTP permite que criemos VLANs em apenas um switch e esse switch replique essas VLANs para todos os outros switches da rede.

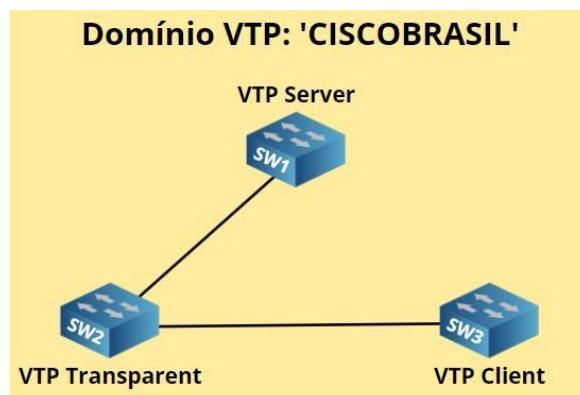


Temos um servidor VTP que é o switch onde podemos criar, modificar e excluir VLANs. Os outros switches são clientes VTP. A configuração do VTP tem um número de revisão que aumentará toda vez que realizarmos alguma alteração. Toda alteração é sincronizada logo em seguida nos clientes VTP. A configuração do VTP é bem simples, só é necessário configurar um nome de domínio VTP e depois configurar o mesmo nome em todos os switches clientes.

Resumindo o que vimos até aqui:

- VTP adiciona / modifica / exclui VLANs.
- Para cada mudança, o número da revisão aumentará.
- A mudança mais recente será enviada a todos os clientes VTP.
- Os clientes VTP sincronizarão com as informações mais recentes.

Além do servidor VTP e do cliente VTP também existe um outro modo VTP, o VTP transparente (VTP Transparent) que é um pouco diferente dos anteriores, eis um exemplo para melhor compreensão:



O switch ‘VTP Transparent’ encaminhará anúncios de mudança, mas não se sincronizará. Com ele, é possível criar VLANs localmente, o que é impossível no VTP Client. Caso criemos a VLAN 20 no servidor VTP, acontecerá os seguintes passos:

1. Criamos a VLAN 20 no VTP Server.
2. O número de revisão aumentará.
3. O servidor VTP encaminhará o anúncio mais recente para o switch VTP transparent.
4. O VTP transparent não se sincronizará, mas encaminhará o anúncio ao cliente VTP.
5. O cliente VTP se sincronizará com as informações mais recentes.

Eis um quadro comparativo dos três modos VTP:

	VTP Server	VTP Client	VTP Transpartent
Cria\Modifica\Exclui VLANs	Sim	Não	Somente local
Sincronização	Sim	Sim	Não
Encaminha anúncios	Sim	Sim	Sim

O VTP é uma ferramenta útil, porém oferece um grande risco de segurança. O problema com o VTP é que um servidor VTP também é um cliente VTP, e qualquer cliente VTP se sincronizará com o número de revisão mais alto. A seguinte situação pode acontecer com VTP:

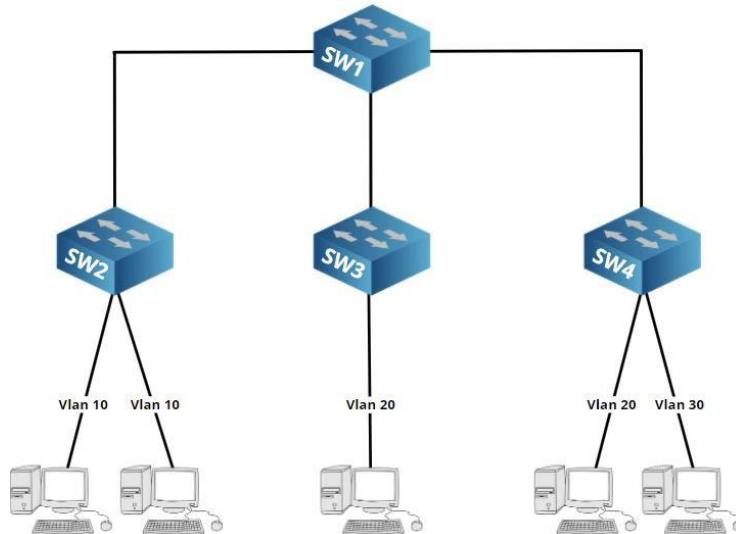
Vamos supor que temos uma rede com um único servidor VTP e alguns switches clientes, tudo está funcionando bem até que um belo dia necessitamos realizar alguns testes e retiramos um dos clientes VTP da rede e em seguida colocamos em um ambiente de laboratório.

1. Retiramos o switch ‘VTP client’ da rede.
2. Configuramos para que não seja mais um cliente VTP, mas um servidor VTP.
3. Realizamos os testes, criando algumas VLANs, modificando, etc.
4. Cada alteração que fazemos, o número de revisão aumenta.
5. Assim que terminarmos o teste, excluímos todas as VLANs.
6. Fazemos o rollback, configurando o modo de VTP Server para o VTP Client.
7. Reconectamos o switch à rede de produção.

Qual será o resultado? O número de revisão VTP do switch que realizamos os testes é maior que o número de revisão dos switches da rede de produção. O cliente VTP anunciará suas informações para os outros switches, eles sincronizarão com as informações mais recentes e então, todas as VLANs desaparecerão! Um cliente VTP pode **sobrescrever** um servidor VTP se o número de revisão for maior, afinal um servidor VTP também é um cliente VTP.

Sim, parece bobo, mas funciona dessa forma! Isso é um grande risco, pois perderemos todas as informações das VLANs.

Mais um detalhe sobre o VTP, observe a imagem abaixo:

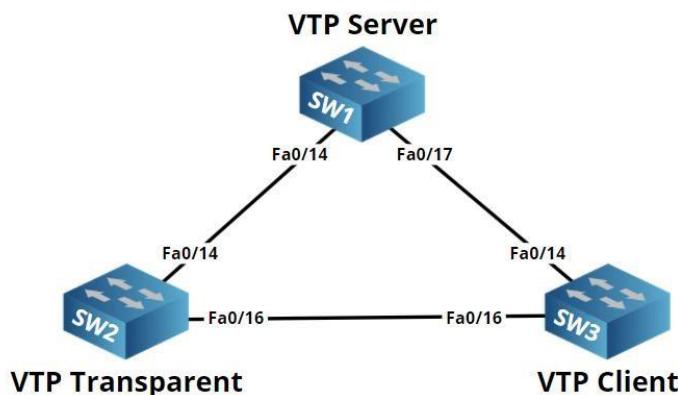


Observe, temos computadores nas VLANs 10, 20 e 30. Os links entre os switches são trunks que usam o protocolo 802.1Q e transportam todo o tráfego da VLAN. Um de nossos computadores na VLAN 10 envia um frame de broadcast, para onde esse frame de broadcast irá?

Os frames de broadcast devem ser inundados por nossos switches e, como nossas interfaces trunks transportam todas as VLANs, esse broadcast irá para todos os lugares. Porém, se você olhar para o switch do meio, observará que não há nenhum computador na VLAN 10, apenas na VLAN 20. Isso significa que esse pacote broadcast está apenas consumindo largura de banda de forma inútil.

Para evitar esse desperdício, temos que usar uma ferramenta chamada ‘VTP pruning’, com ele teremos certeza que não haverá tráfego desnecessário de VLANs nas interfaces trunk quando não houver nenhum dispositivo para aquela VLAN específica. Dependendo do modelo do switch, a ‘VTP pruning’ virá ativado ou desativado por padrão.

Agora, vamos ver como acontece a configuração do VTP, para essa tarefa usaremos três switches, todos eles estão com sem nenhuma configuração:



```
SW1#show vtp status
VTP Version : running VTP1 (VTP2 capable)
Configuration Revision : 0
Maximum VLANs supported locally : 1005
Number of existing VLANs : 5
VTP Operating Mode : Server
```

```
VTP Domain Name : 
VTP Pruning Mode : Disabled
VTP V2 Mode : Disabled
VTP Traps Generation : Disabled
MD5 digest : 0x57 0xCD 0x40 0x65 0x63 0x59 0x47 0xBD
Configuration last modified by 0.0.0.0 at 0-0-00 00:00:00
Local updater ID is 0.0.0.0 (no valid interface found)
```

```
SW2#show vtp status

VTP Version : running VTP1 (VTP2 capable)
Configuration Revision : 0
Maximum VLANs supported locally : 1005
Number of existing VLANs : 5
VTP Operating Mode : Server
VTP Domain Name :
VTP Pruning Mode : Disabled
VTP V2 Mode : Disabled
VTP Traps Generation : Disabled
MD5 digest : 0x57 0xCD 0x40 0x65 0x63 0x59 0x47 0xBD
Configuration last modified by 0.0.0.0 at 0-0-00 00:00:00
Local updater ID is 0.0.0.0 (no valid interface found)
```

```
SW3#show vtp status

VTP Version : 2
Configuration Revision : 0
Maximum VLANs supported locally : 1005
Number of existing VLANs : 5
VTP Operating Mode : Server
VTP Domain Name :
VTP Pruning Mode : Disabled
VTP V2 Mode : Disabled
VTP Traps Generation : Disabled
MD5 digest : 0x57 0xCD 0x40 0x65 0x63 0x59 0x47 0xBD
Configuration last modified by 0.0.0.0 at 0-0-00 00:00:00
Local updater ID is 0.0.0.0 (no valid interface found)
```

Dependendo do modelo do switch, a saída do comando ‘show vtp status’ será semelhante a saída acima. Há alguns detalhes interessantes na saída desse comando:

- Configuration revision 0: Cada vez que adicionamos ou removemos VLANs, esse número muda. No momento o número é 0, pois não criamos ou removemos nenhuma VLAN.
- VTP Operating mode: O padrão é VTP Server.
- VTP Pruning: Ajudará a evitar tráfego desnecessário nos links trunks.
- VTP V2 Mode: O switch é capaz de executar a versão 2 do VTP, porém, atualmente ele está executando a versão 1.

Vamos criar uma VLAN no SW1 para analisarmos o comportamento da rede:

```
SW1(config)#vlan 10  
SW1(config-vlan)#name Impressoras
```

Vlan criada, vamos ver a database de vlans do switch:

```
SW1#show vlan  
  
VLAN Name          Status    Ports  
-----  
1     default       active    Fa0/1, Fa0/2, Fa0/3, Fa0/4  
                                Fa0/5, Fa0/6, Fa0/7, Fa0/8  
                                Fa0/9, Fa0/10, Fa0/11, Fa0/12  
                                Fa0/13, Fa0/14, Fa0/15, Fa0/22  
                                Fa0/23, Fa0/24, Gi0/1, Gi0/2  
  
10    Impressoras   active
```

Vamos ver se houve alguma mudança na ‘Configuration Revision’ (número de revisão):

```
SW1#show vtp status  
  
VTP Version        : running VTP1 (VTP2 capable)  
  
Configuration Revision : 1
```

Vamos verificar os demais switches:

```
SW2#show vtp status  
  
VTP Version        : running VTP1 (VTP2 capable)  
  
Configuration Revision : 0
```

```
SW3#show vtp status  
  
VTP Version        : 2  
  
Configuration Revision : 0
```

Observe que não houve nenhuma modificação no SW2 ou SW3, o motivo é porque ainda não configuramos o **VTP domain-name**. Mas antes de configura-lo, vamos habilitar o debug para acompanhamos em tempo real as trocas de informações na rede:

```
SW2#debug sw-vlan vtp events  
vtp events debugging is on
```

```
SW3#debug sw-vlan vtp events  
vtp events debugging is on
```

Agora, vamos configurar o domínio VTP, para isso basta digitarmos com comando ‘vtp domain’ mais o nome do domínio:

```
SW1(config)#vtp domain Luiz_Silverio  
Changing VTP domain name from NULL to Luiz_Silverio
```

```
SW2#  
VTP LOG RUNTIME: Summary packet received in NULL domain state  
VTP LOG RUNTIME: Summary packet received, domain = Luiz_Silverio, rev = 1, followers = 1,  
length 77, trunk Fa0/16  
VTP LOG RUNTIME: Transitioning from NULL to Luiz_Silverio domain  
VTP LOG RUNTIME: Summary packet rev 1 greater than domain Luiz_Silverio S rev 0
```

Há duas coisas interessantes que podemos ver com o comando acima:

- O switch recebe um pacote VTP do domínio “Luiz_Silverio” e decide mudar seu próprio nome de domínio de “NULL” (nada) para “Luiz_Silverio”. Ele só mudará o nome de domínio se ele ainda não tiver um nome de domínio.
- O switch vê que o pacote VTP tem um número de revisão maior (1) do que ele tem atualmente (0) e, como resultado, ele se sincronizará.

Vamos desabilitar o debug em ambos os switches:

```
SW2#no debug all  
All possible debugging has been turned off
```

```
SW3#no debug all  
All possible debugging has been turned off
```

O número de revisão no SW2 e SW3 agora é ‘1’:

```
SW2#show vtp status  
VTP Version : running VTP1 (VTP2 capable)  
Configuration Revision : 1
```

```
SW3#show vtp status  
VTP Version : 2  
Configuration Revision : 1
```

E todas as vlan estão sincronizadas, o que demonstra que foram aprendidas através do VTP:

```
SW2#show vlan

VLAN Name          Status    Ports
-----
1     default       active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                           Fa0/5, Fa0/6, Fa0/7, Fa0/8
                           Fa0/9, Fa0/10, Fa0/11, Fa0/12
                           Fa0/13, Fa0/14, Fa0/15,
                           Fa0/23, Fa0/24, Gi0/1, Gi0/2
10    Impressoras    active
```

```
SW3#show vlan

VLAN Name          Status    Ports
-----
1     default       active    Fa0/1, Fa0/2, Fa0/3, Fa0/4
                           Fa0/5, Fa0/6, Fa0/7, Fa0/8
                           Fa0/9, Fa0/10, Fa0/11, Fa0/12
                           Fa0/20, Fa0/22, Fa0/23,
                           Gi0/1, Gi0/2
10    Impressoras    active
```

Como todos os switches estão no modo VTP Server, é possível criar VLANs em qualquer switch e todos serão sincronizados, vamos criar a vlan 20 no SW2 e vlan 30 no SW3:

```
SW2 (config) # vlan 20
SW2 (config-vlan) # name Servidores
```

```
SW3 (config) # vlan 30
SW3 (config-vlan) # name Usuarios
```

Vamos verificar se as vlans foram replicadas para todos os switches:

```
SW1#show vlan

VLAN Name          Status    Ports
-----
10    Impressoras    active
20    Servidores     active
30    Usuarios       active
```

```
SW2#show vlan
VLAN Name          Status    Ports
-----
10    Impresoras      active
20    Servidores      active
30    Usuarios        active
```

```
SW3#show vlan
VLAN Name          Status    Ports
-----
10    Impresoras      active
20    Servidores      active
30    Usuarios        active
```

Vamos analisar o número da ‘Configuration Revision’:

```
SW1#show vtp status
VTP Version        : running VTP1 (VTP2 capable)
Configuration Revision : 3
```

```
SW2#show vtp status
VTP Version        : running VTP1 (VTP2 capable)
Configuration Revision : 3
```

```
SW3#show vtp status
VTP Version        : 2
Configuration Revision : 3
```

Cada vez que criamos uma vlan o número da ‘Configuration Revision’ foi incrementado. Vamos mudar o modo de operação do SW2 para para cliente:

```
SW2(config)#vtp mode client
Setting device to VTP CLIENT mode.
```

```
SW2#show vtp status
VTP Version        : running VTP1 (VTP2 capable)
Configuration Revision : 3
Maximum VLANs supported locally : 1005
```

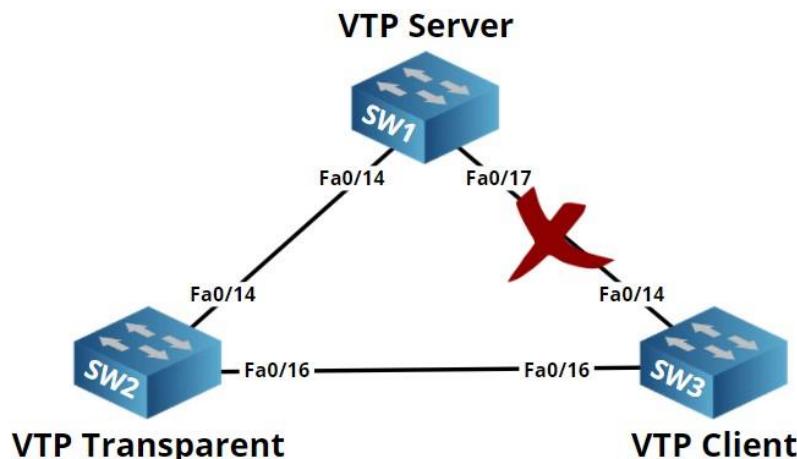
```

Number of existing VLANs      : 7
VTP Operating Mode           : Client

```

O SW2 agora está sendo executado no modo VTP Client.

No momento SW1 e SW3 estão no modo VTP Server. SW2 está executando o modo VTP Client. Vamos desconectar a conexão entre SW1 e SW3, para que não haja mais conexão direta entre eles.



Vamos criar outra Vlan no Sw1 e ver como será o comportamento do SW2 e SW3:

```

SW1 (config) # vlan 40
SW1 (config-vlan) # name Engenharia

```

Agora vamos verificar o SW2 e SW3:

```

SW2#show vlan
-----  

VLAN Name          Status    Ports  

-----  

10    Impressoras   active  

20    Servidores    active  

30    Usuarios      active  

40    Engenharia    active

```

O SW2 aprendeu a vlan 40 do SW1.

```

SW3#show vlan
-----  

VLAN Name          Status    Ports  

-----  

10    Impressoras   active  

20    Servidores    active  

30    Usuarios      active  

40    Engenharia    active

```

SW3 aprendeu a VLAN 40 do SW2. O SW2 como um VTP Client não só irá se sincronizar, como também encaminhará anúncios VTP. Vamos tentar criar uma vlan no SW2:

```
SW2(config)#vlan 50  
%VTP VLAN configuration not allowed when device is in CLIENT mode.
```

Como esperado, um switch que esteja no modo VTP Client não consegue criar VLANs, por isso recebemos essa mensagem de erro.

E o modo VTP Transparent? Vamos mudar o SW2 para o modo VTP Transparent, lembrando que o link entre SW1 e SW3 ainda está desconectado.

```
SW2(config)#vtp mode transparent  
Setting device to VTP TRANSPARENT mode.
```

Vamos criar uma nova vlan no SW1 chamada de ‘Pesquisa’:

```
SW1(config)#vlan 50  
SW1(config-vlan)#name Pesquisa
```

Vamos verificar:

```
SW1#show vlan  
  
VLAN Name          Status    Ports  
-----  
10    Impressoras    active  
20    Servidores     active  
30    Usuarios       active  
40    Engenharia     active  
50    Pesquisa        active
```

Vamos verificar agora se essa vlan aparecerá no SW2:

```
SW2#show vlan  
  
VLAN Name          Status    Ports  
-----  
10    Impressoras    active  
20    Servidores     active  
30    Usuarios       active  
40    Engenharia     active
```

Conforme esperado, a vlan não apareceu no SW2, lembre-se, o SW2 está em modo transparente, portanto ele não sincronizará com os demais switches.

Vamos verificar o SW3:

```
SW1#show vlan  
  
VLAN Name          Status    Ports  
-----  
10    Impressoras    active
```

20	Servidores	active
30	Usuarios	active
40	Engenharia	active
50	Pesquisa	active

Observe que a vlan foi configurada no SW3! Um switch no modo VTP Transparent não sincronizará a si mesmo, mas encaminhará anúncios VTP a outros switches para que eles possam se sincronizar!

Vamos descobrir o que acontecerá se criarmos uma vlan no SW2:

```
SW2(config)#vlan 60
SW2(config-vlan)#name Cameras
```

SW2#show vlan			
VLAN Name	Status	Ports	
<hr/>			
10 Impressoras	active		
20 Servidores	active		
30 Usuarios	active		
40 Engenharia	active		
50 Pesquisa	active		
60 Cameras	active		

Como esperado, podemos criar esta nova VLAN no SW2 sem nenhum problema, afinal, ele está no modo VTP Transparent. Vamos verificar se houve alguma mudança nas vlans do SW1 e SW3:

SW1#show vlan			
VLAN Name	Status	Ports	
<hr/>			
10 Impressoras	active		
20 Servidores	active		
30 Usuarios	active		
40 Engenharia	active		
50 Pesquisa	active		

SW3#show vlan			
VLAN Name	Status	Ports	
<hr/>			
10 Impressoras	active		
20 Servidores	active		

30	Usuarios	active
40	Engenharia	active
50	Pesquisa	active

A VLAN 60 não aparece nos SW1 e SW3 porque o SW2 está no modo VTP Transparente, portanto, o SW2 não anunciará suas vlans, elas serão conhecidas apenas localmente.

Há mais um detalhe que você precisa saber sobre o modo VTP Transparent. Observe o **show running** do SW2:

```
SW2#show running-config
Building configuration...
vlan 10
  name Impressoras
!
vlan 20
  name Servidores
!
vlan 30
  name Usuarios
!
vlan 40
  name Pesquisa
!
vlan 60
  name Cameras
```

Há uma diferença entre o modo VTP Transparent e os modos VTP Server e Client. Quando estamos trabalhando com um switch no modo VTP Transparent ele armazena todas as informações de VLAN na **running-config**. O modo VTP Server e Client armazenam suas informações no banco de dados de VLAN (arquivo **vlan.dat** na memória flash).

2.3 Configure and verify Layer 2 discovery protocols (Cisco Discovery Protocol and LLDP)

Vamos falar de duas ferramentas importantíssimas para gerenciamento de redes, o Cisco Discovery Protocol (CDP) e o Link Layer Discovery Protocol (LLDP).

Quando falamos de gerenciamento de rede, estamos nos referindo ao controle, gerenciamento e manutenção da rede, garantindo assim, que os dispositivos de rede se comuniquem entre si sem falhas. Um bom gerenciamento de redes, previne problemas e propicia soluções mais rápidas em caso de problemas, além do aprimoramento de desempenho por meio do monitoramento de tráfego, detecção de intrusão e de falhas.

Para gerenciar as redes, usamos o **Cisco Discovery Protocol (CDP)** e o **Link Layer Discovery Protocol (LLDP)**, que reúnem informações sobre os dispositivos vizinhos, úteis para decisões de design, solução de problemas e documentação de rede.

CDP - Cisco Discovery Protocol

A maioria das redes são formadas por vários switches e roteadores, e para facilitar nossa vida, é bom ter um mapa da rede que nos mostre a forma que os equipamentos estão conectados entre si, que tipo de dispositivos temos, em que VLAN estão

e os endereços IP que estão sendo usados. O **CDP** é um protocolo de propriedade da Cisco que pode ser executado em qualquer um dos seus dispositivos, ele roda na camada 02 do modelo OSI (data-link layer) e vem habilitado por padrão.

Observe a topologia abaixo:



Imagine que não conhecemos essa topologia de rede, e não sabemos da existência desses três roteadores, sabemos apenas da existência do Roteador R1. Com auxílio do CDP conseguimos montar a topologia apenas com essa informação:

```

R1#show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater

Device ID      LocalIntrfce     Holdtme     Capability   Platform  Port ID
R2              Ser 0/0          167         R S I       3640      Ser 0/0
  
```

A partir do comando ‘**show cdp neighbours**’ é possível ver todos os vizinhos **diretamente conectados**. Descobrimos com o comando acima que o R1 está conectado ao R2, também foi possível descobrir a plataforma (roteador 3640) e as interfaces em ambos os lados. Vamos ver a saída nos outros roteadores:

```

R2#show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater

Device ID      Local Intrfce     Holdtme     Capability   Platform  Port ID
R1              Ser 0/0          144         R S I       3640      Ser 0/0
R3              Fas 1/0          164         R S I       3640      Fas 1/0
  
```

```

R3#show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater

Device ID      Local Intrfce     Holdtme     Capability   Platform  Port ID
R2              Fas 1/0          135         R S I       3640      Fas 1/0
  
```

Agora, nós temos todas as informações necessárias para montar a topologia da rede, como nome, interfaces... Porém, o CDP ainda pode nos informar mais detalhes:

```
R1#show cdp neighbors detail
```

```

-----
Device ID: R2
Entry address(es):
    IP address: 192.168.12.2
Platform: Cisco 3640, Capabilities: Router Switch IGMP
Interface: Serial0/0, Port ID (outgoing port): Serial0/0
Holdtime : 136 sec
Version :
Cisco IOS Software, 3600 Software (C3640-JK903S-M), Version 12.4(16), RELEASE
SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2007 by Cisco Systems, Inc.
Compiled Wed 20-Jun-07 11:43 by prod_rel_team
advertisement version: 2
VTP Management Domain: ''

```

Através do comando ‘**show cdp neighbors detail**’ obtemos ainda mais informações. Por exemplo, é possível ver endereço IP e a versão do IOS. Isso pode ser muito útil em algumas situações, mas também é um risco à segurança. Por padrão, o CDP vem habilitado e é executado em todas as interfaces, portanto, consiste em uma boa prática de segurança desabilitá-lo em certas interfaces:

```

R1(config)#interface serial 0/0
R1(config-if)#no cdp enable

```

Para desabilitar o CDP em uma única interface, basta digitar ‘**no cdp enable**’ dentro daquela interface. Caso a ideia seja desabilitar o CDP em todos as interfaces usamos o seguinte comando em modo global:

```
R1(config)#no cdp run
```

Além de revelar informações da rede, o CDP também é usado pelos telefones IPs da Cisco. Lembre-se, o CDP é executado apenas em hardware Cisco; há também uma versão “aberta”, chamada LLDP, que é executada em hardware Cisco e em equipamentos de outros fabricantes.

LLDP - Link Layer Discovery Protocol

LLDP é um protocolo de descoberta de camada dois, semelhante ao CDP da Cisco. A grande diferença entre os dois é que o LLDP é um padrão aberto, enquanto o CDP é um protocolo proprietário da Cisco.

Os dispositivos Cisco suportam a versão IEEE 802.1ab do LLDP. Isso permite que dispositivos não Cisco anunciem informações sobre si mesmos para os dispositivos Cisco que compõe a rede.

O LLDP usa atributos que contêm descrições de tipo, comprimento e valor (type, length and value descriptions). Eles são chamados de TLVs (type, length and value). Dispositivos que suportam LLDP usam TLVs para enviar e receber informações para seus vizinhos que estão diretamente conectados. Aqui está um exemplo de alguns TLVs básicos:

- Port description TLV

- System name TLV
- System description TLV
- System capabilities TLV
- Management Address TLV

Alguns endpoints (como telefones IP) podem usar LLDP para atribuição de VLAN ou requisitos de PoE (Power over Ethernet). Para isso, foi feito um aprimoramento denominado MED (Media Endpoint Discovery), popularmente conhecido como LLDP-MED.

A configuração do LLDP é bem simples, e pode vir habilitado ou desabilitado por padrão, dependendo do modelo do switch e da versão do IOS. Vamos dar uma olhada em um exemplo:



Acima temos dois switches Cisco da linha Catalyst 3560, conectados diretamente um ao outro. O LLDP vem desabilitado por padrão nesses modelos, vamos habilitá-lo:

```
SW1(config)#lldp run
```

```
SW2(config)#lldp run
```

Esse comando ativa globalmente o LLDP em todas as interfaces. Vamos aplicar o comando ‘**show lldp neighbors**’ para verificar:

```

SW1#show lldp neighbors
Capability codes:
  (R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
  (W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other
Device ID          Local Intf      Hold-time  Capability      Port ID
SW2                Fa0/24           120        B              Fa0/24
Total entries displayed: 1
  
```

A saída do comando é quase igual ao do CDP, e assim como no CDP, é possível detalhar um pouco mais:

```

SW1#show lldp neighbors detail
Chassis id: 0011.bb0b.361a
Port id: Fa0/24
Port Description: FastEthernet0/24
System Name: SW2.ciscobrasil.com
System Description:
Cisco IOS Software, C3560 Software (C3560-ADVIPSERVICESK9-M), Version 12.2(46)SE,
RELEASE SOFTWARE (fc2)
  
```

```

Copyright (c) 1986-2008 by Cisco Systems, Inc.

Compiled Thu 21-Aug-08 15:26 by nachen

Time remaining: 106 seconds

System Capabilities: B,R

Enabled Capabilities: B

Management Addresses - not advertised

Auto Negotiation - supported, enabled

Physical media capabilities:

    100base-TX(FD)

    100base-TX(HD)

    10base-T(FD)

    10base-T(HD)

Media Attachment Unit type: 16

-----
Total entries displayed: 1

```

Agora temos quase uma dezena de informações detalhadas sobre o SW2, como o nome de host, plataforma, versão do IOS, recursos, etc. Um recurso interessante que o LLDP oferece, é que ele também envia descrições da interface. Observe o exemplo abaixo:

```

SW1(config)#interface FastEthernet 0/24
SW1(config-if)#description Trunk_entre_SW1_SW2

```

Essa descrição aparece quando olharmos a saída do comando no SW2:

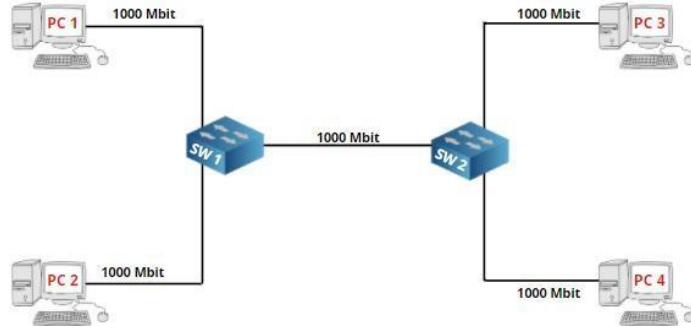
```

SW2#show lldp neighbors detail
Chassis id: 0019.569d.571a
Port id: Fa0/24
Port Description: Trunk_entre_SW1_SW2
System Name: SW1.ciscobrasil.com

```

2.4 Configure and verify (Layer 2/Layer 3) EtherChannel (LACP)

EtherChannel é uma tecnologia que permite agrupar vários links físicos em um único link lógico, também é conhecido como agregação de links (Link aggregation). A seguir, estudaremos como funciona e quais as vantagens do EtherChannel. Vamos começar com a topologia abaixo:



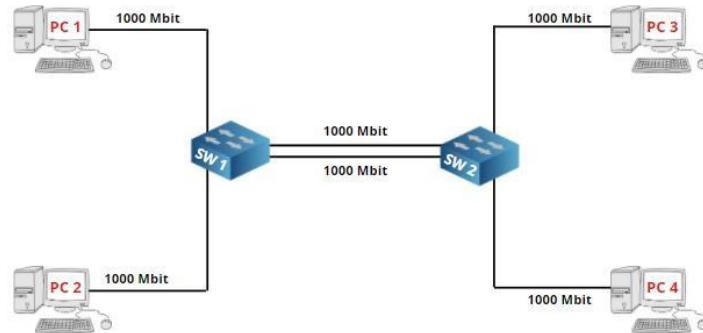
Na topologia acima, temos dois switches e dois computadores conectados a cada um desses switches, todos os computadores estão utilizando interfaces Gigabit Ethernet.

Agora imagine que o PC1 envie 800 Mbit de tráfego para o PC3 e o PC2 envie 600 Mbit de tráfego para o PC4. Pela quantidade de tráfego enviado teremos um gargalho entre os switches. Afinal, estamos enviando 800 + 600 o que dá 1400 Mbit, porém, o link entre o Sw1 e Sw2 é de apenas 1000 Mbit.

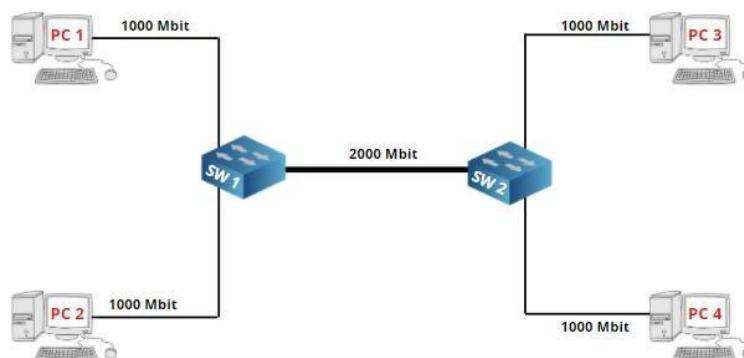
Há duas soluções para este problema:

- Substituir o link entre os switches por uma conexão que tenha uma largura de banda maior, talvez um link de 10 Gigabit.
- Adicionar vários links e agrupa-los em um EtherChannel.

Vamos adotar essa segunda hipótese e adicionar mais links entre os switches:



Observe que adicionamos um novo link, porém, caso deixássemos simplesmente dessa forma, teríamos um problema. Com essa configuração teremos um loop entre os switches, forçando o spanning-tree (ainda falaremos sobre ele) a bloquear um dos dois links. O EtherChannel resolve esse problema, pois cria um **único link virtual** a partir desses links físicos:



O Spanning-tree verá os dois links como apenas um link lógico, portanto, não **haverá loops!** Dois outros detalhes que tornam o EtherChannel ainda mais interessante: Ele fará o **balanceamento de carga** entre os links e cuidará da redundância, ou seja, se um dos links falhar, ele continuará funcionando e usará o link restante.

Lembre-se de que os links físicos continuam sendo um fator limitante. Um único fluxo de tráfego não poderá exceder a mais de 1000 Mbit (link Gigabit único). Um Etherchannel é o equivalente a adicionar mais pistas a uma rodovia. A largura de banda aumenta, mas o limite de velocidade não muda.

Podemos atribuir até 16 interfaces físicas a um EtherChannel, mas **apenas 8 interfaces** estarão ativas por vez. Existem três opções para configurar um EtherChannel:

- PAgP (propriedade da Cisco)
- LACP (padrão IEEE)
- Manual

PAgP e LACP são protocolos de negociação que configuram dinamicamente um Etherchannel. PAgP é um protocolo proprietário da Cisco, portanto, só é possível usá-lo entre dispositivos Cisco. LACP é um padrão IEEE adotado por vários fabricantes, inclusive a Cisco. Também é possível configurar um EtherChannel estático, sem que esses protocolos cuidem da negociação para formação do EtherChannel.

Para que haja a formação do EtherChannel, é necessário que todas as interfaces que participaram tenham a mesma configuração, ou seja:

- Duplex.
- Velocidade.
- VLANs nativas e vlans permitidas.
- Modo switchport (acesso ou trunk).

O PAgP e o LACP irão verificar automaticamente se a configuração das interfaces são as mesmas em ambos os lados.

O protocolo PAgP não foi cobrado nessa nova versão do CCNA, porém, é importante que você saiba da sua existência e que tenha um entendimento básico do seu funcionamento. Isto posto, é hora de aprendermos a configurar o LACP.

Configuração do LACP

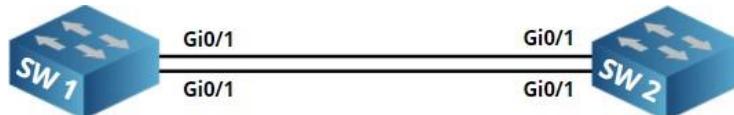
Há dois modos de interface que precisamos conhecer antes de configurarmos o LACP, são eles:

- **Ativo:** A interface solicitará ativamente ao outro lado para se tornar um EtherChannel.
- **Passivo:** A interface espera passivamente que o outro lado peça para se tornar um EtherChannel.

Para que o EtherChannel se forme, pelo menos um dos switches deve usar o modo ativo. Quando ambos os switches utilizam o modo passivo, nada acontece.

	Ativo	Passivo
Ativo	Sim	Sim
Passivo	Sim	Não

Vamos aprender como isso funciona na prática, observe a topologia abaixo:



Essa são as opções de configuração para o LACP:

```

SW1(config)#interface GigabitEthernet 0/1
SW1(config-if)#channel-group 1 mode ?
      active      Enable LACP unconditionally
  
```

```

auto      Enable PAgP only if a PAgP device is detected

desirable  Enable PAgP unconditionally

on        Enable EtherChannel only

passive   Enable LACP only if a LACP device is detected

```

Nós usamos o comando ‘channel-group’ dentro da interface para informar que aquela interface fará parte de um EtherChannel. Podemos escolher qualquer número para identificação do Channel group, no caso em tela, o número escolhido foi o número 1.

Vamos configurar o SW1 para usar o modo ativo:

```

SW1(config-if)#interface range GigabitEthernet 0/1 - 2
SW1(config-if)#channel-group 1 mode active

```

E o SW2 para utilizar o modo passivo:

```

SW2(config)#interface range GigabitEthernet 0/1 - 2
SW2(config-if)#channel-group 1 mode passive

```

O Switch cria automaticamente uma nova interface chamada ‘channel group’ mais o número de identificação que foi utilizado:

```
SW1 %LINK-3-UPDOWN: Interface Port-channel1, changed state to up
```

Vamos verificar o EtherChannel, e já aproveitarmos para conhecer alguns comandos de verificação:

```

SW1#show etherchannel summary

Flags: D - down      P - bundled in port-channel

I - stand-alone s - suspended

H - Hot-standby (LACP only)

R - Layer3      S - Layer 2

U - in use      N - not in use, no aggregation

f - failed to allocate aggregator

M - not in use, minimum links not met

m - not in use, port not aggregated due to minimum links not met

u - unsuitable for bundling

w - waiting to be aggregated

d - default port

A - formed by Auto LAG

Number of channel-groups in use: 1

Number of aggregators:          1

Group  Port-channel  Protocol    Ports

```

1	Po1(SU)	LACP	Gi0/1(P)	Gi0/2(P)
----------	----------------	-------------	-----------------	-----------------

Observe que o EtherChannel foi formado. Esse comando é interessante pois traz um resumo do Etherchannel, informando se está ativo, qual protocolo foi usado, etc.

Vamos explorar mais alguns comandos de verificação:

```
SW1#show etherchannel 1 port-channel

Port-channels in the group:

Port-channel: Po1      (Primary Aggregator)

Age of the Port-channel = 0d:00h:01m:43s

Logical slot/port = 16/0          Number of ports = 2

HotStandBy port = null

Port state           = Port-channel Ag-Inuse

Protocol            = LACP

Port security       = Disabled

Load share deferral = Disabled

Ports in the Port-channel:

Index  Load  Port    EC state      No of bits

-----
 0     00   Gi0/1  Active        0
 0     00   Gi0/2  Active        0

Time since last port bundled:  0d:00h:01m:04s  Gi0/2
```

E por último, o comando ‘show interfaces’:

```
SW1#show interfaces GigabitEthernet 0/1 etherchannel

Port state      = Up Mstr Assoc In-Bndl

Channel group = 1          Mode = Active          Gcchange = -
Port-channel   = Po1        GC    = -             Pseudo port-channel = Po1
Port index     = 0          Load = 0x00          Protocol = LACP

Flags: S - Device is sending Slow LACPDUs  F - Device is sending fast LACPDUs.
      A - Device is in active mode.        P - Device is in passive mode.

Local information:
```

		LACP port		Admin	Oper	Port	Port	
Port	Flags	State	Priority	Key	Key	Number	State	
Gi0/1	SA	bndl	32768	0x1	0x1	0x2	0x3D	
Partner's information:								
		LACP port		Admin	Oper	Port	Port	
Port	Flags	Priority	Dev ID	Age	key	Key	Number	State
Gi0/1	SP	32768	5e00.0001.8000	18s	0x0	0x1	0x2	0x3C
Age of the port in the current state: 0d:00h:01m:37s								

Observe que estamos trabalhando com o protocolo LACP, temos uma interface port-channel de número ‘1’, composta por duas interfaces físicas, e também temos informações sobre os vizinhos que formam o Etherchannel.

Vamos verificar agora a configuração da interface port-channel de número 1:

```
SW1#show run interface Port-channel 1
interface Port-channel1
end
```

Agora, vamos olhar a configuração das interfaces que fazem parte do EtherChannel, observe que elas só têm a configuração do ‘port-channel’:

```
SW1#show run interface GigabitEthernet 0/1
interface GigabitEthernet0/1
  channel-group 1 mode active
end
```

```
SW1#show run interface GigabitEthernet 0/2
interface GigabitEthernet0/2
  channel-group 1 mode passive
end
```

Caso seja necessário realizar alguma alteração, como configurar o EtherChannel como trunk, a modificação deve ser realizada na interface port-channel. Por exemplo:

```
SW1(config)#interface Port-channel 1
SW1(config-if)#switchport trunk encapsulation dot1q
SW1(config-if)#switchport mode trunk
```

Quando fazemos isso, o switch automaticamente adiciona a configuração as interfaces físicas:

```
SW1#show run interface GigabitEthernet 0/1
interface GigabitEthernet0/1
```

```

switchport trunk encapsulation dot1q
switchport mode trunk
channel-group 1 mode active
end

```

```

SW1#show run interface GigabitEthernet 0/2
interface GigabitEthernet0/2
switchport trunk encapsulation dot1q
switchport mode trunk
channel-group 1 mode passive
end

```

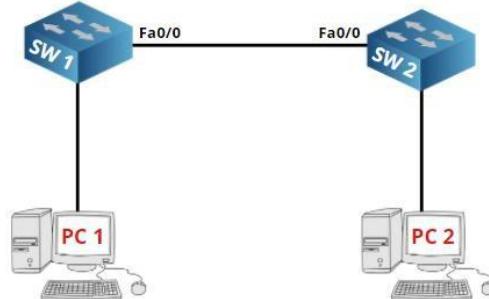
2.5 Describe the need for and basic operations of Rapid PVST+ Spanning Tree Protocol and identify basic operations

Talvez, esse seja o tópico mais importante desse capítulo de ‘acesso a rede’. Spanning-Tree é um tema grande e bem complicado, mas devagar vamos entender seu funcionamento, como é configurado e quando deve ser utilizado.

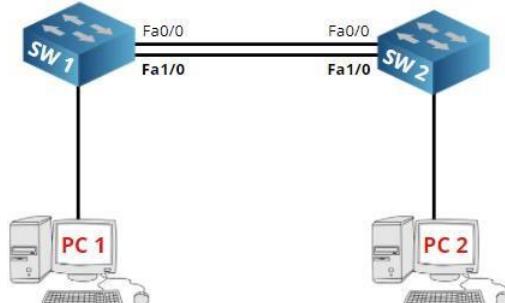
Introdução ao Spanning Tree

Spanning-tree é um protocolo que opera na camada dois do modelo OSI, ou seja, sempre será configurado em switches. A função principal do spanning-tree é eliminar possíveis loops na rede.

Caso você não saiba o que é loop de rede, observe esse exemplo abaixo:



Na imagem acima, temos dois switches conectados através de um único cabo, portanto, há um **único ponto de falha**. Para eliminar esse único ponto de falha adicionaremos outro cabo:



Com mais esse cabo de rede criamos **redundância** entre os switches. Infelizmente, a redundância também provoca **loops**. Vamos entender porque esses loops ocorrem:

O PC1 envia uma solicitação ARP afim de descobrir o endereço MAC do PC2. Uma solicitação ARP é um quadro de **broadcast**.

SW1 irá encaminhar este frame de broadcast em todas as suas interfaces, exceto a interface onde ele recebeu o quadro.

SW2 receberá os dois quadros de broadcast, e encaminhará para todas as interfaces, exceto a interface em que ele recebeu o quadro. Isso significa que o quadro recebido na interface Fa0/0 será encaminhado para interface Fa1/0.

Esse frame recebido na interface Fa1/0 será encaminhado para interface Fa0/0.

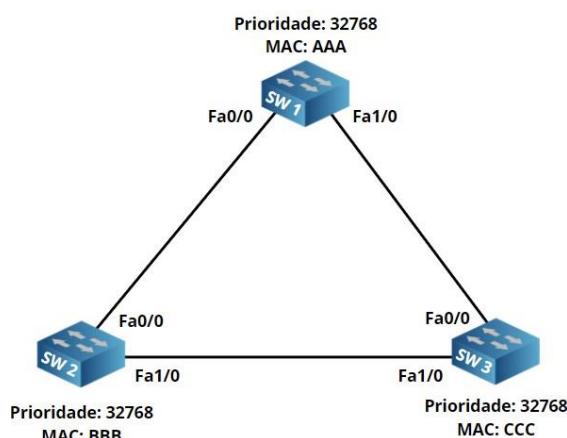
Teremos um cenário de idas e vindas infinito, e o loop estará formado! Ambos os switches continuarão encaminhando esse frame continuamente até que um dos switches trave por estar sobrecarregado.

Só há uma maneira de parar esse loop antes que um dos switches trave: Desconectando um dos cabos.

Os frames Ethernet **não têm um valor TTL** (Time to Live), portanto, eles ficarão em loop para sempre. Além das solicitações ARP, existem muitos outros quadros de broadcast.

2.5.a Root port, root bridge (primary/secondary), and other port names

O Spanning-tree elimina os loops da rede bloqueando algumas interfaces. Abaixo, veremos como se dá o processo de escolha das interfaces que serão bloqueadas.



Na topologia acima temos três switches, e dessa vez adicionamos redundância de um jeito diferente. Não adicionamos só um cabo, adicionamos um novo switch, construindo assim, uma topologia em forma de triângulo. Observe bem e verá que também temos um loop aqui.

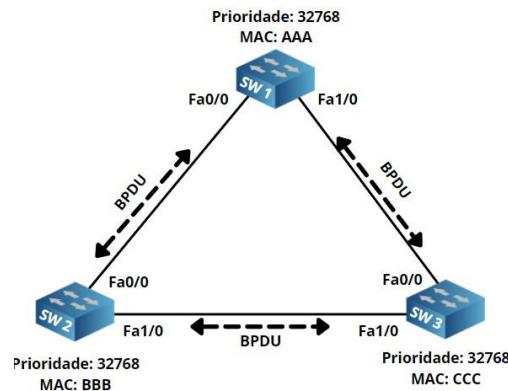
Embaixo de cada switch há um endereço MAC ‘simplificado’ e um número de prioridade. Entenderemos a função dessa prioridade a seguir. Resumindo, nós temos o seguinte cenário:

- SW1: MAC AAA – Prioridade 32768
- SW2: MAC BBB – Prioridade 32768
- SW3: MAC CCC – Prioridade 32768

Assim que o spanning-tree é habilitado, todos os switches enviam um frame ‘especial’, chamado BPDU (Bridge Protocol Data Unit). O BPDU carrega duas informações essenciais para o spanning tree:

- Endereço MAC
- Prioridade

O endereço MAC e a prioridade juntos constituem o ID do **switch (bridge ID)**. O BPDU é enviado entre os switches, conforme mostra a imagem abaixo:

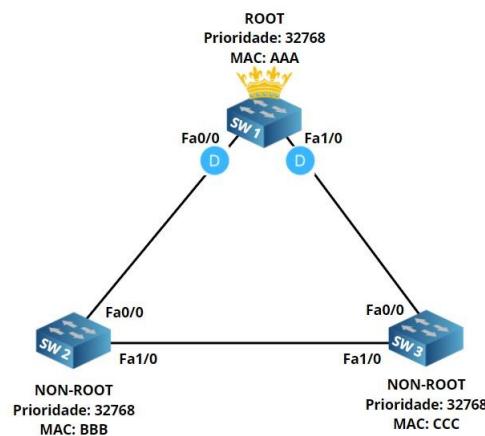


Vamos entender como funciona o cálculo da ‘bridge-ID’:

- Em primeiro lugar, o spanning-tree **elegerá o root bridge**. Root bridge será o switch que tiver o melhor “bridge ID”.
- O switch com o **menor bridge ID** é considerado o melhor switch.
- Por padrão, a prioridade é **32768**, mas esse valor pode ser alterado.

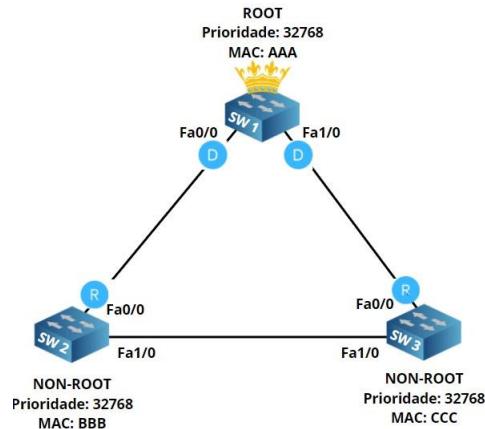
Então, quem se tornará o Root bridge? No exemplo acima será o SW1! Lembre-se, a prioridade e o endereço MAC constituem o bridge ID, como a prioridade é a mesma em todos os switches, o endereço MAC será o responsável por desempatar. SW1 tem o endereço MAC mais baixo, consequentemente possui a melhor bridge ID, tornando-se assim o root bridge.

As portas no Root bridge serão sempre portas **designadas (designated)**, o que significa que estarão sempre no modo de **encaminhamento (forwarding)**. Observe a imagem abaixo:



O SW1 foi eleito como o root bridge. O “D” nas interfaces significa designado.

Agora que já entendemos o processo de eleição do root bridge, a próxima etapa é entendermos o processo que os outros switches utilizam para descobrir e definir o menor caminho até o root bridge! O caminho mais curto para o root bridge é chamado de “**porta raiz**” (**root port**). Dê uma olhada no exemplo:



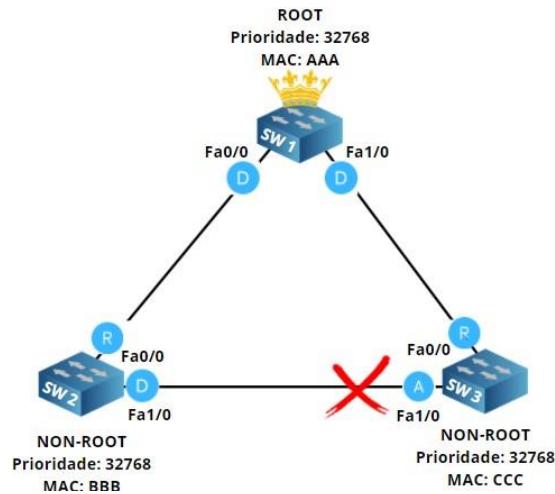
O “R” significa “porta raiz” tanto no SW2 como no SW3. Nos dois switches a interface Fa0/0 é o caminho mais curto para chegar a root bridge. No exemplo acima, não estamos utilizando a verdadeira métrica que o spanning-tree utiliza. Definir o “caminho mais curto” no spanning-tree significa que ele observará a **velocidade da interface**. Cada interface tem um custo e será utilizado o caminho de menor custo. Abaixo, uma visão geral das interfaces e seus custos:

- 10 Mbit = Custo 100
- 100 Mbit = Custo 19
- 1000 Mbit = Custo 4

Temos portas designadas na root bridge e portas raiz nos switches non-root bridges, mas, ainda temos um loop! Precisamos desligar uma porta entre SW2 e SW3 para interromper esse loop. Qual porta vamos bloquear? Bloquearemos no SW2 ou no SW3? Veremos novamente os cálculos para o melhor bridge id. Lembre-se da fórmula:

- ID da ponte = Prioridade + endereço MAC.

Quanto menor o ID, melhor. Ambos os switches têm a mesma prioridade, mas o endereço MAC do SW2 é mais baixo, o que significa que SW2 será o escolhido. SW3 terá que bloquear uma das suas portas, o que efetivamente bloqueará o loop! Dê uma olhada no exemplo:

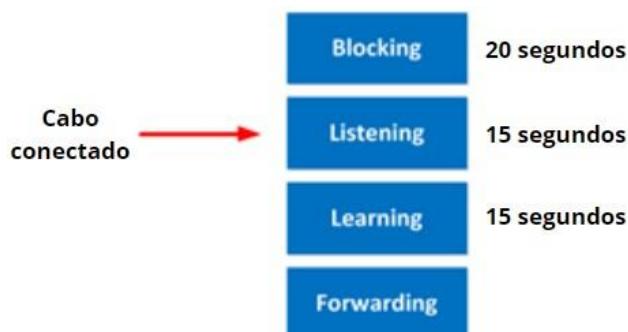


Na imagem acima, a interface Fa1/0 do SW3 vem seguida da letra “A”, que significa ‘alternativo’ (alternativo). Uma porta no estado ‘alternativo’ está bloqueada! Às vezes, a porta ‘alternativo’ é chamada de porta **ND (No Designated)**.

Se já trabalhou com um switch Cisco, deve ter notado que sempre que conectamos um cabo, o led acima da interface fica laranja e depois de um tempo fica verde. O que está acontecendo é que o spanning-tree está determinando o status da interface. Abaixo, uma descrição mais detalhada sobre o que acontece quando conectamos um cabo a interface de rede do switch:

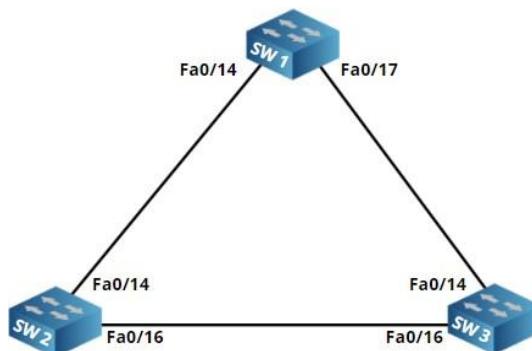
- A porta entra no modo de **escuta (listening mode)** por 15 segundos. Nesta fase ela receberá e enviará BPDUs, porém, sem aprender endereços MAC ou transmitir dados.
- A porta entrará no modo de **aprendizagem (learning mode)** por 15 segundos. Ainda estamos enviando e recebendo BPDUs, mas agora o switch também aprenderá os endereços MAC, mas, sem ainda não transmite dados.
- Agora vamos para o modo de **encaminhamento (forwarding mode)**, finalmente, o switch começa a transmitir os dados!

Um pequeno esquema para ajudar na compreensão:



Configuração do Spanning-Tree em switches Cisco

É chegado a hora de colocarmos a mão na massa. Vamos aprender como configuramos e manipulamos o spanning-tree. Para isso, vamos continuar utilizando a mesma topologia:



O Spanning tree vem habilitado por padrão, então podemos começar com uns comandos ‘show’ para verificarmos essa configuração ‘default’ antes de realizarmos qualquer configuração.

```
SW1#show spanning-tree
VLAN0001
Spanning tree enabled protocol ieee
Root ID      Priority      32769
Address      000f.34ca.1000
Cost         19
Port         19 (FastEthernet0/17)
Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
Bridge ID   Priority      32769  (priority 32768 sys-id-ext 1)
Address      0011.bb0b.3600
```

Hello Time	2 sec	Max Age	20 sec	Forward Delay	15 sec
Aging Time 300					
Interface	Role	Sts	Cost	Prio.Nbr	Type
-----	-----	-----	-----	-----	-----
Fa0/14	Desg	FWD	19	128.16	P2p
Fa0/17	Root	FWD	19	128.19	P2p

O comando ‘show spanning-tree’ é o comando ‘show’ mais importante. Vou dividir a interpretação da saída do comando para facilitar a explicação:

VLAN0001
Spanning tree enabled protocol ieee

Estamos examinando as informações de spanning tree para VLAN 1. Spanning-tree possui várias versões, e a versão padrão em switches Cisco é **PVST** (Per VLAN spanning-tree).

Root ID	Priority	32769
Address	000f.34ca.1000	
Cost	19	
Port	19 (FastEthernet0/17)	

Aqui temos as informações do root bridge. Essa parte nos mostra a prioridade, que é de 32769 e o endereço MAC: 000f.34ca.1000. Da perspectiva do SW1, ele possui o custo de 19 para alcançar o switch root bridge, e a porta que leva até a root bridge é chamada de root port, e no SW1 é a interface fa0/17.

Bridge ID Priority	32769 (priority 32768 sys-id-ext 1)
Address	0011.bb0b.3600

Esta parte é bem interessante, ela nos mostra as informações locais do switch, no caso, o SW1. Observe que existe mais de um número no campo priority:

- Priority 32769
- Priority 32768 sys-id-ext 1

O valor ‘sys-id-ext’ que você vê, é o número da VLAN. A prioridade é 32768, mas o spanning-tree adiciona o número da VLAN, então terminamos com o valor de prioridade 32769. Por último, temos o endereço MAC do SW1, que é 0011.bb0b.3600.

Hello Time 2 seg. Max Age 20 seg Forward Delay 15 seg

Aqui estão algumas informações sobre os diferentes tempos (timers) do Spanning Tree:

- Hello time: A cada 2 segundos um BPDU é enviado.
- Max Age: Se não recebemos BPDUs por 20 segundos, o switch conclui que algo mudou na rede, e verifica novamente a topologia.
- Forward Delay: Este temporizador é usado para os estados de escuta (listening) e aprendizagem (learning). Permanecemos em cada estado durante o ‘atraso de encaminhamento’ (forward delay), que por padrão, é 15 segundos.

Interface	Role	Sts	Cost	Prio.Nbr	Type
-----	-----	-----	-----	-----	-----

Fa0/14	Desg FWD 19	128.16	P2p
Fa0/17	Root FWD 19	128.19	P2p

A última parte do comando ‘show spanning-tree’, mostra as interfaces e seus status. Observe que temos duas interfaces no SW1:

- Fa0/14 é uma porta designada (designated port) e está no modo de encaminhamento (fowarding Mode (FWD)).
- Fa0/17 é uma porta raiz e está no modo de encaminhamento (fowarding Mode (FWD)).

O prio.nbr que aparece na saída, é a prioridade de porta que falamos anteriormente.

Como apenas os switches não raiz (non-root bridge) têm uma porta raiz (root-port), podemos concluir que SW1 não é o switch root, pois a interface fa0/17 está encaminhando o tráfego para o root bridge.

Vamos dar uma olhada no SW2:

SW2#show spanning-tree					
VLAN0001					
Spanning tree enabled protocol ieee					
Root ID Priority 32769					
Address 000f.34ca.1000					
Cost 19					
Port 18 (FastEthernet0/16)					
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec					
Bridge ID Priority 32769 (priority 32768 sys-id-ext 1)					
Address 0019.569d.5700					
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec					
Aging Time 300					
Interface Role Sts Cost Prio.Nbr Type					
<hr/>					
Fa0/14 Altn BLK 19 128.16 P2p					
Fa0/16 Root FWD 19 128.18 P2p					

Atenção para parte que destaco abaixo:

Root ID	Priority	32769
	Address	000f.34ca.1000
	Cost	19
	Port	18 (FastEthernet0/16)

Aqui temos informações sobre o root bridge. Essas informações são bem semelhantes às que vemos no SW1. A port root no SW2 é a Fa0/16.

Bridge ID	Priority	32769 (priority 32768 sys-id-ext 1)
Address	0019.569d.5700	

Algumas informações sobre o SW2. A prioridade é a mesma que o SW1, apenas o endereço MAC (0019.569d.5700) é diferente.

Interface	Role	Sts	Cost	Prio.Nbr	Type
<hr/>					
Fa0/14	Altn	BLK	19	128.16	P2p
Fa0/16	Root	FWD	19	128.18	P2p

Essa parte merece uma atenção especial, observe:

- A interface fa0/14 é uma porta alternativa (alternate port) e está no modo de bloqueio (blocking mode (BLK)).
- A interface fa0/16 é uma porta raiz e está no modo de encaminhamento (forwarding Mode (FWD)).

Vamos até o switch 03:

```
SW3#show spanning-tree

VLAN0001

Spanning tree enabled protocol ieee

Root ID      Priority      32769
Address      000f.34ca.1000

This bridge is the root

Hello Time    2 sec  Max Age 20 sec  Forward Delay 15 sec

Bridge ID  Priority      32769 (priority 32768 sys-id-ext 1)
Address      000f.34ca.1000

Hello Time    2 sec  Max Age 20 sec  Forward Delay 15 sec

Aging Time 300

Interface      Role Sts Cost      Prio.Nbr Type
-----      -----
Fa0/14        Desg FWD 19      128.14    P2p
Fa0/16        Desg FWD 19      128.16    P2p
```

Assim como nos switches anteriores, vamos quebrar essa saída do comando ‘show’ para um melhor entendimento:

```

Root ID      Priority      32769
Address       000f.34ca.1000
This bridge is the root

```

Conforme esperado, o SW3 é a root bridge da rede. Já sabíamos disso porque SW1 e SW2 são non-root, apesar disso, vamos verificar também o SW3.

```

Bridge ID      Priority      32769 (priority 32768 sys-id-ext 1)
Address       000f.34ca.1000

```

Ambas as interfaces do SW3 estão em FWD – fowarding mode.

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/14	Desg	FWD	19	128.14	P2p
Fa0/16	Desg	FWD	19	128.16	P2p

Vamos verificar porque o SW3 foi escolhido como root bridge:

```

SW1#show spanning-tree | begin Bridge ID
Bridge ID      Priority      32769 (priority 32768 sys-id-ext 1)
Address       0011.bb0b.3600

```

```

SW2#show spanning-tree | begin Bridge ID
Bridge ID      Priority      32769 (priority 32768 sys-id-ext 1)
Address       0019.569d.5700

```

```

SW3#show spanning-tree | begin Bridge ID
Bridge ID      Priority      32769 (priority 32768 sys-id-ext 1)
Address       000f.34ca.1000

```

A prioridade é a mesma em todos os switches (32768), portanto, temos que olhar para os endereços MAC:

- SW1: 0011.bb0b.3600
- SW2: 0019.569d.5700
- SW3: 000f.34ca.1000

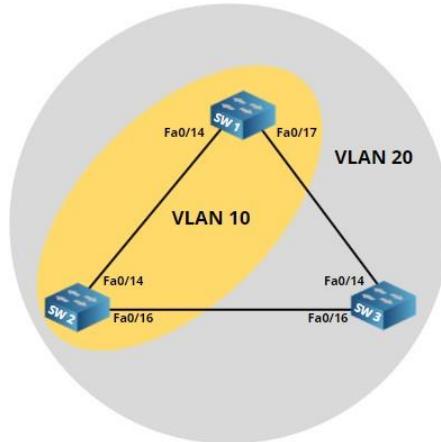
SW3 tem o endereço MAC mais baixo, por isso se tornou a root bridge. Por que a interface fa0/14 do SW2 foi bloqueada e não a interface fa0/14 do SW1? Mais uma vez, temos que olhar para o identificador da ponte (bridge identifier). A prioridade é 32768 em ambos os switches, portanto, temos que comparar o endereço MAC:

- SW1: 0011.bb0b.3600
- SW2: 0019.569d.5700

SW1 tem o endereço MAC inferior e, portanto, um ‘bridge identifier’ melhor. É por isso que o SW2 perdeu a batalha e teve que desabilitar a interface fa0/14.

Per VLAN Spanning Tree (PVST)

Vamos começar com esta topologia abaixo:



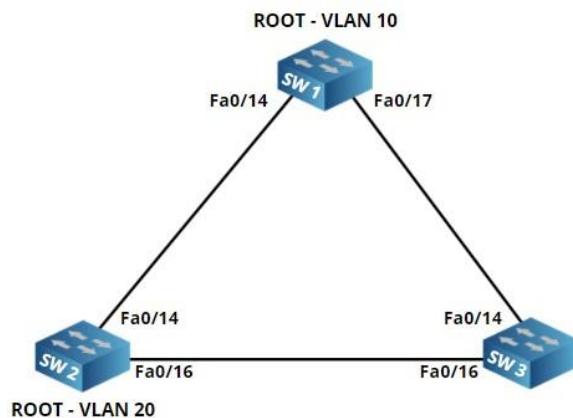
- VLAN 10 está configurada no SW1 e SW2.
- VLAN 20 está configurada no SW1, SW2 e SW3.

Pense comigo, há loop na VLAN 10? E na VLAN 20?

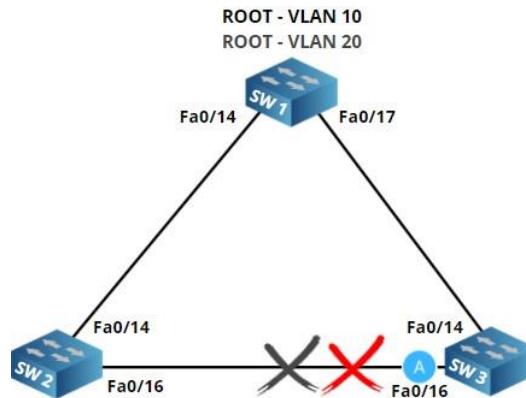
Há uma grande diferença entre a topologia **física** e **lógica**. Não há loop na VLAN 10, pois ela só está no link entre SW1 e SW2. No entanto, haverá loop na VLAN 20, pois ela está configurada em todos os switches.

A versão mais antiga do spanning-tree, **CST (Common Spanning-Tree - padrão 802.1D)**, trabalhava com apenas uma instância, ou seja, apenas um processo do spanning-tree para toda a rede, independentemente do número de vlans. Atualmente, os switches Cisco trabalham com a **PVST (Per VLAN Spanning-Tree)**, essa versão é capaz de calcular uma topologia para **cada VLAN**. A partir da versão 15.2 (4) E do IOS, o modo padrão do STP passou a ser o Rapid PVST +

No exemplo abaixo, temos três switches e duas VLANs. As duas Vlans estão configuradas nos três switches:

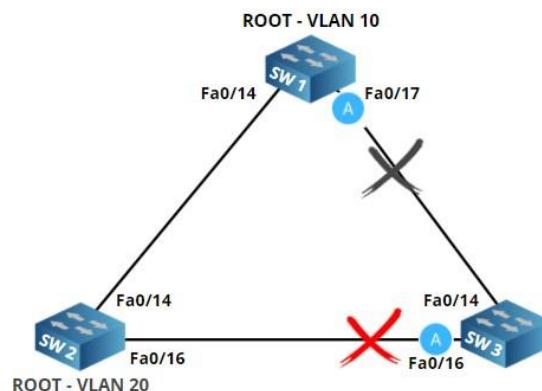


Observe que temos duas roots bridges. Com a utilização do PVST, é possível criar uma root bridge diferente para cada VLAN. SW1 poderia ser a root bridge para VLAN 10 e SW2 pode ser a root bridge para VLAN 20. Mas qual a vantagem de fazer isso? Acredite, não é só para complicar, observe o exemplo abaixo:



Se elegermos apenas um switch como root bridge para ambas as VLANs, uma interface será totalmente bloqueada, consequentemente não passando tráfego de nenhuma VLAN. No exemplo acima, SW1 é a root bridge para VLAN 10 e 20 e, como resultado, a interface fa0/16 no SW3 está bloqueada para ambas as VLANs, ou seja, nenhum tráfego será encaminhado pela interface fa0/16. Imagine se essa fosse uma interface de 10 Gigabit. Seria um desperdício imenso ter uma interface desse porte totalmente parada!

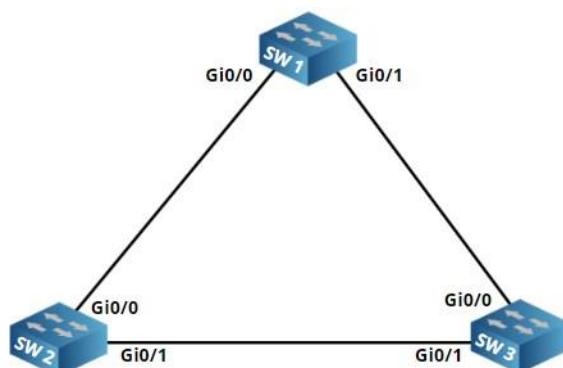
Quando colocamos outro switch como a root bridge para VLAN 20, teremos o seguinte resultado:



Observe que o SW2 se tornou a root bridge para VLAN 20. Como resultado, a interface fa0/16 no SW2 está bloqueada para VLAN 10, enquanto a interface fa0/17 no SW1 está bloqueada para VLAN 20. A vantagem de ter várias roots bridge é que podemos fazer balanceamento de carga.

Spanning-tree – Configuração do Root Bridge

Hora de aprendermos as diferentes opções de configuração do root bridge. Eis a topologia que usaremos:



Temos três switches e em cada switch configuramos três vlans:

SW1, SW2 & SW3

```
(config)#vlan 10  
(config)#vlan 20  
(config)#vlan 30
```

Vamos configurar as interfaces que estão conectando um switch ao outro como interface trunk:

```
SW1, SW2 & SW3  
  
(config)#interface range GigabitEthernet 0/0 - 1  
(config-if-range)#switchport trunk encapsulation dot1q  
(config-if-range)#switchport mode trunk
```

Tudo configurado, é hora de verificarmos a bridge ID dos switches:

```
SW1#show spanning-tree bridge detail  
  
VLAN0001  
  
    Bridge ID  Priority      32769  (priority 32768 sys-id-ext 1)  
              Address      5254.001a.935a  
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec  
  
VLAN0010  
  
    Bridge ID  Priority      32778  (priority 32768 sys-id-ext 10)  
              Address      5254.001a.935a  
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec  
  
VLAN0020  
  
    Bridge ID  Priority      32788  (priority 32768 sys-id-ext 20)  
              Address      5254.001a.935a  
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec  
  
VLAN0030  
  
    Bridge ID  Priority      32798  (priority 32768 sys-id-ext 30)  
              Address      5254.001a.935a  
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
```

```
SW2#show spanning-tree bridge detail  
  
VLAN0001  
  
    Bridge ID  Priority      32769  (priority 32768 sys-id-ext 1)  
              Address      5254.0015.bc74  
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
```

```

VLAN0010

  Bridge ID  Priority      32778  (priority 32768 sys-id-ext 10)
                Address      5254.0015.bc74
                Hello Time    2 sec  Max Age 20 sec  Forward Delay 15 sec

VLAN0020

  Bridge ID  Priority      32788  (priority 32768 sys-id-ext 20)
                Address      5254.0015.bc74
                Hello Time    2 sec  Max Age 20 sec  Forward Delay 15 sec

VLAN0030

  Bridge ID  Priority      32798  (priority 32768 sys-id-ext 30)
                Address      5254.0015.bc74
                Hello Time    2 sec  Max Age 20 sec  Forward Delay 15 sec

```

```

SW3#show spanning-tree bridge detail

VLAN001

  Bridge ID  Priority      32769  (priority 32768 sys-id-ext 1)
                Address      5254.001d.e6bb
                Hello Time    2 sec  Max Age 20 sec  Forward Delay 15 sec

VLAN0010

  Bridge ID  Priority      32778  (priority 32768 sys-id-ext 10)
                Address      5254.001d.e6bb
                Hello Time    2 sec  Max Age 20 sec  Forward Delay 15 sec

VLAN0020

  Bridge ID  Priority      32788  (priority 32768 sys-id-ext 20)
                Address      5254.001d.e6bb
                Hello Time    2 sec  Max Age 20 sec  Forward Delay 15 sec

VLAN0030

  Bridge ID  Priority      32798  (priority 32768 sys-id-ext 30)
                Address      5254.001d.e6bb
                Hello Time    2 sec  Max Age 20 sec  Forward Delay 15 sec

```

Lembre-se, a prioridade padrão em cada switch é de 32768, portanto, sem nenhuma configuração extra o endereço MAC será o número responsável por desempatar a escolha do root bridge. No caso acima, o SW2 foi escolhido como root bridge para todas as vlans:

```

SW2#show spanning-tree vlan 10

VLAN0010

    Spanning tree enabled protocol ieee

    Root ID      Priority      32778
                  Address       5254.0015.bc74

    This bridge is the root

    Hello Time    2 sec  Max Age 20 sec  Forward Delay 15 sec

    Bridge ID    Priority      32778  (priority 32768 sys-id-ext 10)
                  Address       5254.0015.bc74

    Hello Time    2 sec  Max Age 20 sec  Forward Delay 15 sec
    Aging Time   15  sec

Interface        Role Sts Cost      Prio.Nbr Type
-----
Gi0/0            Desg FWD 4       128.1    P2p
Gi0/1            Desg FWD 4       128.2    P2p

SW2#show spanning-tree vlan 20

```

```

VLAN0020

    Spanning tree enabled protocol ieee

    Root ID      Priority      32788
                  Address       5254.0015.bc74

    This bridge is the root

    Hello Time    2 sec  Max Age 20 sec  Forward Delay 15 sec

    Bridge ID    Priority      32788  (priority 32768 sys-id-ext 20)
                  Address       5254.0015.bc74

    Hello Time    2 sec  Max Age 20 sec  Forward Delay 15 sec
    Aging Time   300  sec

Interface        Role Sts Cost      Prio.Nbr Type
-----
Gi0/0            Desg FWD 4       128.1    P2p
Gi0/1            Desg FWD 4       128.2    P2p

```

```
SW2#show spanning-tree vlan 30
```

VLAN0030

Spanning tree enabled protocol ieee

Root ID Priority 32798

Address 5254.0015.bc74

This bridge is the root

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32798 (priority 32768 sys-id-ext 30)

Address 5254.0015.bc74

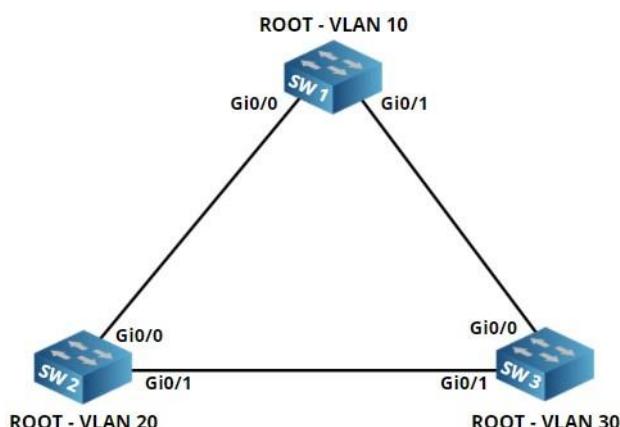
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Aging Time 300 sec

Interface	Role	Sts	Cost	Prio.Nbr	Type
-----------	------	-----	------	----------	------

Gi0/0	Desg	FWD	4	128.1	P2p
Gi0/1	Desg	FWD	4	128.2	P2p

Vamos trabalhar na configuração para que ela fique desse jeito:



É possível mudar a configuração do root bridge para cada vlan com o comando: ‘spanning-tree’. Observe abaixo:

```
SW1(config)#spanning-tree vlan 10 ?  
forward-time Set the forward delay for the spanning tree  
hello-time Set the hello interval for the spanning tree  
max-age Set the max age interval for the spanning tree  
priority Set the bridge priority for the spanning tree  
root Configure switch as root
```

Nós temos duas opções:

- Priority: Nessa opção, podemos alterar manualmente a prioridade da bridge.
- Root: Permite configurar o switch como root.

Vamos descobrir a diferença entre esses dois parâmetros:

Root Parameter

Vamos verificar as opções do parâmetro ‘root’:

```
SW1(config)#spanning-tree vlan 10 root ?  
primary    Configure this switch as primary root for this spanning tree  
secondary   Configure switch as secondary root
```

Podemos configurar o switch para se tornar o root bridge primário ou secundário:

```
SW1(config)#spanning-tree vlan 10 root primary
```

Vamos verificar:

```
SW1#show spanning-tree vlan 10  
  
VLAN0010  
  
    Spanning tree enabled protocol ieee  
  
    Root ID      Priority     24586  
                  Address      5254.001a.935a  
  
                  This bridge is the root  
  
                  Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec  
  
    Bridge ID    Priority      24586  (priority 24576 sys-id-ext 10)  
                  Address      5254.001a.935a  
  
                  Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec  
  
                  Aging Time  15  sec  
  
    Interface      Role Sts Cost      Prio.Nbr Type  
-----  
Gi0/0           Desg FWD 4        128.1      P2p  
Gi0/1           Desg FWD 4        128.2      P2p
```

Observe que o SW1 se tornou o root bridge para a VLAN 10 e sua prioridade agora é de 24586. Vamos configurar o SW2 para ser o switch root bridge secundário:

```
SW2(config)#spanning-tree vlan 10 root secondary
```

Vamos verificar o resultado do comando:

```
SW2#show spanning-tree vlan 10  
  
VLAN0010
```

```

Spanning tree enabled protocol ieee

Root ID    Priority    24586
            Address      5254.001a.935a
            Cost          4
            Port         1 (GigabitEthernet0/0)

Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

Bridge ID  Priority    28682  (priority 28672 sys-id-ext 10)
            Address      5254.0015.bc74
            Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
            Aging Time   300 sec

Interface        Role Sts Cost      Prio.Nbr Type
-----  -----
Gi0/0           Root FWD 4       128.1    P2p
Gi0/1           Desg FWD 4      128.2    P2p

```

A prioridade do SW2 passou a ser de 28682, mas repare bem, em nenhum lugar no output do comando diz que ele é o ‘secondary root bridge’. Na verdade, quando usamos os parâmetros ‘root primary’ ou ‘root secondary’, o IOS define o novo número de prioridade. Observe como ficou a configuração:

```

SW1#show running-config | include priority
spanning-tree vlan 10 priority 24576

```

```

SW2#show running-config | include priority
spanning-tree vlan 10 priority 28672

```

O switch verifica a prioridade do root bridge atual e em seguida diminui sua própria prioridade para que ele se torne a nova root bridge.

Priority Parameter

Podemos configurar o número da prioridade manualmente:

```

SW2(config)#spanning-tree vlan 20 priority ?
<0-61440>  bridge priority in increments of 4096

```

Vamos configurar a Vlan 20 no SW2:

```

SW2(config)#spanning-tree vlan 20 priority 0

```

Configuramos a prioridade como 0, dessa forma, temos a menor possibilidade possível. Vamos verificar se o SW2 se tornou a root bridge para a Vlan 20:

```

SW2#show spanning-tree vlan 20

```

```
VLAN0020
```

```
Spanning tree enabled protocol ieee

Root ID  Priority  20
          Address  5254.0015.bc74
          This bridge is the root

          Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

Bridge ID  Priority  20      (priority 0 sys-id-ext 20)
          Address  5254.0015.bc74
          Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
          Aging Time  300 sec

Interface        Role Sts Cost      Prio.Nbr Type
-----  -----
Gi0/0            Desg FWD 4       128.1    P2p
Gi0/1            Desg FWD 4       128.2    P2p
```

A prioridade agora é 20 (prioridade 0 e sys-id-ext 20). Vamos configurar o SW3 para se tornar a root bridge para VLAN 30:

```
SW3(config)#spanning-tree vlan 30 priority 0
```

Vamos conferir:

```
SW3#show spanning-tree vlan 30

VLAN0030

Spanning tree enabled protocol ieee

Root ID  Priority  30
          Address  5254.001d.e6bb
          This bridge is the root

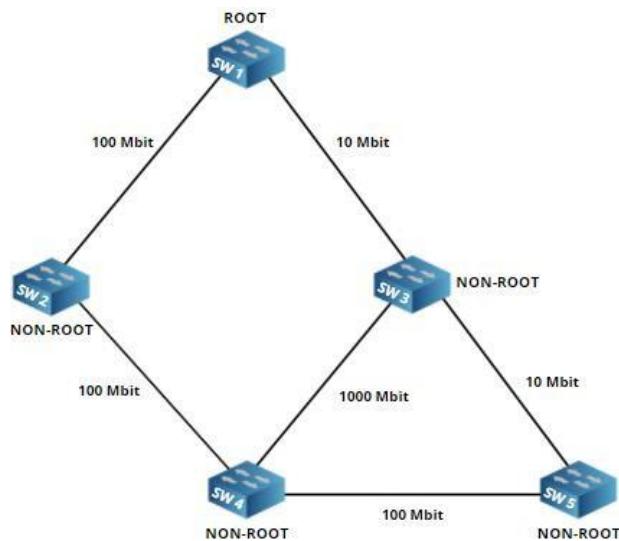
          Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

Bridge ID  Priority  30      (priority 0 sys-id-ext 30)
          Address  5254.001d.e6bb
          Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
          Aging Time  300 sec

Interface        Role Sts Cost      Prio.Nbr Type
-----  -----
Gi0/0            Desg FWD 4       128.1    P2p
```

Calculo de custo do Spanning Tree

Non-root bridges precisam encontrar o caminho mais curto para a root bridge. Mas como o switch define qual caminho será usado quando há velocidades diferentes como Ethernet, FastEthernet e Gigabit? Observe a topologia abaixo, pois usaremos ela para explicar o cálculo de custo do spanning tree.



Na imagem acima, temos uma rede com vários switches conectados entre si com velocidades variáveis, como Ethernet (10 Mbit), FastEthernet (100Mbit) e Gigabit (1000Mbit).

O SW1 é a root bridge, portanto, todos os outros switches são ‘não raiz’ (Non-root bridges) e precisam encontrar o caminho mais curto para o root bridge. Para cada velocidade de link é atribuído um custo:

Largura de banda	Custo
10 Mbit	100
100 Mbit	19
1000 Mbit	4

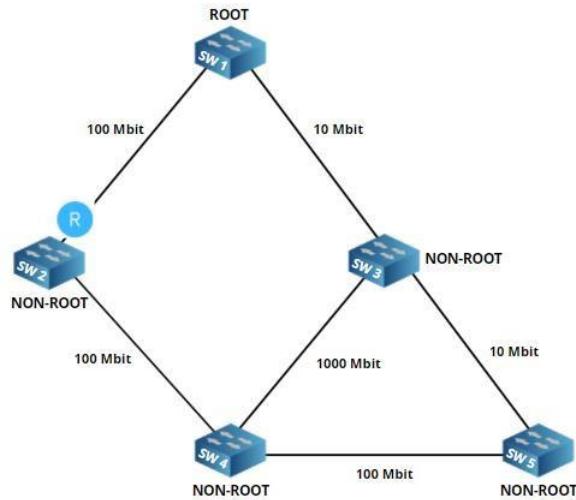
Quanto mais lenta for a interface, maior será o custo. O caminho com menor custo será usado para chegar a root bridge. Observe o BPDU abaixo:

Protocol Identifier	Protocol Version Identifier	BPDU Type	Flags	Root Identifier	Root Path Cost	Bridge Identifier	Port Identifier	Message Age	Max Age	Hello Time	Forward Delay
---------------------	-----------------------------	-----------	-------	-----------------	----------------	-------------------	-----------------	-------------	---------	------------	---------------

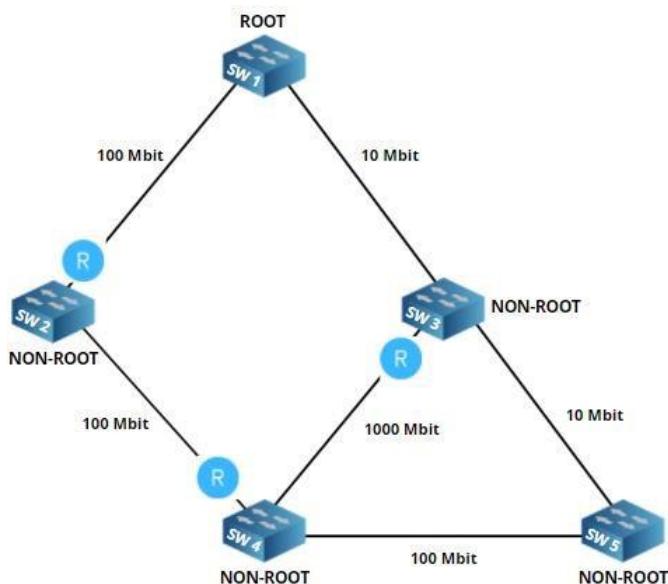
BPDU

No BPDU, há um campo chamado ‘root path cost’ (algo como *custo do caminho para a raiz*). É nesse espaço que o switch informa o custo do caminho mais curto para a root bridge. Assim que os switches descobrem qual switch está declarado como root bridge, eles procurarão o caminho mais curto para chegar até ele. BPDUs são encaminhados da root bridge para todos os switches.

Abaixo um exemplo dos diferentes custos do spanning tree na topologia que estamos usando:



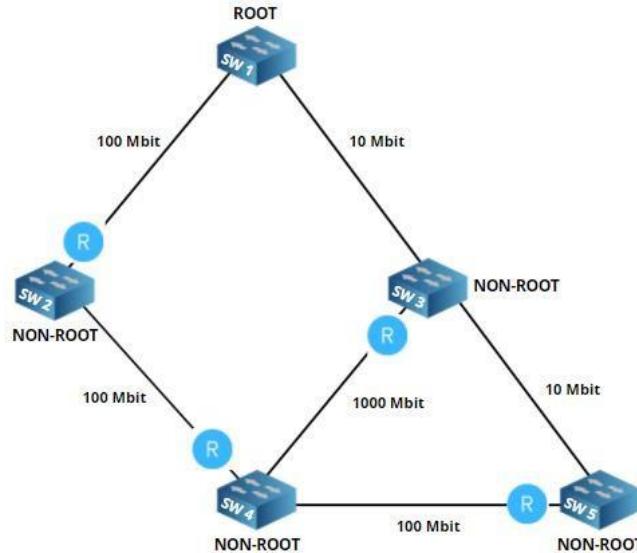
O SW2 usará o link para SW1 como sua root port, pois esta é uma interface de 100 Mbit, e como vimos anteriormente tem um custo de 19. Ele encaminhará BPDUs para SW4; no campo ‘root path cost’ do BPDU haverá um custo de 19. SW3 também está recebendo BPDUs de SW1, então é possível que neste momento ele selecione a interface de 10 Mbit como a root port. Vamos continuar:



A figura acima necessita de explicações mais detalhadas:

- SW3 recebe BPDUs em sua interface de 10 Mbit (custo 100) e na interface de 1000 Mbit (custo 4). Ele usará a interface de 1000 Mbit como sua root port (o caminho mais curto para a root bridge é $19 + 19 + 4 = 42$).
- SW3 encaminhará BPDUs para o SW4. O campo ‘root path cost’ será 100.
- SW4 recebe um BPDU de SW2 com ‘root path cost’ de 19.
- SW4 recebe um BPDU de SW3 com ‘root path cost’ de 100.
- O caminho através de SW2 é mais curto, então está se tornará a root port para o SW4.
- SW4 encaminhará BPDUs para SW3 e SW5. No campo ‘root path cost’ do BPDU encontraremos um custo de 38 (seu ‘root path cost’ de 19 + seu custo de interface própria de 19).
- SW3 encaminhará BPDUs para SW5, inserindo um custo de 42 no ‘root path cost’ ($19 + 19 + 4$).

A imagem completa ficará assim:



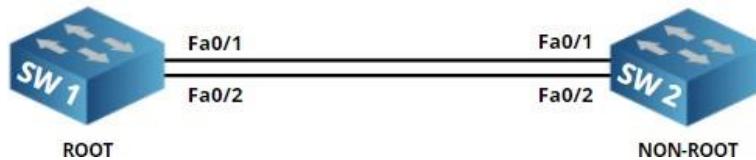
SW5 recebe BPDUs do SW3 e SW4. No BPDU se examinarmos o campo ‘root path cost’, veremos as seguintes informações:

- BPDU do SW3: custo 42
- BPDU do SW4: custo 38

SW5 irá adicionar o custo de sua própria interface para SW4, de modo que o custo total para alcançar a root bridge através do SW4 será $38 + 19$ (custo de interface de 100 Mbit) = 57. O custo total para alcançar a root bridge através do SW3 é $42 + 100$ (Interface de 10 Mbit) = 142. Como resultado, ele selecionará a interface para SW4 como sua root port.

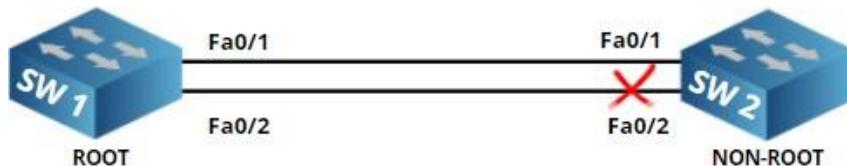
Lembre-se de que os switches tomam decisões apenas sobre os BPDUs que recebem! Eles não têm ideia de como é a topologia. A única coisa que eles sabem é em qual interface receberam o melhor BPDU. **Lembrando:** O melhor BPDU é aquele com o caminho mais curto para a root bridge!

E se o custo for igual?



Observe a topologia acima. SW1 é a root bridge e SW2 é o non-root. Temos dois links entre esses switches para que haja redundância, mas lembre-se, redundância não controlada significa loops, então o spanning tree terá que bloquear uma das interfaces do SW2.

SW2 receberá BPDUs em ambas as interfaces e o campo ‘root path cost’ será o mesmo! Qual interface então será bloqueada? Fa0/1 ou fa0/2? Quando o custo é igual, o spanning tree examinará a **prioridade da porta**. Por padrão, a prioridade da porta é a mesma para todas as interfaces, o que significa que o **número da interface** será o fator de desempate.



O número de interface mais baixo, será o escolhido. Logo, a interface fa0/2 será bloqueada. A prioridade da porta é um valor que podemos alterar para que possamos escolher manualmente qual interface será bloqueada!

O spanning tree toma decisões de acordo com a lista abaixo:

1. **Lowest bridge ID**: O switch com o bridge ID mais baixo torna-se o root bridge.
2. **Lowest path cost to root bridge**: Quando o switch recebe vários BPUDUs, ele escolhe como root port a interface que tem o custo mais baixo para a root bridge.
3. **Lowest sender bridge ID**: Quando um switch é conectado a dois switches que levam até a root bridge e o custo para chegar até a root bridge é o mesmo, o switch seleciona a interface que se conecta ao switch com o bridge ID mais baixa como o root port.
4. **Lowest sender port ID**: Quando o switch tem duas interfaces conectadas ao mesmo switch e o custo para alcançar a bridge root é o mesmo, ele usará a interface com o número mais baixo como a root port.

Rapid Spanning-Tree (RSTP)

Rapid Spanning-Tree não é uma revolução do Spanning-tree original, mas uma evolução. Nos bastidores, algumas coisas foram alteradas para acelerar o processo, em termos de configuração é igual ao que vimos até agora.

Vamos mergulhar no Rapid Spanning-Tree e veremos quais são as diferenças com Spanning-Tree clássico. Primeiro, dê uma olhada na imagem abaixo:



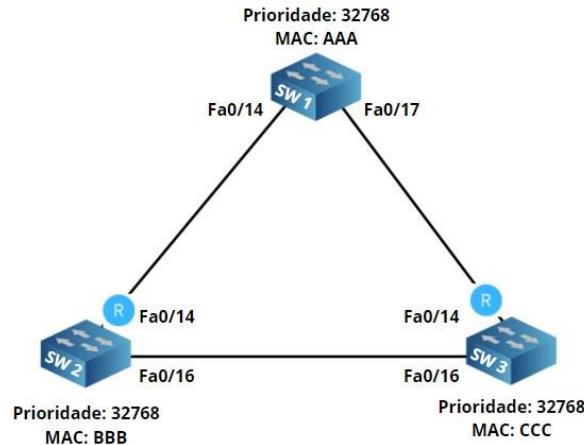
Lembra-se dos estados da porta do spanning tree? Temos um estado de porta de bloqueio, escuta, aprendizagem e encaminhamento. Esta é a primeira diferença entre o spanning-tree e o rapid spanning-tree. O rapid spanning-tree tem apenas três estados de porta:

- Descartando
- Aprendendo
- Encaminhamento

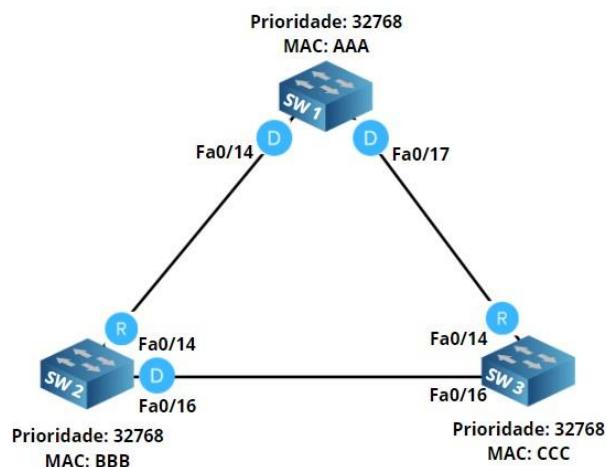
Já estudamos sobre sobre aprendizagem e encaminhamento, mas **descartar** é um novo conceito. Bem, basicamente, ele combina o estado de bloqueio e escuta. O quadro abaixo apresente uma boa visão geral:

Spanning-Tree Tradicional	Rapid Spanning-Tree	Porta está ativa?	Aprendendo endereços MAC?
Blocking	Discarding	Não	Não
Listening	Discarding	Sim	Não
Learning	Learning	Sim	Sim
Forwarding	Forwarding	Sim	Sim

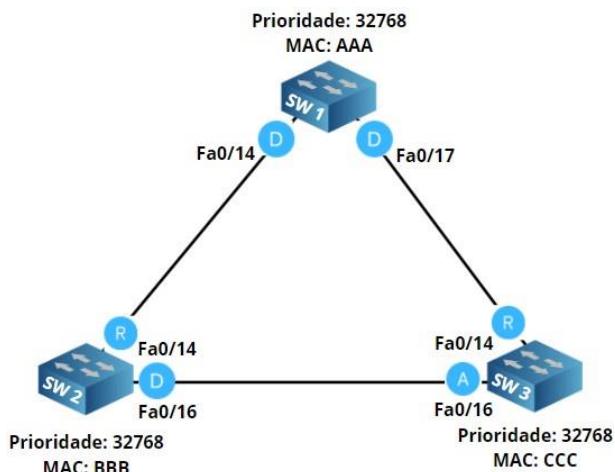
Vamos fazer uma pequena revisão das funções de porta que o Spanning-Tree tradicional utiliza e correlaciona-las com o rapid spanning-tree:



O switch com a melhor bridge ID (prioridade + endereço MAC) torna-se o root bridge. Os outros switches (não raiz) precisam encontrar o caminho de custo mais curto para a root bridge, descobrindo assim sua root port. Até aqui, não há nenhuma novidade, esse é o mesmo funcionamento do rapid spanning tree. A próxima etapa é selecionar as designated ports:

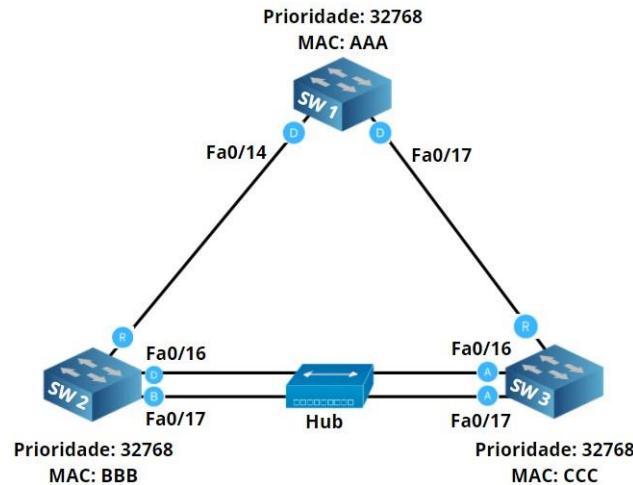


Em cada segmento, pode haver apenas uma porta designada ou teremos um loop. A porta se tornará a porta designada se puder enviar o melhor BPDU. SW1 como a ponte raiz sempre terá as melhores portas, portanto, todas as interfaces serão designadas. A interface fa0/16 do SW2 será a porta designada porque tem um bridge ID melhor que o SW3. Até aqui, nenhuma novidade em comparação com o spanning tree tradicional. As interfaces restantes serão bloqueadas:



O SW3 colocará a interface fa0/16 que conecta ao SW2 como porta alternativa! O rapid spanning tree funciona da mesma forma.

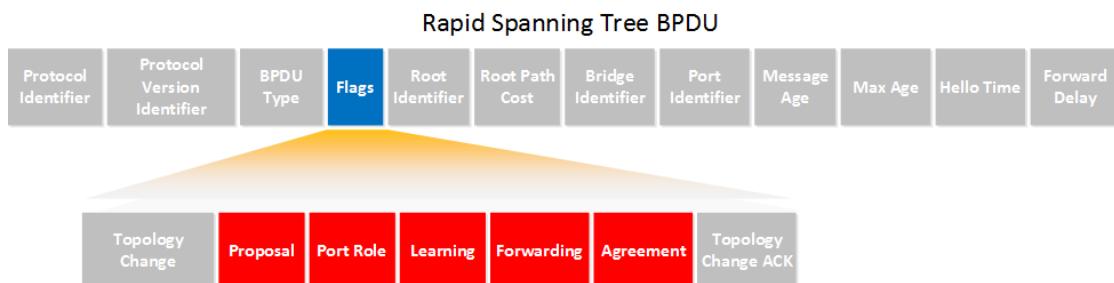
Vamos ver um exemplo com um estado de porta introduzido pelo rapid spanning tree:



Agora, nós temos um novo estado de porta, observe a interface fa0/17 do SW2. É chamada de **porta de backup** e faz parte das mudanças introduzidas pelo rapid spanning tree. É muito improvável que você veja essa porta em uma rede de produção. Entre SW2 e SW3, adicionamos um hub. Normalmente (sem o hub no meio), fa0/16 e fa0/17 seriam portas designadas.

Por causa do hub, as interfaces fa0/16 e fa0/17 do SW2 agora estão no **mesmo domínio de colisão**. Fa0/16 será eleita como a porta designada e fa0/17 se tornará a **porta de backup** para a interface fa0/16. A razão pela qual SW2 vê a interface fa0/17 como uma porta backup é porque ela recebe seus próprios BPDUs nas interfaces fa0/16 e fa0/17, e entende que tem duas conexões com o mesmo segmento. Se retirarmos o hub, as interfaces fa0/16 e fa0/17 se tornaram portas designadas como no spanning tree tradicional.

Outro campo diferente é o BPDU, dê uma olhada:



No Spanning tree tradicional, o campo ‘flags’ tinha apenas dois bits em uso:

- Mudança de topologia (Topology change)
- Reconhecimento de alteração de topologia (Topology Change ACK)

Com o rapid spanning tree, todos os bits do campo ‘flags’ foram utilizados. O campo ‘Port Role’ advindo da porta que originou o BPDU utilizará as seguintes opções:

- Unknown (Desconhecido)
- Porta alternativa/backup.
- Root Port
- Designated Port.

Este novo BPDU é chamado de **BPDU versão 2**. Os switches que estiverem executando a versão antiga do spanning-tree eliminarão esta nova versão do BPDU. Porém, o rapid spanning tree e o spanning tree tradicional são **compatíveis**, pois o Rapid Spanning Tree consegue lidar com switches que executam a versão mais antiga do Spanning Tree.

Vamos examinar outras alterações que o rapid spanning tree introduziu:

BPDUs agora são enviados em todos os ‘hello timers’. No spanning tree clássico, apenas a root bridge gera os BPDUs, já o Rapid Spanning Tree funciona de maneira diferente. Agora, todos os switches geram BPDUs a **cada dois segundos (hello time)**. Dois segundos é o tempo padrão do ‘hello’, porém, esse tempo pode ser alterado.

O spanning tree clássico utiliza um tempo de espera máximo de 20 segundos antes de descartar o BPDU, já o rapid spanning tree funciona de maneira um pouco diferente! BPDUs agora são usados como ferramentas do KeepAlive. Se um switch deixar de receber três BPDUs de um switch vizinho, ele assumirá que a conectividade com esse switch foi perdida, e removerá todos os endereços MAC imediatamente.

A velocidade de transição (tempo de convergência) é a característica mais importante do rapid spanning tree. No spanning tree tradicional, uma interface tem que passar pelos estados de escuta e aprendizagem antes de entrar no estado de encaminhamento, o que leva 30 segundos.

O rapid spanning tree não utiliza temporizadores para decidir se uma interface pode passar para o estado de encaminhamento ou não. Ele utiliza um mecanismo de negociação. Voltaremos nesse mecanismo em breve.

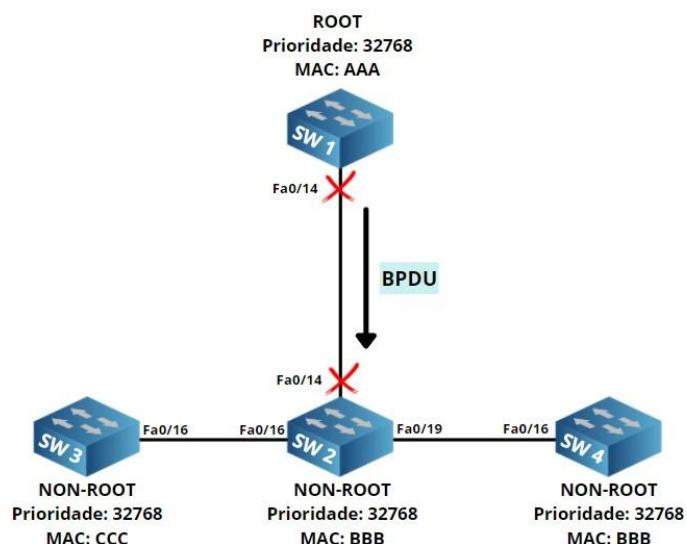
No Spanning tree clássico temos o portfast, quando ele está habilitado o switch ignora o estado de escuta e aprendizagem e coloca imediatamente a interface no estado de encaminhamento, essa interface não gerará alterações de topologia quando a ficar up ou down. O rapid spanning tree tem um recurso semelhante, mas com um novo nome: Edge port (porta de borda).

O rapid spanning tree só coloca interfaces no estado de encaminhamento, no estilo da portfast, em portas de borda (Edge port) ou interfaces ponto a ponto. Existem dois tipos de link:

1. Point-to-point - Ponto a ponto (full duplex)
2. Shared - Compartilhado (half duplex)

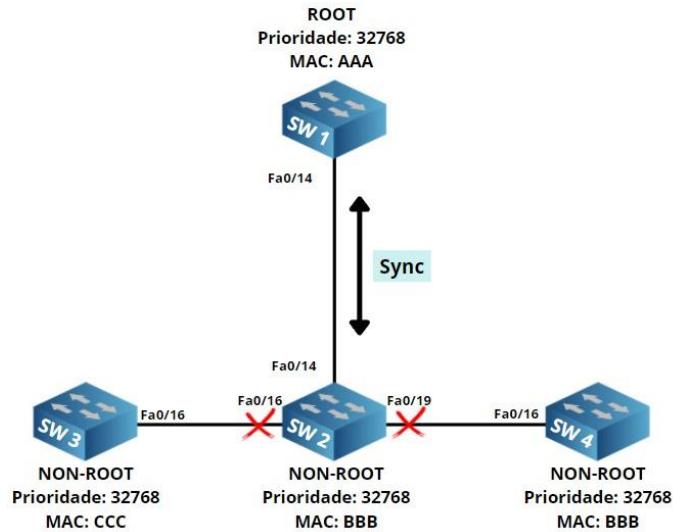
Normalmente, conectamos switches em outros switches, por isso todas as nossas interfaces são configuradas como full duplex e o rapid spanning tree enxerga essas interfaces como ponto a ponto. Se introduzirmos um hub na rede, teremos half duplex, que é visto como uma interface compartilhada pelo rapid spanning tree.

Vamos analisar maismeticulosamente o mecanismo de negociação que comentamos anteriormente:

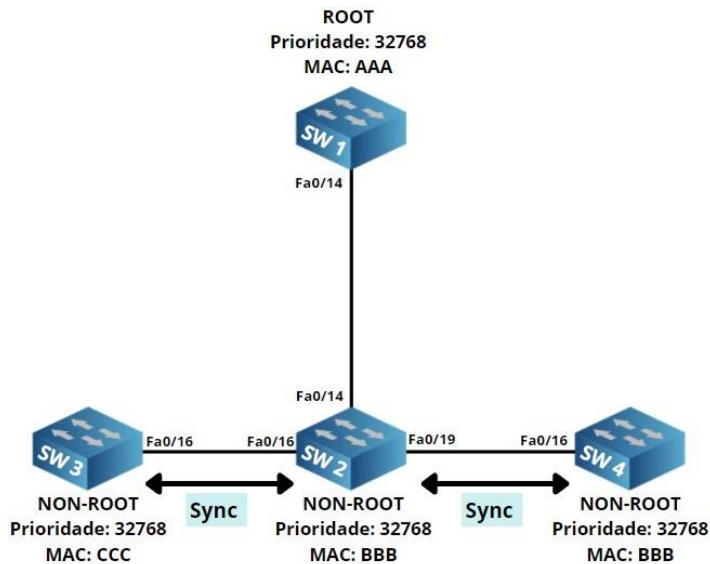


A partir dessa imagem, vou explicar o mecanismo de funcionamento do rapid spanning tree. O SW1 é o root bridge da rede, e SW2, SW3 e SW4 são non-root bridges.

Assim que o link entre SW1 e SW2 fica UP, as interfaces entram em modo de bloqueio. SW2 receberá um BPDU de SW1 e começará uma **negociação** chamada de **sincronização (sync)**:



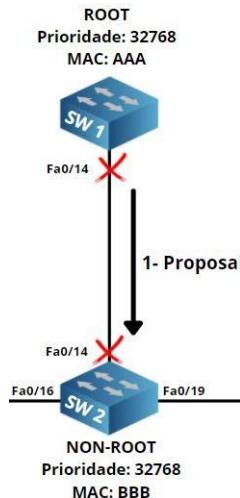
Assim que o SW2 recebe o BPDU da root bridge, ele bloqueia todas as suas interfaces que não forem edge ports. As portas que não são edge ports, são as interfaces que se conectam a outros switches, enquanto as edge ports (portas de borda) são as interfaces que têm o portfast configurado. Assim que SW2 bloquear suas portas que conectam a outros switches, o link entre SW1 e SW2 entrará no estado de encaminhamento. SW2 agora fará o seguinte:



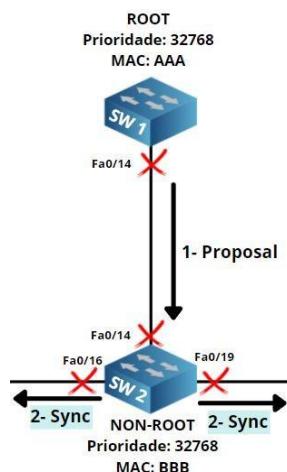
SW2 executará uma operação de sincronização (sync) com SW3 e SW4, fazendo assim que eles passem rapidamente para o estado de encaminhamento.

Não desista de aprender rapid spanning tree, o tópico é extenso, mas estamos terminando. Podemos concluir até agora que: O rapid spanning tree usa esse mecanismo de sincronização em vez do mecanismo “baseado em timer (marcação de tempo)”, o tradicional: Listening > Learning > fowarding.

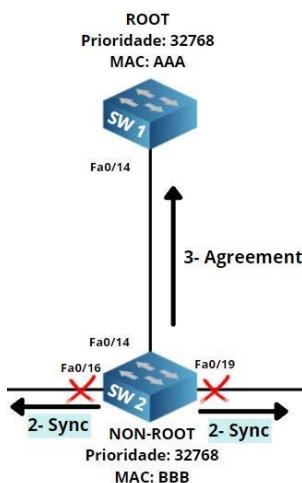
Vamos dar uma olhada no mecanismo de sincronização, um passo a passo do que está acontecendo entre SW1 e SW2:



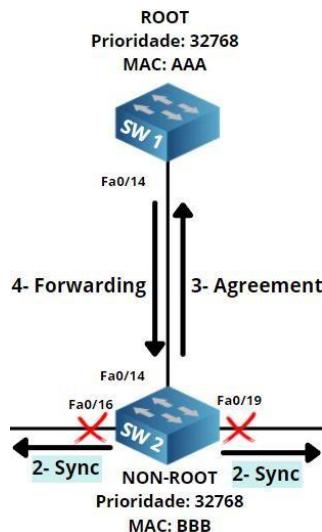
No início, as interfaces serão bloqueadas até que troquem BPDUs entre si. Neste momento, SW2 descobrirá que SW1 é a root bridge. O mecanismo de sincronização será iniciado com o SW1 definindo o ‘proposal bit’ no campo ‘flags’ do BPDU. Quando SW2 receber a ‘proposta’ ele seguirá conforme imagem abaixo:



SW2 irá bloquear todas as suas interfaces não periféricas e iniciará a sincronização com SW3 e SW4, assim que esse processo começar o SW2 informará ao SW1:

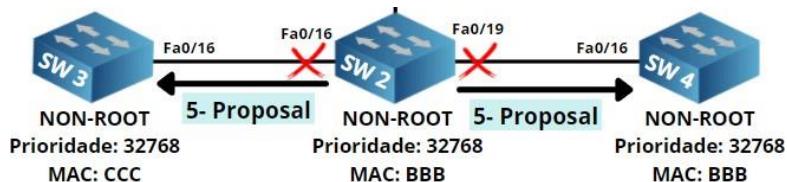


Tão logo o SW2 esteja com suas interfaces em modo de sincronização, ele informará ao SW1 enviando um “agreement”. Este “agreement” é uma cópia do ‘proposal BPDU’, onde o ‘proposal bit’ está desativado e o ‘agreement bit’ está ativado. A interface fa0/14 do SW2 entrará no modo de encaminhamento. Quando o SW1 receber o “agreement”, acontecerá o seguinte:

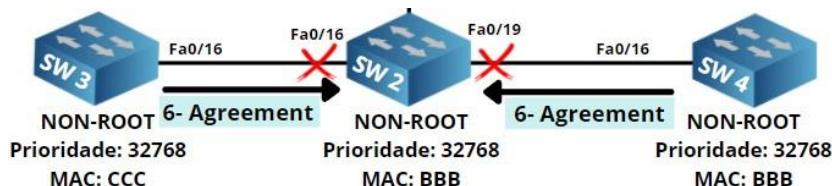


Assim que SW1 receber o “agreement” do SW2, ele colocará imediatamente a interface fa0/14 em modo de encaminhamento.

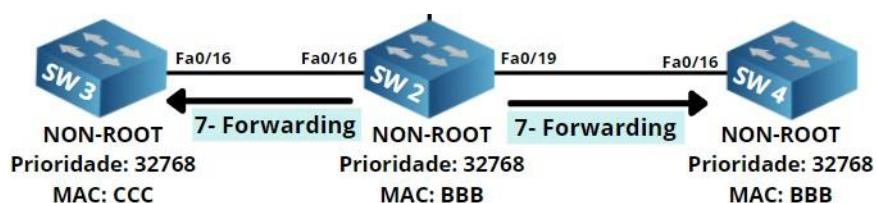
E quanto à interface fa0/16 e fa0/19 do SW2?



O mesmo mecanismo de sincronização ocorrerá agora nessas interfaces. SW2 enviará uma ‘proposal’ nas interfaces fa0/16 e fa0/19 para SW3 e SW4. SW3 e SW4 enviarão um ‘agreement’:



SW3 e SW4 não têm nenhuma outra interface, então eles enviarão um ‘agreement’ de volta para SW2:



SW2 colocará suas interfaces fa0/16 e fa0/19 no estado de encaminhamento e pronto. Este mecanismo de sincronização é apenas um par de mensagens que vão e vêm muito rápido, é muito mais rápido do que o mecanismo baseado em ‘tempo’ do spanning tree tradicional.

Ainda há mais três opções novas que precisamos ver:

- UplinkFast
- Mecanismo de mudança de topologia.
- Compatibilidade com Spanning Tree tradicional.

Para configurar o spanning tree tradicional, devemos habilitar o UplinkFast. O rapid spanning tree usa UpLinkFast por default. Quando um switch perde sua root port, ele coloca a ‘alternate port’ em estado de encaminhamento imediatamente.

A diferença é que o spanning tree tradicional precisa de frames multicast para atualizar as tabelas de endereços MAC dos switches.

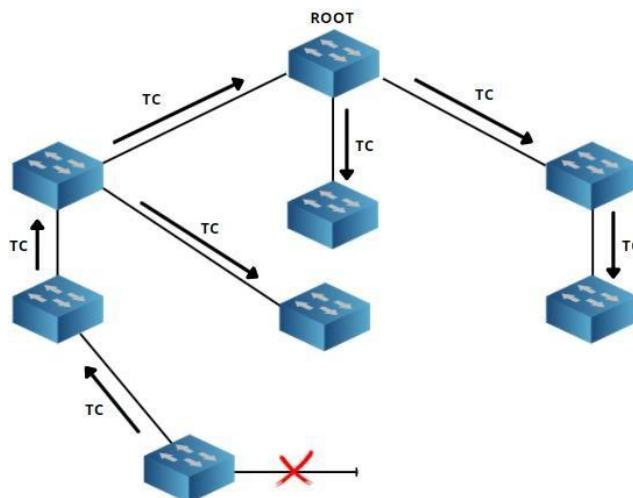
No rapid spanning tree o mecanismo de mudança de topologia é diferente.

No spanning tree tradicional, uma falha de link acionaria uma mudança de topologia. Com o rapid spanning tree, uma falha de link não é considerada uma mudança de topologia. Apenas interfaces não periféricas (link com outros switches) são consideradas como alteração de topologia. Assim que o switch detecta uma mudança na topologia, acontece o seguinte:

- Começa uma mudança de topologia utilizando o ‘timer’ que possui o dobro do tempo padrão do ‘hello timer’. Esse processo será realizado em todas as portas designadas non-edge e root ports.
- Irá apagar os endereços MAC que foram aprendidos por essas portas.
- Se a topologia mudar enquanto o ‘timer’ estiver ativo, ele definirá o ‘bit de mudança de topologia’ nos BPDUs que estão sendo enviados por essas portas. BPDUs também serão enviados pela root port.

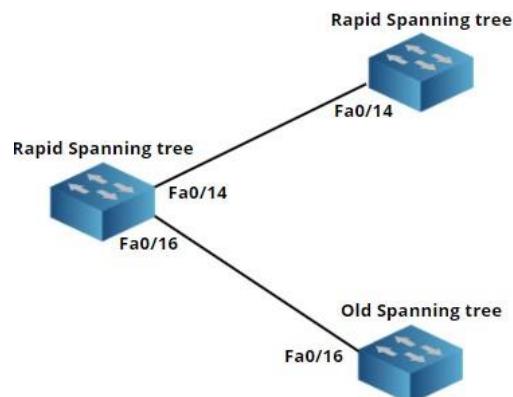
Quando um switch vizinho receber o BPDU com o ‘bits de mudança de topologia’ marcado, acontecerá os seguintes passos:

- O switch limpará todos os endereços MAC de todas as interfaces, exceto aquela em que recebeu o BPDU com a mudança de topologia ativo.
- Iniciará o processo de mudança de topologia, enviando BPDUs em todas as portas designadas e na root port, com o ‘bit de mudança de topologia (Topology Change Bit) marcado’.



Em vez de enviar uma mudança de topologia até a root bridge como é feito no spanning tree tradicional, a mudança de topologia agora é rapidamente ‘inundada’ por toda a rede.

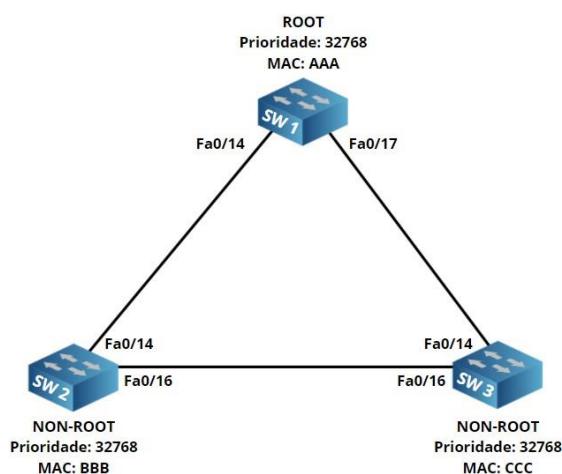
Por último, vamos falar sobre compatibilidade: O ‘rapid spanning tree’ e o ‘spanning tree’ tradicional **são compatíveis**. No entanto, quando um switch executando ‘rapid spanning tree’ se comunica com um switch executando o ‘spanning tree’ tradicional, todos os recursos do ‘rapid spanning tree’ ficam desativados!



No exemplo acima temos três switches. O link entre o SW1 e SW2 está executando o ‘rapid spanning tree’ e entre SW2 e SW3 está rodando o spanning tree tradicional.

Rapid Spanning Tree - Configuração

Observe a topologia abaixo:



Essa é a topologia que iremos utilizar, nela o SW1 é o root bridge. O primeiro passo é habilitarmos o rapid spanning tree nos três dispositivos:

```
SW1(config)#spanning-tree mode rapid-pvst
```

```
SW2(config)#spanning-tree mode rapid-pvst
```

```
SW3(config)#spanning-tree mode rapid-pvst
```

Esse é o comando para habilitarmos os rapid spanning tree nos switches: ‘spanning-tree mode rapid-pvst’, vamos calcular o rapid spanning tree para cada vlan.

Mas antes, vamos ver o mecanismo de sincronização (sync), para isso, vou desativar as duas interfaces do SW1:

```
SW1(config)#interface fa0/14
```

```
SW1(config-if)#shutdown
```

```
SW1(config)#interface f0/17
```

```
SW1(config-if)#shutdown
```

Agora vamos ativar o debug nos três switches para acompanhamos em tempo real:

```
SW1#debug spanning-tree events
```

```
Spanning Tree event debugging is on
```

```
SW2#debug spanning-tree events
```

```
Spanning Tree event debugging is on
```

```
SW3#debug spanning-tree events
```

```
Spanning Tree event debugging is on
```

Vamos ativar a interface fa0/14 no SW1 para vermos as reações que irão ocorrer nos switches:

```
SW1#
```

```
setting bridge id (which=3) prio 4097 prio cfg 4096 sysid 1 (on) id  
1001.0011.bb0b.3600
```

```
RSTP(1): initializing port Fa0/14
```

```
RSTP(1): Fa0/14 is now designated
```

```
RSTP(1): transmitting a proposal on Fa0/14
```

A interface fa0/14 no SW1 será bloqueada e enviará um ‘proposal’ para o SW2:

```
SW2#
```

```
RSTP(1): initializing port Fa0/14
```

```
RSTP(1): Fa0/14 is now designated
```

```
RSTP(1): transmitting a proposal on Fa0/14
```

```
RSTP(1): updting roles, received superior bpdu on Fa0/14
```

```
RSTP(1): Fa0/14 is now root port
```

Aparentemente, o SW2 pensou que ele era o root bridge, pois recebeu um BPDU superior em sua interface fa0/14. Dessa forma, ele muda sua interface fa0/14 para a root port.

```
SW2# RSTP(1): syncing port Fa0/16
```

A interface fa0/16 que é o link com SW3 entrará em modo de sincronização.

```
SW2# RSTP(1): synced Fa0/14
```

```
RSTP(1): transmitting an agreement on Fa0/14 as a response to a proposal
```

Em resposta ao ‘proposal’ enviado pelo SW1, o SW2 enviará um agreement (acordo).

```

SW1# RSTP(1): received an agreement on Fa0/14
%LINK-3-UPDOWN: Interface FastEthernet0/14, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/14, changed state
to up

```

SW1 recebe o agreement do SW2 e coloca a interface fa0/14 em modo de encaminhamento.

```
SW2# RSTP(1): transmitting a proposal on Fa0/16
```

SW3 responderá o ‘proposal’ do SW2 e enviará um ‘agreement’.

```

SW2# RSTP(1): received an agreement on Fa0/16
%LINK-3-UPDOWN: Interface FastEthernet0/14, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/14, changed state
to up

```

SW2 recebe o agreement do SW3, e imediatamente a interface fa0/14 muda seu status para UP entrando em modo de encaminhamento. Como vimos, todo o processo foi realizado de maneira rápida e sem o uso de temporizadores ‘timers’!

Vamos habilitar a interface fa0/17 para que a conectividade seja totalmente restaurada.

```

SW1(config)#interface fa0/17
SW1(config-if)#no shutdown

```

Vamos para um segundo, e dar uma visão geral de como está o rapid spanning tree até o momento:

```

SW1#show spanning-tree

VLAN0001

Spanning tree enabled protocol rstp

Root ID      Priority      4097
Address      0011.bb0b.3600

This bridge is the root

Hello Time    2 sec  Max Age 20 sec  Forward Delay 15 sec

Bridge ID    Priority      4097  (priority 4096 sys-id-ext 1)
Address      0011.bb0b.3600

Hello Time    2 sec  Max Age 20 sec  Forward Delay 15 sec

Aging Time   300

Interface          Role Sts Cost      Prio.Nbr Type
-----  -----  -----  -----  -----

```

Fa0/14	Desg FWD 19	128.16	P2p
Fa0/17	Desg FWD 19	128.19	P2p

O comando ‘show’ nos mostra que SW1 é a root bridge e que estamos executando o rapid spanning tree. Um detalhe importante que o comando nos mostra é que o tipo do link é **p2p (Point-to-point)**. O motivo é que as Interfaces FastEthernet trabalham em full duplex por padrão. Vamos executar o mesmo comando nos outros switches:

```
SW2#show spanning-tree

VLAN0001

  Spanning tree enabled protocol rstp

    Root ID      Priority    4097
                  Address     0011.bb0b.3600
                  Cost        19
                  Port       16 (FastEthernet0/14)
                  Hello Time 2 sec  Max Age 20 sec  Forward Delay 15 sec
    Bridge ID   Priority    8193  (priority 8192 sys-id-ext 1)
                  Address     0019.569d.5700
                  Hello Time 2 sec  Max Age 20 sec  Forward Delay 15 sec
                  Aging Time 300

    Interface      Role Sts Cost      Prio.Nbr Type
    -----
    Fa0/14        Root FWD 19      128.16    P2p
    Fa0/16        Desg FWD 19     128.18    P2p
```

```
SW3#show spanning-tree

VLAN0001

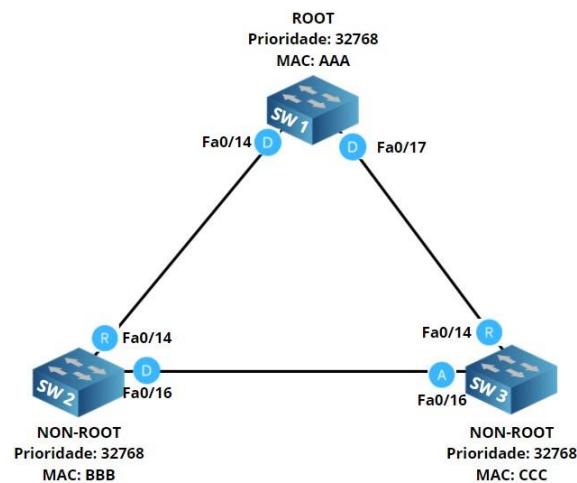
  Spanning tree enabled protocol rstp

    Root ID      Priority    4097
                  Address     0011.bb0b.3600
                  Cost        19
                  Port       14 (FastEthernet0/14)
                  Hello Time 2 sec  Max Age 20 sec  Forward Delay 15 sec
    Bridge ID   Priority    32769 (priority 32768 sys-id-ext 1)
                  Address     000f.34ca.1000
```

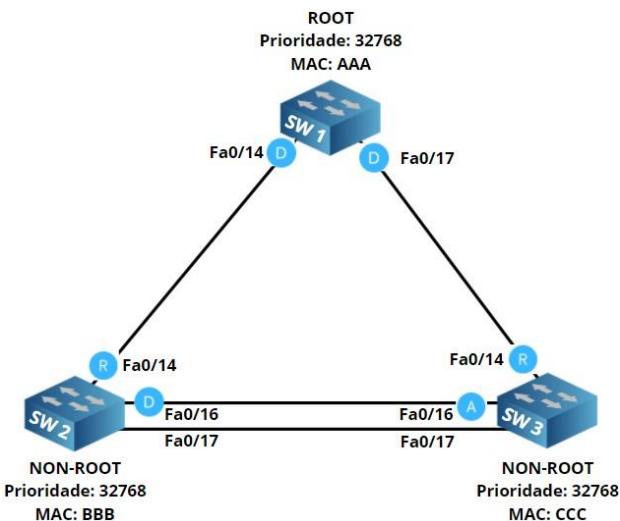
Hello Time	2 sec	Max Age	20 sec	Forward Delay	15 sec
Aging Time 300					
Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/14	Root	FWD	19	128.14	P2p

Estas são as saídas do comando no SW2 e SW3, não há mudança em relação ao Sw1.

Nossa topologia no momento está dessa forma:



Vamos adicionar outro link entre o SW2 e SW3 para ver como influenciará na nossa topologia:



SW2#show spanning-tree | begin Interface

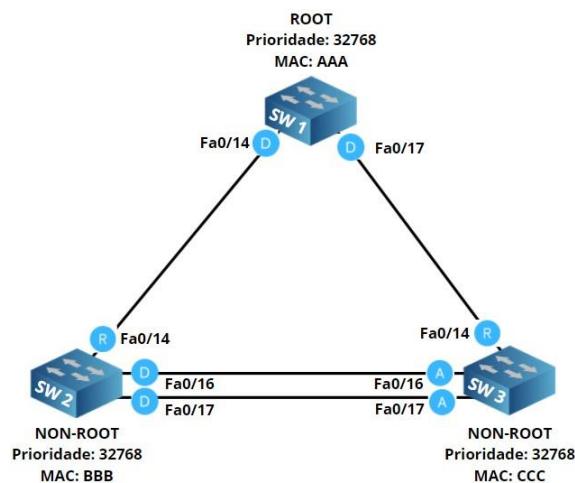
Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/14	Root	FWD	19	128.16	P2p
Fa0/16	Desg	FWD	19	128.18	P2p

Fa0/17	Desg FWD 19	128.19	P2p
--------	-------------	--------	-----

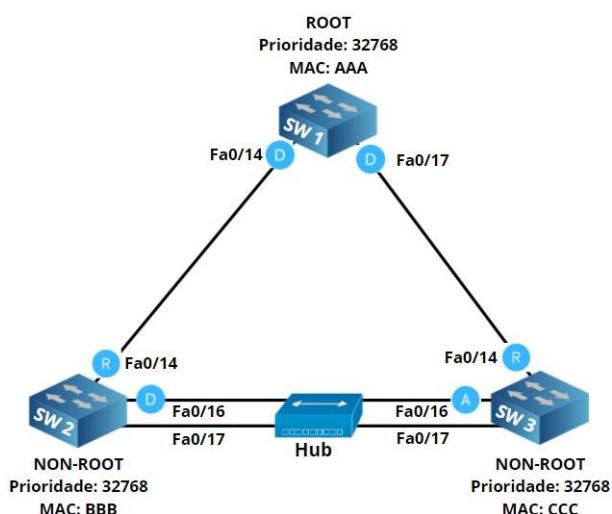
```
SW3#show spanning-tree | begin Interface
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
Fa0/14	Root	FWD	19	128.14	P2p
Fa0/16	Altn	BLK	19	128.16	P2p
Fa0/17	Altn	BLK	19	128.17	P2p

Agora, temos outra porta ‘Designada’ no SW2 e mais uma porta ‘Alternativa’ no SW3. Vou adicionar essa informação a nossa topologia:



Perceba que até agora não há muita diferença entre o rapid spanning tree e o spanning tree clássico, mas vamos deixar o estudo mais interessante adicionando um Hub entre o SW2 e SW3:



Vamos olhar novamente as interfaces:

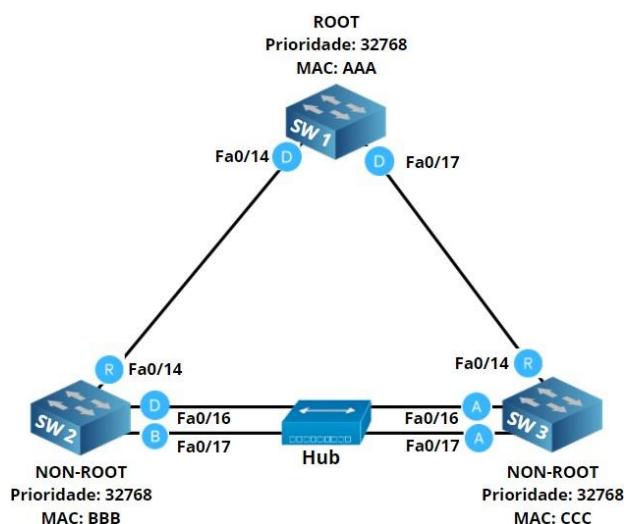
```
SW2#show spanning-tree | begin Interface
```

Interface	Role	Sts	Cost	Prio.	Nbr	Type
Fa0/14	Root	FWD	19	128.5		P2p
Fa0/16	Desg	FWD	100	128.3	Shr	
Fa0/17	Back	BLK	100	128.4	Shr	

```
SW3#show spanning-tree | begin Interface
```

Interface	Role	Sts	Cost	Prio.	Nbr	Type
Fa0/14	Root	FWD	19	128.5		P2p
Fa0/16	Altn	BLK	100	128.3	Shr	
Fa0/17	Altn	BLK	100	128.4	Shr	

Observe, agora temos uma mudança significativa, o Sw2 possui uma porta de backup devido ao hub. Observe que o tipo de link mudou, no momento ele está como Shr (shared ou compartilhado). Tudo isso porque o hub faz com que os switches tratem essas interfaces como Half duplex. Observe a topologia novamente:



Provavelmente você não verá esse cenário na vida real, afinal, não usamos mais hubs nos dias atuais.

BPDUs são enviados a cada dois segundos (o chamado hello time), e podemos observar isso através do comando debug:

```
SW2#debug spanning-tree bpdu
STP: VLAN0001 rx BPDU: config protocol = rstp, packet from FastEthernet0/14,
linktype IEEE_SPANNING, enctype 2, encsize 17
STP: enc 01 80 C2 00 00 00 11 BB 0B 36 10 00 27 42 42 03
```

```

STP: Data
000002023C10010011BB0B36000000000010010011BB0B360080100000140002000F00

STP: VLAN0001 Fa0/14:0000 02 02 3C 10010011BB0B3600 00000000 10010011BB0B3600
8010 0000 1400 0200 0F00

RSTP(1): Fa0/14 repeated msg

RSTP(1): Fa0/14 rcvd info remaining 6

RSTP(1): sending BPDU out Fa0/16

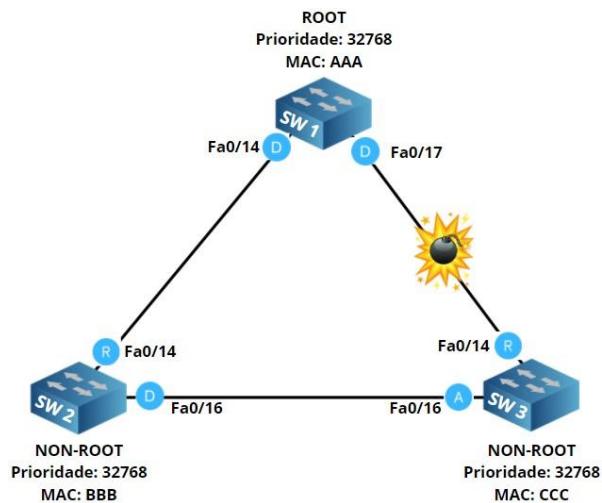
RSTP(1): sending BPDU out Fa0/17

STP: VLAN0001 rx BPDU: config protocol = rstp, packet f

```

Não são informações inteligíveis, porém, é possível ver que está havendo troca de BPDUs.

Vamos retirar o hub da rede e simular uma falha entre o SW1 e SW3:



Vamos dar um ‘shutdown’ na interface fa0/17 do SW1:

```

SW1(config)#interface fa0/17
SW1(config-if)#shutdown

```

O SW3 percebeu que havia algo de errado com a root port e imediatamente mudou a interface fa0/16, retirando o status de porta alternativa para root port.

```

SW3#
RSTP(1): updrt rolesroot port Fa0/14 is going down
RSTP(1): Fa0/16 is now root port

```

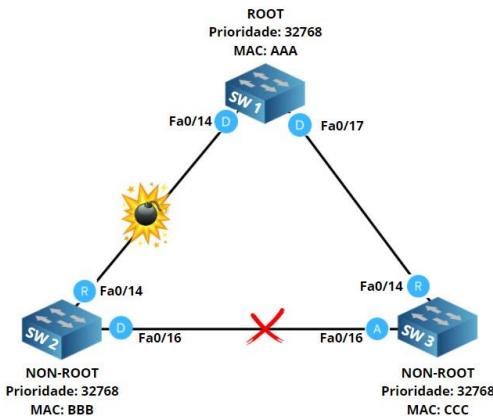
Vamos voltar com a interface fa0/17 no SW1 antes de continuarmos:

```

SW1(config)#interface fa0/17
SW1(config-if)#no shutdown

```

Agora, vamos simular uma falha indireta no SW3, colocando em shutdown o link que conecta o SW1 ao SW2.



```
SW1(config)#interface fa0/14
SW1(config-if)#shutdown
```

Ao desligar essa interface, afetaremos indiretamente o SW3:

```
SW2#
RSTP(1): upd roles, root port Fa0/14 going down
RSTP(1): we become the root bridge
RSTP(1): upd roles, received superior bpdu on Fa0/16
RSTP(1): Fa0/16 is now root port
```

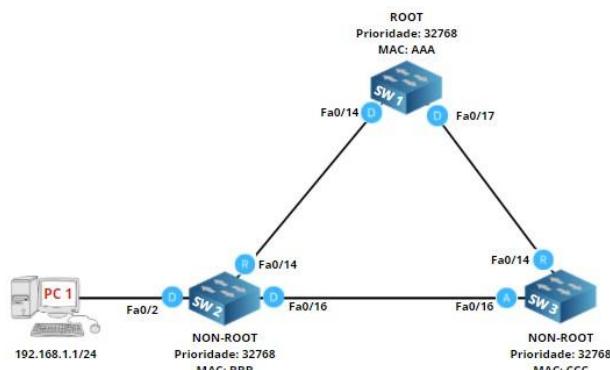
```
SW3#
03:41:29: RSTP(1): upd rolessuperior bpdu on Fa0/16 (synced=0)
03:41:29: RSTP(1): Fa0/16 is now designated
```

O Sw2 acreditará que ele é o root brigde até receber um BPDU superior do SW3.

Vamos religar a interface fa0/14 e prosseguir:

```
SW1(config)#interface fa0/14
SW1(config-if)#no shutdown
```

Vamos adicionar um computador à interface fa0/2 do SW2 e ver como o rapid spanning tree lida com interfaces conectadas a outros dispositivos:



```
SW2(config)#interface fa0/2
SW2(config-if)#no shutdown
RSTP(1): initializing port Fa0/2
RSTP(1): Fa0/2 is now designated
RSTP(1): transmitting a proposal on Fa0/2
%LINK-3-UPDOWN: Interface FastEthernet0/2, changed state to up
RSTP(1): transmitting a proposal on Fa0/2
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/2, changed state to up
RSTP(1): transmitting a proposal on Fa0/2
RSTP(1): Fa0/2 fdwhile Expired
```

Observe que o sistema de sincronização envia uma série de ‘proposal’ para o computador. Depois de um tempo eles expiram e a porta entra em estado de encaminhamento, porém, isso leva um tempinho.

Vamos informar ao switch que a interface que está ligando o computador é uma interface de borda.

```
SW2(config-if)#spanning-tree portfast
%Warning: portfast should only be enabled on ports connected to a single
host. Connecting hubs, concentrators, switches, bridges, etc... to this
interface when portfast is enabled, can cause temporary bridging loops.
Use with CAUTION

%Portfast has been configured on FastEthernet0/2 but will only
have effect when the interface is in a non-trunking mode.
```

Agora, vamos desativar e ativar a interface para acompanhamos em tempo real o comportamento do rapid spanning tree:

```
SW2(config)#interface fa0/2
SW2(config-if)#shutdown
SW2(config-if)#no shutdown
```

A interface irá entrar no modo de encaminhamento imediatamente. O switch sabe que essa é uma interface de borda e por isso não há necessidade de enviar ‘proposal’ para ela.

```
SW2#  
RSTP(1): initializing port Fa0/2  
RSTP(1): Fa0/2 is now designated  
*Mar 1 04:08:32.931: %LINK-3-UPDOWN: Interface FastEthernet0/2, changed state to up
```

Para fechamos esse tópico, ainda precisamos estudar a compatibilidade: Vamos mudar o SW2 para PVST e deixarmos o SW1 e SW3 no rapid-PVST:

```
SW2(config)#spanning-tree mode pvst
```

Eis o que acontece:

```
SW2(config)#  
RSTP(1): updт roles, non-tracked event  
setting bridge id (which=3) prio 8193 prio cfg 8192 sysid 1 (on) id 2001.0019.569d.5700  
set portid: VLAN0001 Fa0/2: new port id 8004  
STP: VLAN0001 Fa0/2 ->jump to forwarding from blocking  
set portid: VLAN0001 Fa0/14: new port id 8010  
STP: VLAN0001 Fa0/14 -> listening  
set portid: VLAN0001 Fa0/16: new port id 8012  
STP: VLAN0001 Fa0/16 -> listening^Z  
STP: VLAN0001 heard root 4097-0011.bb0b.3600 on Fa0/16 supersedes 8193-0019.569d.5700  
STP: VLAN0001 new root is 4097, 0011.bb0b.3600 on port Fa0/16, cost 38  
STP: VLAN0001 new root port Fa0/14, cost 19  
STP: VLAN0001 Fa0/14 -> learning  
STP: VLAN0001 Fa0/16 -> learning  
STP: VLAN0001 sent Topology Change Notice on Fa0/14  
STP: VLAN0001 Fa0/14 -> forwarding  
STP: VLAN0001 Fa0/16 -> forwarding
```

Perceba que o SW2 recebe BPDU da root bridge e que as interfaces passam pelo estado de escuta (listening) e aprendizagem (learning), até chegar no estado de encaminhamento (forwarding). Quando os switches que estão rodando rapid spanning tree recebem BPDU dos switches rodando spanning tree tradicional, eles próprios geram BPDU do spanning tree tradicional para que tudo continue funcionando.

Podemos confirmar que tudo está funcionando como previsto através do comando ‘show’:

SW1#show spanning-tree begin Interface						
Interface	Role	Sts	Cost	Prio.Nbr	Type	
Fa0/14	Desg	FWD	19	128.16	P2p	Peer(STP)
Fa0/17	Desg	FWD	19	128.19	P2p	

SW2#show spanning-tree begin Interface						
Interface	Role	Sts	Cost	Prio.Nbr	Type	
Fa0/2	Desg	FWD	19	128.4	P2p	Edge
Fa0/14	Root	FWD	19	128.16	P2p	
Fa0/16	Desg	FWD	19	128.18	P2p	

SW3#show spanning-tree begin Interface						
Interface	Role	Sts	Cost	Prio.Nbr	Type	
Fa0/14	Root	FWD	19	128.14	P2p	
Fa0/16	Altn	BLK	19	128.16	P2p	Peer(STP)

As configurações são simples, porém o conceito é extenso, apesar de não ser complicado.

2.5.b Port states (forwarding/blocking)

Já falamos um pouco sobre os status da porta na introdução ao spanning-tree, mas agora entraremos nos detalhes para cobrir tudo que é cobrado nesse tópico.

Quando conectamos um cabo em alguma porta do switch o led localizado em cima da interface fica laranja e depois de um tempo fica verde. O que ocorre nos bastidores é que o spanning-tree está determinando o ‘estado (status)’ da interface.

Assim que conectamos um cabo ocorre a sequência abaixo:

- A porta entra em estado de **escuta (listening mode)** por 15 segundos. Apenas a root port ou a designated port entram no estado de escuta. Nenhuma transmissão de dados ocorre nesses 15 segundos, pois o switch está verificando se houve alguma mudança na topologia da rede.
- A porta entra em estado de **aprendizagem (learning mode)** por 15 segundos. Neste momento as interfaces estão aprendendo os endereços MAC de origem dos quadros ethernet que chegaram até ela. É o momento em que a Tabela de endereços Mac é preenchida.
- A porta entra em estado de **encaminhamento (fowarding mode)**, esse é o ‘estado final’ da interface, é quando ela finalmente começa a transmitir dados!

Quando uma porta não é a root port ou a designated port ela estará em modo de bloqueio (**blocking mode**).

Podemos afirmar que leva 30 segundos para as interfaces passarem do modo escuta para o modo de encaminhamento. Porém, quando uma interface está no modo de bloqueio e a topologia é alterada, é possível que essa interface tenha que passar para o modo de encaminhamento. Neste caso específico, o modo de bloqueio durará 20 segundos antes de passar para o estado de escuta. Resumindo, levará 50 segundos para que a interface esteja no estado de encaminhamento: 20 (bloqueio) + 15 (escuta) + 15 (aprendizagem).

Estado	Encaminhando frames	Aprendendo endereço MAC	Duração
Blocking	Não	Não	20 segundos
Listening	Não	Não	15 segundos
Learning	Não	Sim	15 segundos
Fowarding	Sim	Sim	-

Vamos ver como essas mensagens aparecem no log de um switch Cisco. Temos um switch conectado a um roteador, e vamos desconectar e conectar o cabo novamente e acompanhar através de alguns comandos como o switch se comporta:

```
SW1#show spanning-tree vlan

VLAN0001

  Spanning tree enabled protocol ieee

    Root ID      Priority    32769
                  Address     0019.569d.5700
                  This bridge is the root

    Hello Time    2 sec  Max Age 20 sec  Forward Delay 15 sec

    Bridge ID    Priority    32769  (priority 32768 sys-id-ext 1)
                  Address     0019.569d.5700
                  Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
                  Aging Time  300

    Interface      Role Sts Cost      Prio.Nbr Type
    -----  -----
    Fa0/1          Desg LIS 19       128.4      P2p
```

Observe que a função (role) da porta está como ‘designated’ e o status está como ‘listening’. Vamos aplicar novamente o comando ‘Show spanning-tree vlan’ depois de 15 segundos para verificarmos a saída:

```
SW1#show spanning-tree vlan 1

VLAN0001

  Spanning tree enabled protocol ieee

    Root ID      Priority    32769
```

```

Address      0019.569d.5700

This bridge is the root

Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec


Bridge ID  Priority      32769  (priority 32768 sys-id-ext 1)

Address      0019.569d.5700

Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

Aging Time 300


Interface          Role Sts Cost      Prio.Nbr Type
-----  

Fa0/1             Desg LRN 19       128.4      P2p

```

Agora a porta está com status de learning. Deixemos passar mais 15 segundos e aplicamos novamente o comando:

```

SW1#show spanning-tree vlan 1

VLAN0001

Spanning tree enabled protocol ieee

Root ID    Priority      32769

Address      0019.569d.5700

This bridge is the root

Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec


Bridge ID  Priority      32769  (priority 32768 sys-id-ext 1)

Address      0019.569d.5700

Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

Aging Time 15


Interface          Role Sts Cost      Prio.Nbr Type
-----  

Fa0/1             Desg FWD 19       128.4      P2p

```

Agora a porta entrou em estado de encaminhamento e você acabou de ver na prática tudo que expliquei na teoria. Mas, um método melhor para ver toda essa mudança acontecendo é usando o comando ‘debug’:

```
SW1#debug spanning-tree events  
Spanning Tree event debugging is on
```

E agora vamos desabilitar e habilitar a interface novamente, observe o log abaixo:

```
SW1#  
00:14:57: STP: VLAN0001 Fa0/1 -> listening  
00:15:12: STP: VLAN0001 Fa0/1 -> learning  
00:15:27: STP: VLAN0001 Fa0/1 -> forwarding
```

2.5.c PortFast benefits

PortFast é uma solução prioritária da Cisco para lidar de forma mais rápida com mudanças na topologia de rede. O PortFast tem duas funções principais:

- As interfaces com portfast habilitado entram em modo de encaminhamento imediatamente, elas não entram no estado de escuta e aprendizagem.
- Switches não geram notificações de mudança de topologia em caso de mudança em interfaces que tenham o portfast habilitado.

Por isso, o portfast deve ser habilitado em interfaces que conectam hosts, afinal, essas interfaces provavelmente ficarão up e down o tempo todo (computadores são ligados e desligados várias vezes durante o dia). Não devemos habilitar o portfast em uma interface que esteja conectada a um hub ou switch.

Vamos ver a diferença de uma interface com e sem portfast. Para isto, usaremos a seguinte topologia:



Na topologia acima, temos dois switches e um computador.

PortFast desabilitado

Para acompanhamos em tempo real o que está acontecendo no dispositivo, vamos habilitar o debug no SW1:

```
SW1#debug spanning-tree events  
Spanning Tree event debugging is on
```

Assim que conectamos o PC1, eis o que acontece:

```
SW1#  
STP: VLAN0001 Fa0/1 -> listening  
STP: VLAN0001 Fa0/1 -> learning  
STP: VLAN0001 Fa0/1 -> forwarding
```

Esse é o comportamento típico do Spanning tree, observe que a interface passou por todos os estados: Escuta, aprendizagem e por fim encaminhamento.

Toda vez que desconectarmos o PC1, o SW1 irá gerar um aviso de mudança de topologia (TCN – Topology Change Notification). O comando abaixo mostra em detalhes esse processo acontecendo:

```

SW1#show spanning-tree detail

VLAN0001 is executing the ieee compatible Spanning Tree protocol

Bridge Identifier has priority 32768, sysid 1, address 0019.569d.5700

Configured hello time 2, max age 20, forward delay 15

Current root has priority 32769, address 0011.bb0b.3600

Root port is 26 (FastEthernet0/24), cost of root path is 19

Topology change flag not set, detected flag not set

Number of topology changes 5 last change occurred 00:02:09 ago

    from FastEthernet0/1

Times: hold 1, topology change 35, notification 2

    hello 2, max age 20, forward delay 15

Timers: hello 0, topology change 0, notification 0, aging 300

```

Tivemos cinco mudanças de topologia na Vlan. Vamos desplugar o PC1 do switch para vermos o que acontece:

```

SW1#
STP: VLAN0001 sent Topology Change Notice on Fa0/24

```

O spanning-tree enviou uma notificação de mudança de topologia (TCN) na interface que o SW2 está conectado e aumentou o contador de notificações de mudança de topologia (TCN):

```

SW1#show spanning-tree detail | include changes

Number of topology changes 6 last change occurred 00:01:12 ago

```

Toda as vezes que houver uma mudança na interface conectada ao PC1 o SW1 enviará uma TCN na rede. Vamos ver a diferença com o portsfast habilitado.

Portfast habilitado

Vamos habilitar o portfast na interface fa0/1, interface que o host está conectado para ver o que acontece:

```

SW1(config)#interface FastEthernet 0/1
SW1(config-if)#spanning-tree portfast

%Warning: portfast should only be enabled on ports connected to a single
host. Connecting hubs, concentrators, switches, bridges, etc... to this
interface when portfast is enabled, can cause temporary bridging loops.

Use with CAUTION

%Portfast has been configured on FastEthernet0/1 but will only
have effect when the interface is in a non-trunking mode.

```

O IOS emite um aviso que o portfast não deve ser habilitado em interfaces que conectam outros switches. O comando ‘spanning-tree portfast default’ habilita o portfast em todas as interfaces do switch que estejam em modo de acesso. Isso poupa muito tempo, pois já não é mais necessário entrar de porta em porta para aplicar o comando.

Vamos conectar o PC1 novamente ao switch:

```
SW1#  
STP: VLAN0001 Fa0/1 ->jump to forwarding from blocking
```

Observe que o switch pulou diretamente do modo de bloqueio para o modo de encaminhamento, ele também não gerou nenhuma mensagem de TCN.

2.6 Compare Cisco Wireless Architectures and AP modes

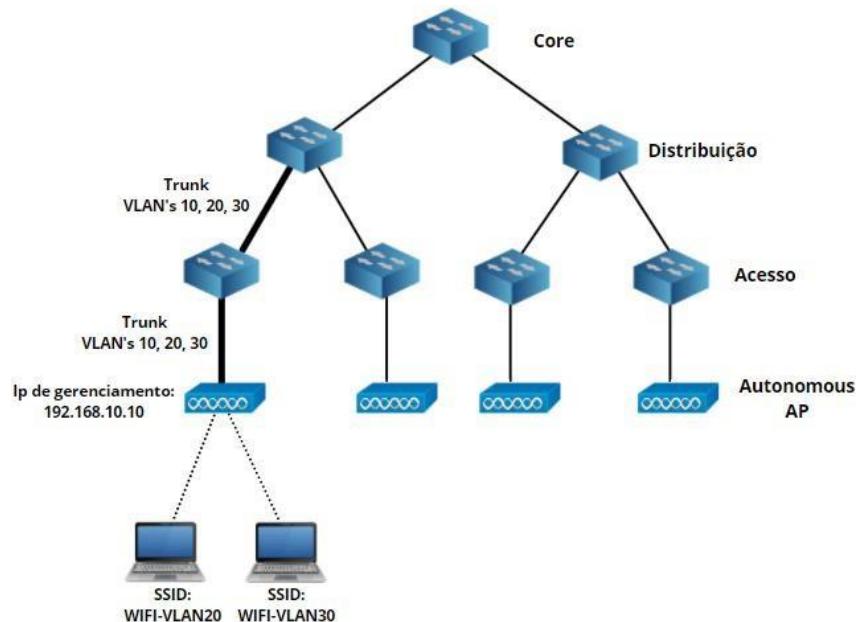
Este capítulo trata sobre os diferentes tipos de arquitetura wireless que a Cisco oferece pra redes corporativas.

Autonomous AP Architecture

Na grande maioria das vezes, usamos a rede sem fio como extensão da rede cabeada, onde compartilhamos as mesmas vlans e demais recursos.

Um AP autônomo (autonomous AP) tem todos os requerimentos necessários para atender clientes sem fio e conecta-los a rede cabeada. O AP pode oferecer um ou mais BSS e conectar vlans a SSIDs.

Observe na topologia abaixo a representação de uma rede corporativa:



As linhas destacadas (mais grossas) representam a conexão para um autonomous Access Point, esse access point está na camada de acesso conectado no switch através de um trunk. Nele temos três vlans:

- VLAN 10: Gerenciamento
- VLAN 20: para SSID “WIFI-VLAN20”
- VLAN 30: para SSID “WIFI-VLAN30”

O protocolo 802.1Q se faz necessário pois o autonomous Access Point precisa ter acesso às três vlans. O autonomous Access Point possui um endereço de IP para gerenciamento. Esse endereço IP serve para que possamos nos conectar a ele remotamente e configurarmos os seguintes itens:

- Parâmetros RF:
 - Canal
 - Potência de transmissão
- VLANs
- SSIDs

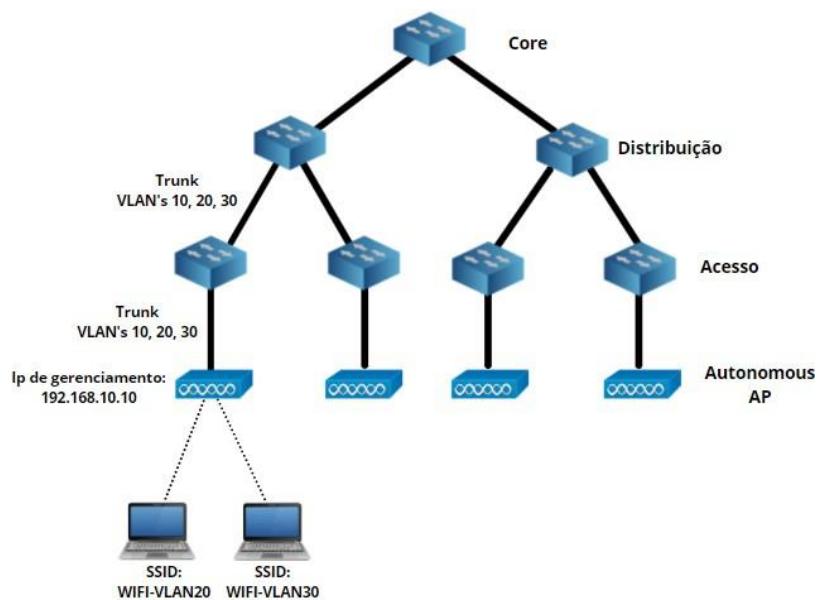
É uma boa prática de segurança separar as vlan de dados da vlan de gerenciamento, por isso, utilizamos uma vlan só para o gerenciamento. Os usuários da rede wireless conseguem comunicar uns com os outros sem passarem pela rede cabeada.

Um dos grandes problemas dos APs autônomos é que eles são configurados individualmente, isso pode ser uma grande dor de cabeça, por exemplo:

Um recurso normal nas empresas é que os usuários sejam capazes de fazer ‘roaming’ de um AP para outro sem perder a conexão e o endereço IP que foi atribuído por DHCP. Isso é possível com o autonomous Access Point, porém nesse caso, deveremos configurar manualmente o mesmo SSID e Vlan em todos os access points.

Também será necessário configurar os parâmetros de RF, como o canal que queremos usar, por exemplo. Com múltiplos APs teremos que descobrir quais canais e frequências devemos usar para que haja sobreposição mas sem interferências significativas, também teremos que assegurar que quando um AP falhar, outro assuma seu lugar para que não haja pontos sem cobertura.

A exigência de vlans em todos os lugares acarreta um outro problema: Vlans estendidas:



Quando um usuário wireless faz roaming de um AP para outro que está conectado a camada de acesso de outra camada de distribuição, significa que essa vlan terá que atravessar a camada de núcleo, ou seja, a vlan deverá abranger toda a rede.

Caso seja necessário configurar um nosso SSID, deveremos configurá-la em todos os APs, também precisaremos configurar uma nova vlan em todos os APs e switches. Não há um ponto central para monitorar o tráfego wireless e utilizar ferramentas como QoS, detecção de intrusão etc.

É possível deixar a vida um pouco mais fácil através de uma plataforma de gerenciamento como Cisco Prime Infrastructure.

Importante lembrar que a solução **autonomous AP** também é conhecida como **Local Mac Architecture**.

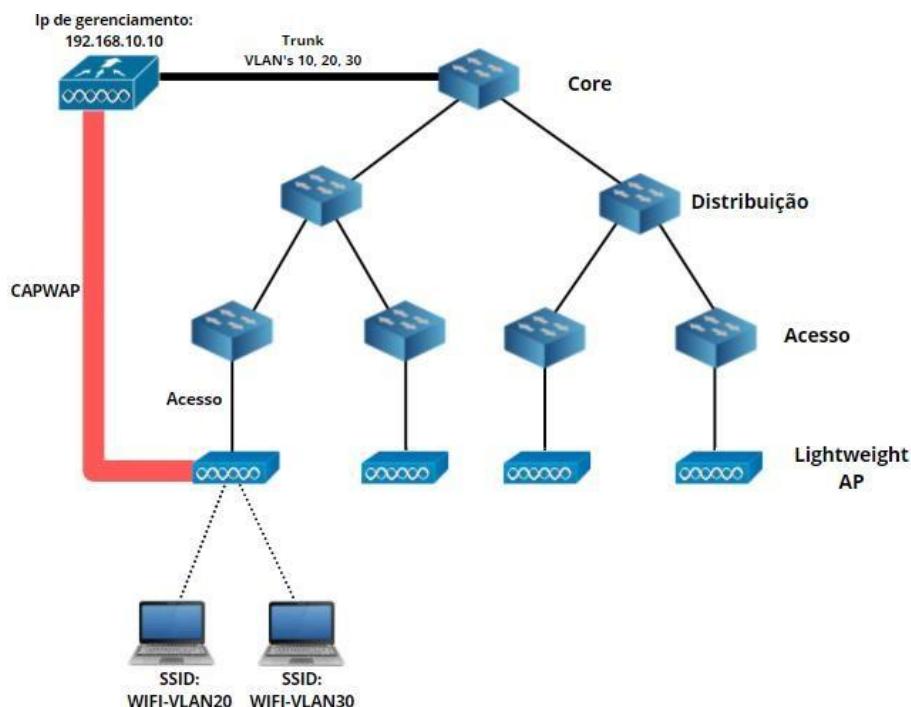
Como você pode ver, há alguns problemas e limitações para utilização dos APs autônomos, de forma que eles são uma boa solução para pequenas redes, mas se tornam inviáveis para médias e grandes redes.

Split-MAC Architecture

APs autônomos trabalham sozinho, são configurados individualmente, conforme explicamos anteriormente, essa arquitetura descentralizada tem algumas desvantagens:

- Para que haja roaming, é necessário configurar VLANs e SSIDs em todos os APs.
- Para criação de um novo SSID é necessário criar uma nova VLAN em todos os switches.
- As VLANs podem se espalhar por toda a rede cabeada.
- Precisamos configurar os parâmetros RF manualmente.
- Não existe um ponto central na rede para gerenciamento do tráfego wireless para a realização de QoS entre outros.

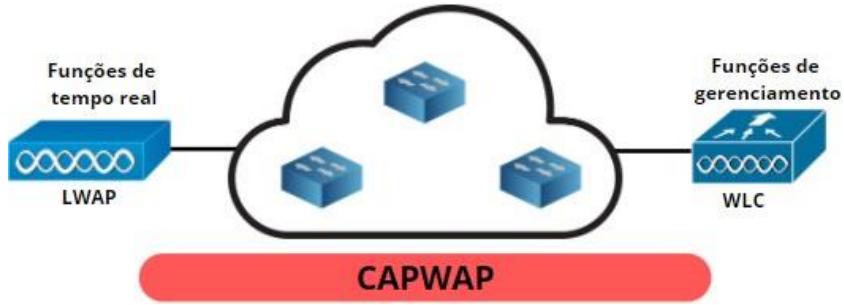
Para superar essas limitações, algumas funções executadas pelo AP foram movidas para um ponto central chamado de Wireless Lan Controller (WLC).



Em uma rede sem fio, temos **funções em tempo real e de gerenciamento**. O AP deve lidar com funções em tempo real, mas tudo o que não é sensível a atrasos pode ser realizado a partir do ponto central. Separei abaixo as funções de gerenciamento e as funções de tempo real:

- **Funções de gerenciamento:**
 - Autenticação de cliente
 - Gerenciamento de segurança
 - Associação e reassociação (roaming)
 - Qualidade de serviço (QoS)
- **Funções em tempo real:**
 - Transmissão de frames 802.11
 - Gerenciamento MAC
 - Criptografia

Ao retirarmos as funções que não são de tempo real e movê-las para o **WLC** (ponto central), acabamos tirando parte da ‘inteligência’ do AP, por isso, os access point nesse modelo são chamados de **lightweight APs (LAP)**, podemos traduzir de maneira bem brasileirada como ‘ponto de acesso leve’:



Um WLC pode controlar uma grande quantidade de APs na rede. Obrigatoriamente, um lightweight AP deve estar vinculado a um WLC, não funcionando por conta própria. A divisão de funções entre o AP e o WLC é o que chamamos de arquitetura Split-MAC.

Uma exceção é a arquitetura FlexConnect, onde o AP se vincula a um WLC, mas também pode funcionar de forma autônoma.

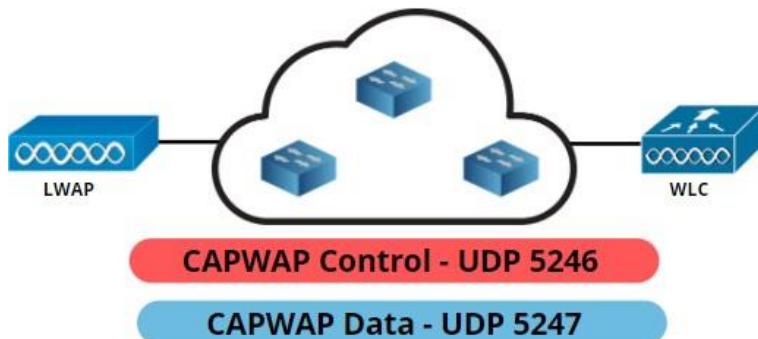
Quando um lightweight AP é inicializado, ele usa mecanismos de descoberta para pesquisar e se conectar a um WLC. O AP precisa se autenticar para conectar com o WLC. Essa autenticação é realizada através de certificados X.509 pré-instalados no AP e no WLC. Isso evita que alguém adicione um AP não autorizado à rede.

CAPWAP

A AP e WLC se conectam através de um protocolo de ‘tunelamento ou encapsulamento’, chamado Control And Provisioning of Wireless Access Points (**CAPWAP**). O CAPWAP encapsula todos os dados entre o lightweight AP e o WLC.

CAPWAP é um padrão definido pelas RFCs: 5415, 5416, 5417 e 5418. É baseado no Lightweight Access Point Protocol (LWAPP), uma solução proprietária legada da Cisco.

Existem dois tipos de túnel:

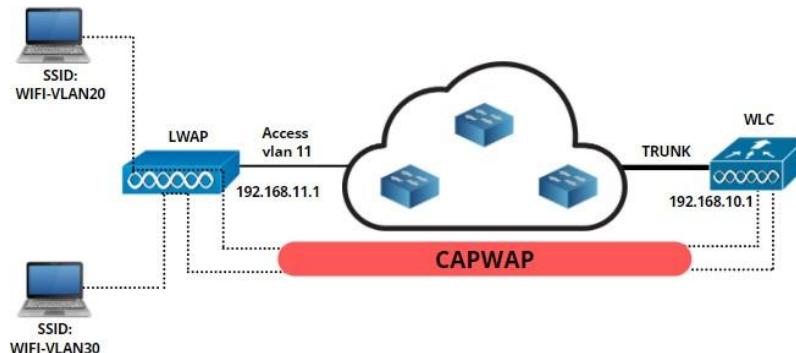


- Mensagens de controle CAPWAP:** Contêm informações sobre o gerenciamento da WLAN. Utilizado para configurar e gerenciar APs. As mensagens de controle são criptografadas.
- Mensagens de dados CAPWAP:** Encapsulam pacotes de e para clientes wireless associados ao AP. Por padrão, essas mensagens não são criptografadas.

Cada um deles utiliza uma porta UDP diferente, conforme o desenho acima mostra.

O tráfego do túnel pode ser **comutado ou roteado**. Usar um túnel significa que os APs lightweight e o WLC não precisam estar na mesma VLAN. Isso é bem interessante porque os APs normalmente estão na camada de acesso e o WLC está em um local central (camada de núcleo ou em um datacenter conectado ao núcleo).

O túnel CAPWAP faz com que o AP e o WLC não sejam separados apenas fisicamente, mas também logicamente. Vamos entender melhor com um desenho:

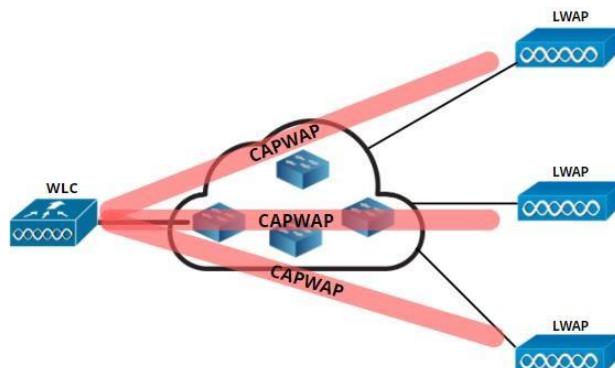


O AP lightweight e o WLC possuem endereços IPs em redes distintas, mas isso não é problema para construção do túnel, pois, se houver conectividade entre o AP e o WLC o túnel será formado. Observe que lightweight AP está conectado a uma porta de switch que está no modo de acesso, na VLAN 11, e possui dois SSIDs:

- WIFI-VLAN20: Que como o nome indica, é a VLAN 20
- WIFI-VLAN30: Que como o nome indica, é a VLAN 30

Você pode estar se perguntando: Como o ap lightweight consegue usar SSIDs com as VLANs 20 e 30 estando conectado a uma interface na VLAN 11? Bom, esse é um dos motivos do túnel CAPWAP; ele canaliza o tráfego da VLAN 20 e 30 da WLC até o AP lightweight.

Isso significa que a WLC precisa de acesso a todas as VLANs, portanto, ela precisa se conectar a uma interface trunk. O AP lightweight pode estar em qualquer VLAN, contanto que tenha conectividade com o WLC.



Com o WLC executando funções de gerenciamento para todos os APs lightweight, é possível executar funções que não podemos executar com APs autônomos. Como o WLC tem acesso às informações RF de todos os APs, abriu uma janela grande de oportunidade:

- **Monitoramento de RF:** O WLC faz a varredura dos canais e monitora o uso de RF de todos os APs. Ele usa esses dados para selecionar os melhores canais, a potência a ser usada e detectar APs não autorizados.
- **Sistema de proteção contra intrusão sem fio (WIPS):** O WLC pode monitorar os dados dos clientes wireless para detectar e prevenir o acesso não autorizado à rede
- **Gerenciamento de segurança:** O WLC pode autenticar clientes wireless em um servidor externo (RADIUS) e forçar os clientes a usar endereços IPs distribuídos por um servidor DHCP confiável.
- **Balanceamento de carga dinâmico:** Quando dois APs estão próximos um do outro, o WLC pode associar clientes ao AP menos usado. Isso ajuda a distribuir a carga entre os APs.
- **Roaming de clientes:** Clientes wireless podem fazer roaming entre APs sem qualquer interferência ou atraso na conexão.
- **Atribuição dinâmica de canal:** O WLC escolhe automaticamente o melhor canal RF para cada AP.
- **Otimização da potência de transmissão:** O WLC define automaticamente a potência de transmissão para cada AP com base na área de cobertura necessária.

- **Cobertura sem fio ‘self-healing’**: Semelhante ao Wolverine, com fator de cura. Quando um rádio AP morre, o WLC pode aumentar a potência de transmissão dos APs circundantes para eliminar um buraco na cobertura.

Cloud-Based AP Architecture

Até agora vimos arquitetura MAC local com APs autônomos e suas deficiências e arquitetura split-MAC com APs lightweight e WLCs, e como ela resolve alguns dos problemas da arquitetura MAC local.

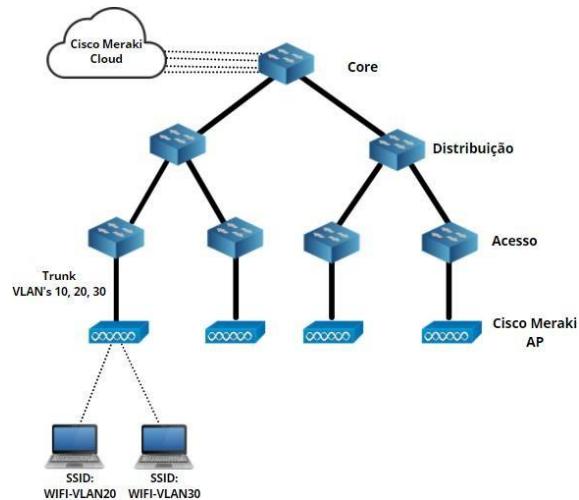
APs lightweight exigem WLC para funcionar. Com o WLC, o gerenciamento dos APs é bem mais fácil. Porém, por controlar todos os APs o WLC acaba se tornando um dispositivo vital para o funcionamento da rede, por isso, é interessante que tenhamos redundância, o que acaba tornando a configuração mais complicada.

Uma alternativa é uma arquitetura AP baseada em nuvem, nessa arquitetura, a função da controladora é enviada para a nuvem. A Cisco oferece isso com seus produtos Meraki.

A Meraki oferece produtos de segurança, switches e dispositivos wireless baseados na nuvem. Nessa arquitetura, quando o AP é ligado, ele se conecta automaticamente à nuvem e se configura. Por meio do painel da Meraki baseado em nuvem é possível:

- Configurar APs.
- Enviar atualizações para APs.
- Monitorar o desempenho da rede wireless.
- Gerar relatórios.

Assim como WLC tradicional, a “nuvem” instrui cada AP sobre qual canal e potência de transmissão usar. Ele também coleta parâmetros de RF como interferência, utilização, APs invasores, etc.



Semelhante à arquitetura split-MAC a arquitetura cloud possui dois caminhos de tráfego:

- **Control Plane**: Configura, gerencia e monitora APs.
- **Data Plane**: Tráfego de e para clientes sem fio.

O Control Plane fica na nuvem e é utilizado apenas para gerenciamento. O Data Plane permanece na rede local e não é encaminhado para a nuvem. Isso significa que cada AP requer uma porta trunk para o switch, semelhante à arquitetura de AP autônomo.

Resumo

- **Arquitetura autônoma de AP**:
 - Toda a inteligência está dentro do AP.
 - Necessário trunk entre o switch e o AP.

- APs autônomos funcionam de forma autônoma, são configurados de forma individual.
- Se quiser que haja roaming, é necessário configurar as VLANs e SSIDs em todos os switches.
- Há problemas em ter de estender as VLANs por toda rede.
- Não há ponto central para gerenciamento, o que torna difícil\impossível implantação de recursos como policiamento, gerenciamento de RF, etc.
- **Arquitetura Split-MAC:**
 - A inteligência do AP é ‘transplantada’ para o WLC:
 - As funções de tempo real permanecem no AP lightweight.
 - As funções de gerenciamento ficam no WLC.
- **CAPWAP:**
 - Canaliza todo o tráfego do AP lightweight para o WLC.
 - Utiliza um túnel para controle e outro para dados, cada um utilizando uma porta UDP diferente.
 - Podemos rotear ou comutar o tráfego CAPWAP.
 - Isso significa que o AP lightweight e o WLC são separados física e logicamente.
 - O AP lightweight se conecta ao switch através de uma interface no modo de acesso.
 - O WLC se conecta ao switch através de uma interface trunk.
 - O WLC tem acesso a todas as informações do AP, para que possa tomar decisões de gerenciamento de RF, segurança, monitoramento, etc.
- **Arquitetura AP baseada em nuvem:**
 - As funções de gerenciamento são transportadas para a nuvem.
 - Os dados permanecem na rede local, portanto, é utilizado uma interface trunk entre o switch e o AP.

AP Modes

APs Cisco podem operar em modo autônomo ou lightweight; o modo de operação é definido pela imagem do AP que estamos executando.

Um AP que atende clientes wireless está no **modo local (local mode)**. Além do modo local, existem outros modos. Veremos cada um deles a seguir:

1. Local

O modo local é o modo padrão; oferece um BSS em um canal específico. Quando o AP não está transmitindo frames dos clientes wireless, ainda assim está trabalhando nos bastidores, varrendo outros canais para:

- Medir ruído
- Medir interferência
- Descobrir dispositivos invasores

2. Monitor

Um AP no ‘modo monitor’ **não transmite** nada. É um sensor dedicado a:

- Verificar eventos do Sistema de Detecção de Intrusão (IDS)
- Detectar APs invasores
- Determinar a posição das estações sem fio

Como o AP está apenas no modo monitor, ele não transmitirá SSID, dessa forma, os clientes não conseguem se conectar a ele.

3. FlexConnect

É possível conectar um AP que esteja em ‘modo local’, localizado em um escritório remoto, a WLC do escritório central. Isso funciona, mas não é uma boa ideia: O AP encapsulará todos os dados por meio do túnel CAPWAP e enviará através do link WAN, o que pode causar gargalos na rede. Outro problema é que quando o link WAN cair, a rede wireless na filial também ficará offline.

O FlexConnect é um modo AP que resolve situações como essas descritas acima. Com ele, o AP pode comutar localmente o tráfego entre VLANs e SSID quando o túnel CAPWAP estiver indisponível.

4. Sniffer

Um AP no modo sniffer (farejador) apenas recebe frames wireless, literalmente o AP torna-se um sniffer remoto da rede sem fio. É possível se conectar ao AP através de um computador, e utilizando aplicativos como Wildpackets Omnipacket ou Wireshark analisar e solucionar problemas com a rede local de forma remota. Quando um AP está no modo sniffer, ele não transmite SSIDs, dessa forma, os clientes não conseguem se conectar ao AP.

5. Rogue Detector

O ‘modo rogue detector’ faz com que o AP detecte dispositivos não autorizados na rede em tempo integral. O AP verifica os endereços MAC tanto na rede sem fio como na rede cabeada. Quando o AP está no modo ‘detector de invasores’, ele pode alternar entre a detecção de invasores e serviço aos clientes, ou seja, o AP pode transmitir SSID e os clientes podem se conectar a ele.

6. Bridge/Mesh

O AP torna-se uma ponte (bridge) dedicada, seja ponto a ponto ou ponto a multiponto. Dois APs em modo bridge podem conectar dois sites remotos. Vários APs podem formar uma malha (mesh) interna ou externa. Não é possível conectar cliente a APs no modo bridge\mesh.

7. Flex plus bridge

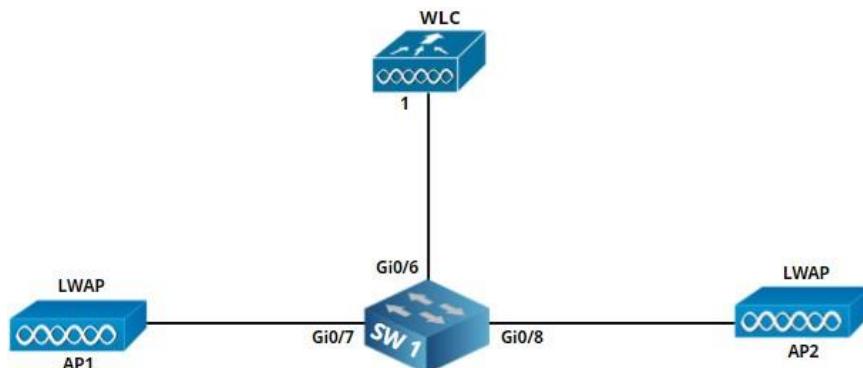
O AP pode operar no modo FlexConnect ou Bridge/Mesh. Este modo combina os dois; ele permite que os APs em modo Bridge/Mesh usem os recursos do FlexConnect.

8. SE-Connect

Um AP no modo SE-Connect dedica seus rádios à análise de espectro dos canais wi-fi. Conectando o AP através de um computador e utilizando aplicativos como MetaGeek Chanalyzer ou Cisco Spectrum Expert é possível descobrir remotamente fontes de interferência que não podem ser resolvidas com um simples sniffer de rede. O AP não transmitirá SSID para que os clientes não possam se conectar a ele.

2.7 Describe physical infrastructure connections of WLAN components (AP, WLC, access/trunk ports, and LAG)

A melhor forma de atender o que é exigido nesse tópico é configurando na prática um WLC e o switch em que ele se conectará. Para isso, usaremos a topologia abaixo:



Essa rede terá três VLANs: 10, 20 e 30:

- VLAN 10 será a VLAN de gerenciamento. O WLC utiliza a interface de gerenciamento para se comunicar com os access point. A interface de gerenciamento também é utilizada para configurar o WLC por meio do SSH ou da GUI.
- VLAN 20 e 30 são para as redes wireless.

Consiste em uma boa prática, separar o tráfego de gerenciamento do tráfego de dados dos clientes da rede wireless, é por isso que temos uma VLAN de gerenciamento separada. Cada SSID pode ser mapeada para uma VLAN diferente, portanto, com duas VLANs podemos criar duas redes wireless separadas. Por exemplo, você pode criar uma rede wireless para usuários corporativos e outra para usuários convidados.

SW1 e o WLC terão endereço IP estáticos na VLAN 10:

- WLC1: 192.168.10.100
- SW1: 192.168.10.254

Vamos configurar o SW1 como um servidor DHCP, dessa forma, os access point receberam endereços IP de forma dinâmica. Os access point poderão encontrar a WLC automaticamente porque estão na mesma VLAN.

Switch

Vamos configurar o SW1. O primeiro passo é a criação das vlans:

```
SW1(config)#vlan 10
SW1(config-vlan)#name GERENCIAMENTO
SW1(config-vlan)#exit
SW1(config)#vlan 20
SW1(config-vlan)#name REDE CORPORATIVA
SW1(config-vlan)#exit
SW1(config)#vlan 30
SW1(config-vlan)#name REDE CONVIDADOS
SW1(config-vlan)#exit
```

Interfaces

O WLC necessita de acesso a todas as três VLANs, portanto, precisamos de uma interface **trunk** entre o WLC e o switch. Vamos configurar essa interface:

```
W1(config)#interface GigabitEthernet 0/6
SW1(config-if)#switchport trunk encapsulation dot1q
SW1(config-if)#switchport mode trunk
SW1(config-if)#switchport trunk allowed vlan 10,20,30
```

As interfaces que conectam os access point são interfaces em modo de acesso. Vamos adicioná-las à VLAN 10, para que possam alcançar a interface de gerenciamento do WLC:

```
SW1(config)#interface range gi0/7 - 8
SW1(config-if-range)#switchport mode access
SW1(config-if-range)#switchport access vlan 10
SW1(config-if-range)#spanning-tree portfast
```

SW1 é um switch L3, então podemos usá-lo como roteador e servidor DHCP. Primeiro, habilitamos o roteamento:

```
SW1(config)#ip routing
```

E criamos as interfaces SVI, uma para cada VLAN:

```
SW1(config)#interface vlan 10
SW1(config-if)#ip address 192.168.10.254 255.255.255.0

SW1(config)#interface vlan 20
SW1(config-if)#ip address 192.168.20.254 255.255.255.0

SW1(config)#interface vlan 30
SW1(config-if)#ip address 192.168.30.254 255.255.255.0
```

Agora, criamos o pool DHCP para cada vlan (Observe os comandos abaixo, eles são os mesmos para criar um pool DHCP em um roteador):

```
SW1(config)#ip dhcp pool VLAN10
SW1(dhcp-config)#network 192.168.10.0 255.255.255.0
SW1(dhcp-config)#default-router 192.168.10.254
```

```
W1(config)#ip dhcp pool VLAN20
SW1(dhcp-config)#network 192.168.20.0 255.255.255.0
SW1(dhcp-config)#default-router 192.168.20.254
```

```
SW1(config)#ip dhcp pool VLAN30
SW1(dhcp-config)#network 192.168.30.0 255.255.255.0
SW1(dhcp-config)#default-router 192.168.30.254
```

Com isso, concluímos a configuração no SW1.

WLC

Ao ligar o WLC receberemos as seguintes mensagens de inicialização:

```
WLCNG Boot Loader Version 1.0.20 (Built on Jan 9 2014 at 19:02:44 by cisco)
Board Revision 0.0 (SN: PSZ18411Q1S, Type: AIR-CT2504-K9) (P)

Verifying boot loader integrity... OK.

OCTEON CN5230C-SCP pass 2.0, Core clock: 750 MHz, DDR clock: 330 MHz (660 Mhz
data rate)

CPU Cores: 4
```

```

DRAM: 1024 MB
Flash: 32 MB
Clearing DRAM ..... done
Network: octeth0', octeth1, octeth2, octeth3
  ' - Active interface
  E - Environment MAC address override
CF Bus 0 (IDE): OK
IDE device 0:
  - Model: 1GB CompactFlash Card Firm: CF B61FK Ser#: C361100177A10q7EIFms
  - Type: Hard Disk
  - Capacity: 977.4 MB = 0.9 GB (2001888 x 512)
Press <ESC> now to access the Boot Menu...

```

Antes de começarmos a configuração, é bom realizarmos o ‘factory reset’ (redefinição de fábrica), para isso, vamos apertar o botão ESC. Dependendo da imagem do software, aparecerá o Boot Loader Menu:

```

=====
Boot Loader Menu
=====

1. Run primary image (8.5.140.0) - Active
2. Run backup image (8.0.121.0)
3. Change active boot image
4. Clear configuration
5. Format FLASH Drive
6. Manually update images
=====
```

Vamos selecionar a opção quatro “clear configuration”, o WLC irá reiniciar:

```

Enter selection: 4
Launching...
Launching images...
init started: BusyBox v1.6.0 (2010-05-13 17:50:10 EDT) multi-call binary
starting pid 670, tty ''': '/etc/init.d/rcS'
Re-building configuration filesystem

```

```
Done.
```

```
Restarting system.
```

```
Enabling mgmt via wireless
```

```
Enabling Provisioning SSID
```

```
SSID: CiscoAirProvision, Admin Status: 1, Interface Name: management, 802.11  
Auth: WPA2-PSK, Wi-Fi Protected Access : Enabled
```

Levará aproximadamente três minutos até o dispositivo reiniciar completamente, então, receberemos a seguinte mensagem:

```
(Cisco Controller)
```

```
Welcome to the Cisco Wizard Configuration Tool
```

```
Use the '-' character to backup
```

```
Would you like to terminate autoinstall? [yes]:
```

O WLC oferece recurso de instalação automática que permite utilizar um arquivo de configuração que esteja hospedado em um servidor TFTP. Não é nossa intenção no momento, vamos pressionar ‘Enter’ para selecionar a opção padrão, que é encerrar a instalação automática.

Agora temos um assistente que fará várias perguntas. Se aparecer algo entre colchetes, basta pressionar Enter, e a opção padrão será selecionada.

Primeiro, definimos nome para o WLC, nome de usuário e senha:

```
System Name [Cisco_e0:4e:85] (31 characters max): WLC1  
Enter Administrative User Name (24 characters max): admin  
Enter Administrative Password (3 to 24 characters): *****  
Re-enter Administrative Password : *****
```

LAG

A próxima opção que aparecerá é sobre o Link Aggregation (LAG). LAG é apenas uma forma de EtherChannel, como o próprio nome indica. Porém não utilizaremos LAG nessa configuração:

```
Enable Link Aggregation (LAG) [yes][NO]:
```

Interface de gerenciamento

Agora, precisamos configurar a interface de gerenciamento:

```
Management Interface IP Address: 192.168.10.100
```

```
Management Interface Netmask: 255.255.255.0
Management Interface Default Router: 192.168.10.254
Management Interface VLAN Identifier (0 = untagged): 10
Management Interface Port Num [1 to 4]: 1
Management Interface DHCP Server IP Address: 192.168.10.254
```

A interface de gerenciamento está na Vlan 10 e conectada na interface um do WLC.

As duas próximas opções são menos intuitivas:

```
Virtual Gateway IP Address: 192.0.2.1
Multicast IP Address: 239.1.1.1
```

Vamos entender melhor essas duas opções:

- **Endereço IP do gateway virtual:** O WLC utiliza uma interface virtual para gerenciamento da rede wireless. Isso inclui DHCP relay, autenticação Web de convidados, VPN e alguns outros recursos. Este endereço só é utilizado na comunicação entre o WLC e os clientes wireless, ele deve ser um IP válido, mas não deve ser um endereço IP usado na Internet ou na própria LAN. Por isso, a rede 192.0.2.0/24 foi designada na RFC 5735 como “TEST-NET-1”, e podemos utilizá-la de forma segura.
- **Endereço IP multicast:** O WLC usa o endereço IP multicast para encaminhar o tráfego para os APs. Esse endereço também não pode estar sendo usado em nenhum outro lugar na rede. O endereço 239.1.1.1 pertence ao escopo multicast (239.0.0.0/8), portanto, é seguro usar.

Nas documentações mais antiga, era comum o uso do endereço IP 1.1.1.1 para o endereço IP do gateway virtual. A Cloudflare lançou seu serviço DNS gratuito em 2018, que usa o endereço IP 1.1.1.1, então não é mais aconselhável usar esse endereço IP.

Também é necessário configurar a mobility e RF group name:

```
Mobility/RF Group Name: Grupo_01
```

Os nomes dos grupos de mobilidade e RF são para WLCs que desejam trabalhar em conjunto. WLCs com o mesmo nome de grupo de mobilidade suportam roaming de clientes e redundância entre WLCs. Usando o mesmo nome de grupo RF, os WLCs podem fazer cálculos de gerenciamento de recursos de rádio (RRM - Radio Resource Management) para o grupo inteiro.

A próxima questão é configurar o SSID:

```
Network Name (SSID): Teste
```

Na verdade, não importa o que configuraremos aqui, já que não o usaremos de qualquer maneira.

O WLC fica entre o servidor DHCP (SW1) e o cliente wireless, ao colocar o DHCP no modo Bridge estamos deixando-o inteiramente transparente para o cliente. Vamos deixá-lo desativado:

```
Configure DHCP Bridging Mode [yes][NO]:
```

Por padrão, o WLC permite endereços IP estáticos para clientes, o que é o mais indicado:

```
Allow Static IP Addresses [YES][no]:
```

Temos a opção de configurar um servidor RADIUS:

```
Configure a RADIUS Server now? [YES][no]: no
Warning! The default WLAN security policy requires a RADIUS server.
Please see documentation for more details.
```

Não vamos configurar um servidor RADIUS no momento, porém, ele nos avisará que a política de segurança padrão requer um servidor RADIUS. É possível ignorar essa mensagem e configura-lo mais tarde.

As próximas perguntas são sobre o país e quais padrões sem fio você deseja habilitar:

```
Enter Country Code list (enter 'help' for a list of countries) [US]: BR

Enable 802.11b Network [YES][no]:
Enable 802.11a Network [YES][no]:
Enable 802.11g Network [YES][no]:
```

O Auto-RF permite que o WLC descubra e defina quais canais e qual potência usar. Vamos deixar ativado:

```
Enable Auto-RF [YES][no]:
```

Hora de configurar um servidor NTP ou então configurar data e hora de forma manual, no momento optarei pela segunda opção:

```
Configure a NTP server now? [YES][no]: no
Configure the system time now? [YES][no]: YES
Enter the date in MM/DD/YY format: 09/21/21
Enter the time in HH:MM:SS format: 13:28:00
```

Não utilizaremos IPv6, portanto, vamos pular essa parte:

```
Would you like to configure IPv6 parameters[YES][no]: no
```

Tudo configurado conforme desejado, é hora de digitar sim e aguardar o WLC reiniciar:

```
Configuration correct? If yes, system will save it and reset. [yes][NO]: yes
Configuration saved!
Resetting system with new configuration...
```

Acontecerá todo o processo de reinicialização novamente, em que precisaremos aguardar em média três minutos:

```
WLCNG Boot Loader Version 1.0.20 (Built on Jan 9 2014 at 19:02:44 by cisco)
Board Revision 0.0 (SN: PSZ18411Q1S, Type: AIR-CT2504-K9) (P)

Verifying boot loader integrity... OK.
```

[output omitted]

Quando a reinicialização terminar, aparecerá o prompt para entrarmos com login e senha:

```
(Cisco Controller)

Enter User Name (or 'Recover-Config' this one-time only to reset configuration to
factory defaults)

User: admin
Password:*****
(Cisco Controller) >
```

O comando abaixo permite verificar se o servidor DHCP no SW1 está funcionando:

```
SW1#show ip dhcp binding

Bindings from all pools not associated with VRF:

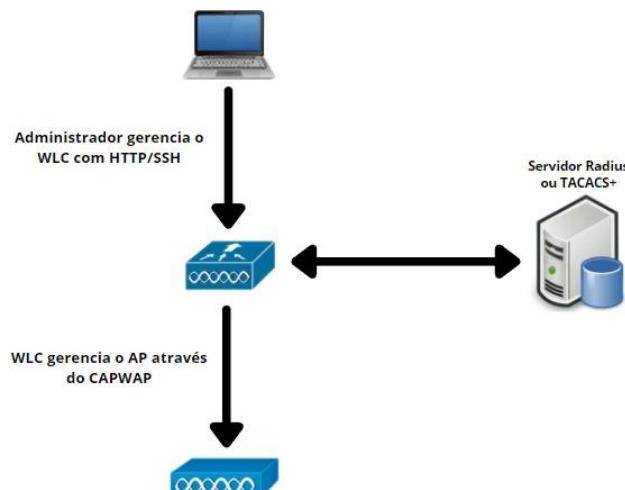
IP address          Client-ID/          Lease expiration      Type
                  Hardware address/
                  User name
192.168.10.22      01d4.6d50.fa18.64    Jan 09 2006 09:33 PM  Automatic
192.168.10.23      0188.1dfc.af4e.58    Jan 09 2006 09:34 PM  Automatic
```

SW1 possui no momento dois clientes DHCP.

2.8 Describe AP and WLC management access connections (Telnet, SSH, HTTP, HTTPS, console, and TACACS+/RADIUS)

Após APs lightweight se registrarem na WLC, a gerencia (parte inteligente) dos APs passam a ser realizada pelo WLC, sequer precisaremos acessá-los diretamente após eles terem se registrado na controladora.

Os WLCs são gerenciados principalmente via HTTPS e SSH, mas também aceitam acesso por meio do Tacacs+/ Radius. Podemos limitar através por meio de ACLs as redes que podem gerenciar o WLC.



Retornando ao exemplo anterior, vamos acessar a WLC que configuramos no último tópico utilizando um navegador e o protocolo HTTP. O endereço que vamos utilizar é o endereço da interface de gerenciamento: <http://192.168.10.100>, aparecerá a seguinte tela:



Clicando em 'login' aparecerá uma tela solicitando usuário e senha que configuramos anteriormente:

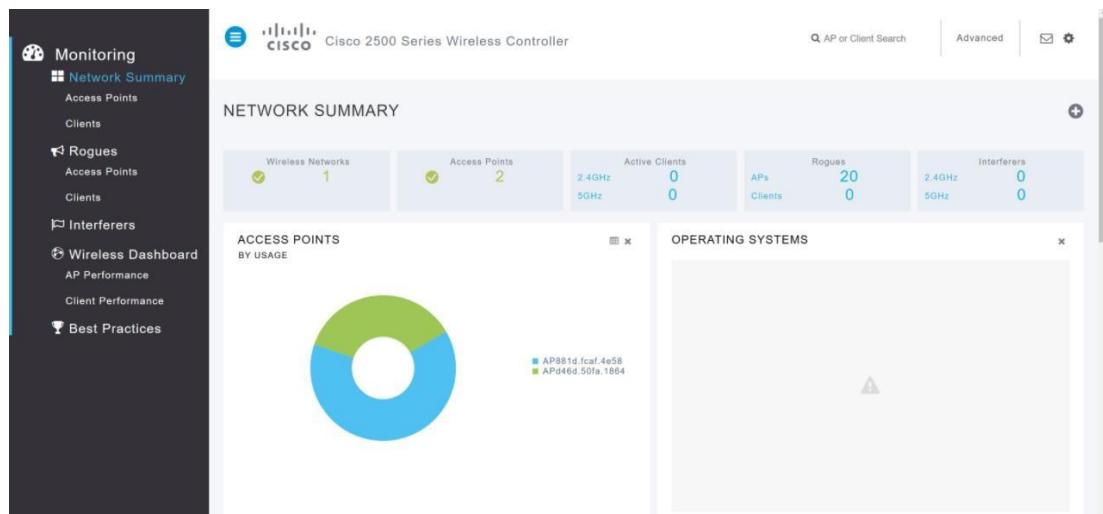
Sign in

<http://192.168.10.100>
Your connection to this site is not private

Username:

Password:

A próxima tela é o dashboard, o painel de monitoramento:



Este painel fornece apenas uma visão básica. Nele, você verá as redes sem fio, access points, clientes ativos, etc. Por exemplo, aqui podemos encontrar os APs:

AP Name	Clients	Usage	Uptime	Channels	Coverage	Interference	Requests	MAC Address
APd46d.50fa.1864	0	891.6 MB	22 Hours 23 Minu...	11	11	0	11	d4:6d:50:fa:18:64
AP881d.fcaf.4e58	0	1.5 GB	22 Hours 23 Minu...	27	1	0	27	88:1d:fcaf:4e:58

Observe que os dois APs realmente se associaram ao WLC. A aba ‘best practices’ (melhores práticas) oferece uma boa visão geral dos itens que devemos configurar:

Category	Item	Status
INFRASTRUCTURE	Application Visibility	<input type="radio"/>
	Disable Aironet IE	<input type="radio"/>
	Disable Internal DHCP	<input checked="" type="radio"/>
	More Optimizations...	<input checked="" type="radio"/>
SECURITY	802.1x on AP	<input type="radio"/>
	CPU ACLs	<input type="radio"/>
	Client Exclusion	<input checked="" type="radio"/>
	More Optimizations...	<input checked="" type="radio"/>
RF MANAGEMENT	Auto Coverage Hole Detection	<input checked="" type="radio"/>
	Auto Dynamic Channel Assignment	<input checked="" type="radio"/>

O painel de monitoramento oferece uma visão geral, mas se quisermos configurar alguma coisa, temos que usar o modo avançado (advanced), ele pode ser encontrado no canto superior direito:

Esta é tela de abertura do modo avançado:

The screenshot shows the Cisco Wireless LAN Controller (WLC) management interface. The top navigation bar includes links for Summary, MONITOR, WLANs, CONTROLLER, WIRELESS, SECURITY, MANAGEMENT, COMMANDS, HELP, and FEEDBACK. On the left, a sidebar menu lists options like Monitor, Summary, Access Points, Cisco Cleanair, Statistics, CDP, Rogues, Clients, Sleeping Clients, Multicast, Applications, and Local Profiling. The main content area displays a summary of the controller's configuration and performance. It includes sections for 'Controller Summary' (Management IP Address: 192.168.10.100, Software Version: 8.5.140.0), 'Access Point Summary' (Total: 2, Up: 2, Down: 0), and 'Rogue Summary' (Active Rogue APs: 22). A 'Session Timeout' section shows no active sessions. Below these are 'Top WLANs' and 'Most Recent Traps' logs.

Observe a quantidade de opções que temos.

2.9 Configure the components of a wireless LAN access for client connectivity using GUI only such as WLAN creation, security settings, QoS profiles, and advanced WLAN settings

Neste tópico vamos configurar alguns parâmetros utilizando a interface gráfica da controladora, para quem não está familiarizado com a expressão, GUI é uma abreviação de ‘Graphical User Interface’ que podemos traduzir como “interface gráfica do usuário”.

Vamos acessar a GUI da WLC que configuramos no penúltimo tópico utilizando um navegador e o protocolo HTTP. O endereço que vamos utilizar é o endereço da interface de gerenciamento: <http://192.168.10.100>. Ao acessarmos, aparecerá a seguinte tela:



Clicando em ‘login’ aparecerá uma tela pedindo o usuário e senha que configuramos anteriormente:



A próxima tela é o dashboard, o painel de monitoramento:

Agora, vamos escolher a opção advanced:

E chegaremos na tela abaixo:

Criação da Wlan

Primeiro, vamos criar uma Wlan, clicando na opção ‘Wlan’ e depois em ‘Create New’

Na tela que irá abrir, informaremos o tipo, o nome e a SSID:

Na próxima tela, precisamos habilita-la e escolher a vlan que ela pertencerá.

Secutiry Settings

Vamos agora para as configurações de segurança. Para acessa-la, basta clicar em “Security” que está ao lado da nossa aba atual: ‘General’. A seguinte tela irá aparecer:

Nessa tela podemos escolher o método de criptografia e autenticação e a senha.

QoS

Na tela ao lado, temos as opções de QoS:

Não se preocupe se você não sabe o que quer dizer cada um desses termos, mais para frente teremos um tópico dedicado somente ao QoS.

Advanced WLAN settings

E por último, temos a tela de configuração avançada da Wlan, com opções variadas.

O propósito desde tópico é apenas demonstrar as telas para que você tenha uma noção do poder da controladora WLC, a maioria dos conceitos para aplicação nessas telas já foi passado em tópicos anteriores.

Com isso, concluímos o segundo grande grupo que a Cisco divide o CCNA 200-301, a parte de acesso a rede “Network Access”. Abaixo teremos algumas perguntas e respostas para fixação do conteúdo.

Exercícios:

1. Quais das etapas a seguir são necessárias para adicionar uma nova VLAN a um switch já instalado na rede? (Selecione 03 respostas)
 - a) Criar a VLAN.
 - b) Nomear a VLAN.
 - c) Configurar um endereço IP para a VLAN.
 - d) Adicionar as portas desejadas à nova VLAN.
 - e) Adicionar a VLAN ao domínio VTP.
2. Depois de conectar um computador a uma porta disponível de um switch, você percebe que o computador não consegue acessar nenhum recurso da LAN. Os demais computadores conectados ao switch estão funcionando normalmente. Qual é a causa mais provável desse problema?
 - a) O roteador não tem uma entrada na tabela de roteamento para o novo host
 - b) A porta do switch está atribuída à VLAN incorreta
 - c) O endereço MAC do computador está configurado incorretamente
 - d) Uma instância STP para o novo computador não foi inicializada
 - e) O switch não possui o endereço MAC do computador na tabela CAM.
3. Um administrador de redes precisa verificar se a interface fa0/5 esta atribuída a VLAN de Marketing. Qual comando realizará esta tarefa?
 - a) Show vlan
 - b) Show mac-address-table
 - c) Show vtp status
 - d) show spanning-tree root
 - e) show ip interface brief
4. Uma nova interface trunk foi adicionada a um switch, por default quais as vlans são permitidas nessa interface trunk?
 - a) Nenhuma vlan
 - b) Apenas as vlans que forem permitidas na criação da interface trunk
 - c) As vlans 1-64
 - d) Todas as vlans
 - e) Todas as vlans exceto a vlan 1
5. Quais os protocolos de encapsulamento são possíveis de configurar em uma interface trunk? (Escolha duas opções)
 - a) VTP
 - b) ISL
 - c) CDP
 - d) 802.1Q
 - e) 802.1p
 - f) LLC
 - g) IETF
6. Um administrador de redes precisa configurar um link 802.1Q entre dois switches. Quais os comandos deverão ser usados para completar essa tarefa? (Escolha duas opções)
 - a) Switch(vlan)# mode trunk
 - b) Switch(config)# switchport access mode trunk
 - c) Switch(config-if)# switchport mode trunk
 - d) Switch(config-if)# switchport trunk encapsulation dot1q
 - e) Switch(config)# switchport access mode 1
 - f) Switch(vlan)# trunk encapsulation dot1q
7. CDP está sendo executado entre dois dispositivos. Quais as informações são fornecidas pelo protocolo? (escolha três opções)
 - a) Device Identifiers
 - b) Capabilities list
 - c) Platform
 - d) Route identifier
8. Em uma interface trunk, quais dos seguintes modos são válidos?
 - a) Blocking
 - b) Auto
 - c) Desirable
 - d) On
 - e) Transparent
 - f) Learning

9. Qual dos protocolos abaixo opera na camada 2 do modelo OSI e é usado para manter uma rede sem loop?
- a) RIP
 - b) STP
 - c) IGRP
 - d) CDP
 - e) VTP
10. Por padrão, qual dos seguintes fatores determina o custo do caminho do spanning tree?
- a) O custo do link individual com base na latência
 - b) A soma dos custos com base na largura de banda
 - c) A contagem total de saltos
 - d) É determinado dinamicamente com base no tráfego
11. Qual o propósito do spanning-tree em uma LAN?
- a) Fornecer um mecanismo de monitoramento para os switches da rede.
 - b) Gerenciar VLANs em vários switches.
 - c) Evitar loops camada 02.
 - d) Segmentar a rede em vários domínios de colisão.
 - e) Suitar loops de roteamento em redes.
12. Quais dois dos seguintes valores o STP leva em consideração para eleger a root bridge? (escolha duas opções)
- a) A versão do BDPU
 - b) Bridge ID
 - c) Número do update do Spanning-tree
 - d) Bridge priority
 - e) Numero da Vlan
13. Quais das três opções abaixo são status das portas no spanning-tree?
- a) Learning
 - b) Spanning
 - c) Listening
 - d) Forwarding
 - e) Initializing
 - f) Filtering
 - g) Permitting
14. Em quais estados do Spanning-Tree uma porta aprende endereços MAC? (escolha duas opções)
- a) Blocking
 - b) Listening
 - c) Forwarding
 - d) Learning
 - e) Relaying
15. Qual parâmetro pode ser diferente nas configurações de uma interface EtherChannel?
- a) Velocidade
 - b) Configurações de negociação do DTP
 - c) Modo de encapsulamento
 - d) Duplex
16. Qual opção é o protocolo EtherChannel padrão da IEEE?
- a) LACP
 - b) PAGP
 - c) CDP
 - d) DTP
17. Qual é a vantagem de utilizar Cisco Wireless LAN Controller?
- a) O gerenciamento central dos APs requer configurações mais complexas.
 - b) SSIDs únicos não podem usar o mesmo método de autenticação.
 - c) Suporta APs autônomos e lightweight.
 - d) Elimina a necessidade de configurar cada access point individualmente.

Resposta: 1) a, b, d – 2) b - 3) a – 4) d – 5) b, d – 6) c, d – 7)a, b, c – 8) b, c, d – 9) b – 10) b, 11) c – 12)b, e – 13)a, c, d – 14)c, d – 15) b – 16) a – 17) d

3. IP Connectivity

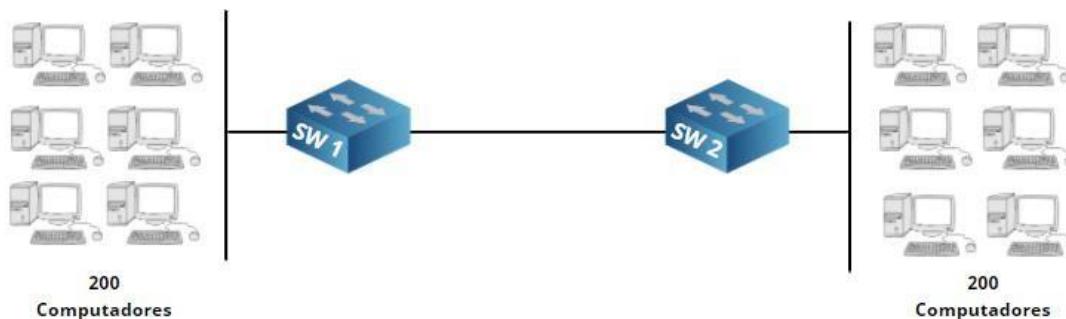
Introdução

Antes de começar, precisamos definir melhor o que é um roteador e o que é roteamento. Sabemos que switches encaminham frames dentro de uma mesma rede e que um roteador “encaminha” pacotes para redes diferentes. Mas o que exatamente é esse encaminhar pacotes?

Os switches “encaminham frames” com base no endereço MAC dos dispositivos. A única preocupação do switch é definir para qual interface encaminhará o quadro Ethernet que acabou de chegar em uma interface, ele faz isso, observando a tabela de endereço MAC que ele criou. Todos esses eventos ocorrem na camada de elance de dados (camada 02 do modelo OSI).

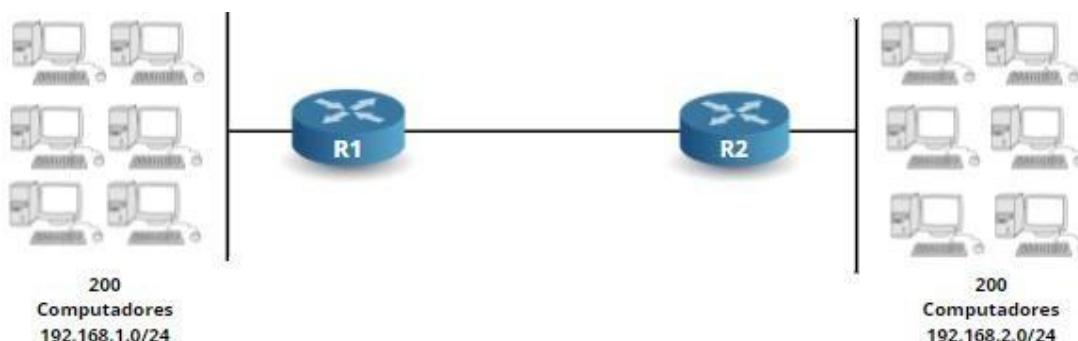
Os roteadores agem de forma semelhante, porém, eles não olham os frames de camada dois, eles examinam os pacotes IP e, como você deve se lembrar, o IP está na camada de rede (camada 3). Resumindo, os roteadores examinam o endereço IP de destino em um pacote, e o envia pela interface correta.

Uma pergunta válida, e que provavelmente está na sua cabeça é: Qual é a grande diferença até aqui? Por que não usamos endereços MAC em todos os lugares e realizamos tudo isso só com os switches? Por que precisamos de endereços IP e rotas? Essas perguntas merecem uma boa resposta... Para isso vou recorrer mais uma vez a ajuda de imagens:



Acima, temos dois switches e em cada um deles estão conectados 200 computadores. Imagine que todos os 400 computadores começem a se comunicar, cada switch terá que aprender 400 endereços MAC, pois terão que saber tanto os endereços MAC dos computadores do lado esquerdo e direito.

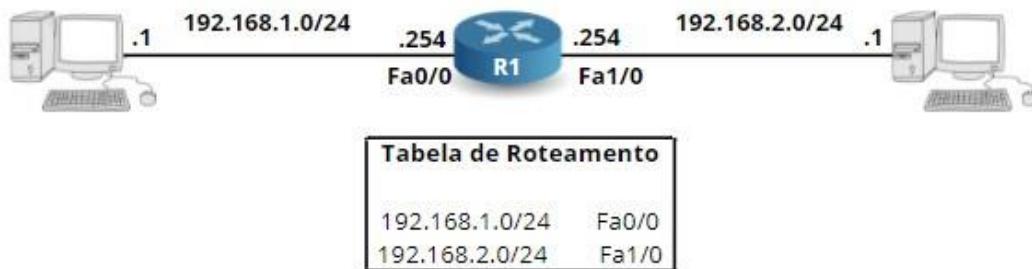
Agora pense em uma rede realmente grande, como por exemplo a Internet. Existem milhões de dispositivos! Seria possível ter milhões de entradas em uma tabela de endereços MAC? De jeito nenhum! O problema com a comutação é que ela não é escalonável; não temos nenhuma hierarquia, apenas endereços planos de 48 bits. Vejamos o mesmo exemplo, mas agora estamos usando roteadores.



Nesse cenário, temos 200 computadores conectados ao roteador R1, sendo que todos estão na rede 192.168.1.0/24, temos mais 200 computadores conectados ao R2, todos na rede 192.168.2.0/24. Os roteadores “encaminham pacotes” com base no endereçamento IP. Neste exemplo, o roteador R1 só precisa saber que a rede 192.168.2.0/24 está atrás do roteador R2, e o roteador R2 só necessita saber que a rede 192.168.1.0/24 está atrás do roteador R1.

Em vez de ter uma tabela com 400 endereços MAC, agora precisamos apenas de uma única entrada em cada roteador para as redes conectadas em outros roteadores. Os switches usam tabelas de endereços mac para encaminhar frames Ethernet e os roteadores **usam uma tabela de roteamento para saber para onde encaminhar os pacotes IP**.

Quando compramos um roteador novo e tiramos da caixa, ele automaticamente criará uma tabela de roteamento, mas as únicas informações que ele sabe, são as interfaces **diretamente conectadas**. Vamos começar com um exemplo simples:



Acima temos um roteador e dois computadores:

- O PC1 possui o endereço IP 192.168.1.1 com máscara de rede 255.255.255.0 e como gateway padrão o endereço IP 192.168.1.254.
- PC2 possui o endereço IP 192.168.2.2 com máscara de rede 255.255.255.0 e como gateway padrão o endereço IP 192.168.2.254.
- O roteador está configurado com o IP 192.168.1.254 com máscara de rede 255.255.255.0 na interface FastEthernet 0/0 e o endereço IP 192.168.2.254 com máscara de rede 255.255.255.0 na interface FastEthernet 1/0.

Quando o PC1 quiser enviar algo para PC2, acontecerá o passo a passo abaixo:

1. PC1 deseja enviar um pacote IP para o endereço 192.168.2.2.
2. Antes de enviar, o PC1 verificará seu próprio endereço IP e a máscara de sub-rede para verificar se o dispositivo está na mesma rede. Ele chegará à conclusão que o endereço 192.168.2.2 está em outra sub-rede. Como resultado, ele encaminhará o pacote IP para seu gateway padrão.
3. O roteador receberá o pacote IP, verificará o endereço IP de destino e em seguida pesquisará a sua tabela de roteamento. O endereço IP 192.168.2.2 corresponde à entrada 192.168.2.0/24, então, o roteador encaminhará o pacote IP para a interface FastEthernet 1/0.
4. PC2 receberá o pacote IP!

Vamos configurar este cenário em um roteador real, mas antes vamos ver a configuração dos computadores, observe às máscaras de sub-rede e o default gateway de cada computador:

```
C:\Documents and Settings\PC1>ipconfig
Windows IP Configuration

Ethernet adapter Local Area Connection:

  Connection-specific DNS Suffix . :
  IP Address . . . . . : 192.168.1.1
  Subnet Mask . . . . . : 255.255.255.0
  Default Gateway . . . . . : 192.168.1.254
```

```
C:\Documents and Settings\PC2>ipconfig
```

```

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix . :

    IP Address. . . . . : 192.168.2.2
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.2.254

```

Agora, vamos configurar os endereços IPs nas interfaces do roteador:

```

R1(config)#interface fastEthernet 0/0
R1(config-if)#no shutdown
R1(config-if)#ip address 192.168.1.254 255.255.255.0
R1(config-if)#exit
R1(config)#interface FastEthernet 1/0
R1(config-if)#no shutdown
Router(config-if)#ip address 192.168.2.254 255.255.255.0

```

Tudo configurado, é hora de verificarmos a tabela de roteamento com o comando ‘show ip route’:

```

R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level
        ia - IS-IS inter area, * - candidate default, U - per-user static
        o - ODR, P - periodic downloaded static route
Gateway of last resort is not set
C    192.168.1.0/24 is directly connected, FastEthernet0/0
C    192.168.2.0/24 is directly connected, FastEthernet1/0

```

Nas duas últimas linhas o roteador nos informa que essas duas redes estão diretamente conectadas. Vamos testar um ping do computador 1 para o computador 2:

```

C:\Users\PC1>ping 192.168.2.2

Pinging 192.168.2.2 with 32 bytes of data:
Reply from 192.168.2.2: bytes=32 time=1ms TTL=128
Reply from 192.168.2.2: bytes=32 time=1ms TTL=128
Reply from 192.168.2.2: bytes=32 time=1ms TTL=128

```

```

Reply from 192.168.2.2: bytes=32 time=1ms TTL=128
Ping statistics for 192.168.2.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

```

Tivemos sucesso no teste de ping! Parabéns, você acaba de rotear com sucesso seu primeiro pacote IP!

Quando acessamos um site ou servidor na Internet, passamos por muitos roteadores até chegar ao destino. É possível ver por quais roteadores os pacotes IP estão passando com o comando **traceroute**. Observe a quantidade de roteadores que o pacote IP passa para chegar até o site www.cisco.com. No exemplo abaixo, usaremos um computador rodando Windows, nele o comando é um pouco diferente do roteador, mas a função é a mesma:

```

C:\Users\Computer>tracert www.cisco.com

Tracing route to e144.dscb.akamaiedge.net [95.100.128.170]
over a maximum of 30 hops:

 1    <1 ms      <1 ms      <1 ms  192.168.154.2
 2    <1 ms      <1 ms      <1 ms  192.168.81.254
 3      9 ms       7 ms       9 ms  10.224.124.1
 4        8 ms        7 ms      10 ms  tb-rc0001-cr101-irb-201.core.as9143.net
[213.51.150.129]
 5      31 ms       10 ms      13 ms  asd-lc0006-cr101-ae5-0.core.as9143.net
[213.51.158.18]
 6      11 ms       12 ms      11 ms  ae1.ams10.ip4.tinet.net [77.67.64.61]
 7      11 ms       14 ms      14 ms  r22.amstnl02.nl.bb.gin.ntt.net [195.69.144.36]
 8        14 ms        15 ms      11 ms  ae-2.r03.amstnl02.nl.bb.gin.ntt.net
[129.250.2.211]
 9      14 ms       11 ms      11 ms  81.20.67.150
10      12 ms       11 ms      11 ms  95.100.128.170

Trace complete.

```

Para chegarmos até o site da Cisco passamos por 10 roteadores. Observe que o computador mostrou não só os endereços IPs como ‘resolveu’ os nomes dos roteadores.

Configuração básica de roteadores Cisco

Essa introdução aos roteadores é essencial, aqui falarei sobre os modelos, primeiro boot, interfaces, etc.

Integrated Services Routers

A Cisco possui vários modelos de roteadores que atendem os mais variados tipos de clientes, desde roteadores para pequenas empresas com apenas alguns usuários, até roteadores gigantescos como os usados em data centers.

Você que está começando agora no universo Cisco, em princípio trabalhará com alguns roteadores menores, entre eles os ‘Integrated Services Routers’ (*roteadores de serviços integrados*) que possuem funções que vão além de simplesmente rotear pacotes, eles também oferecem alguns outros serviços como wireless, Voice over IP, etc.

Abaixo, foto dos roteadores das séries 1800, 2800 e 3800:



Apesar desses roteadores não serem os modelos mais novos, ainda são muito populares. Vamos examinar um roteador da série 2800 mais de perto:



Este roteador possui:

- 1x porta USB.
- 2 interfaces FastEthernet.
- 1x porta de console.
- 1x porta AUX.
- 1 slot de memória Flash.
- slots WIC.

Os slots WIC podem ser usados para adicionar alguns tipos de placas. Por exemplo, modems DSL, interface serial, access points, e assim por diante. Abaixo um módulo WIC para uma interface serial:

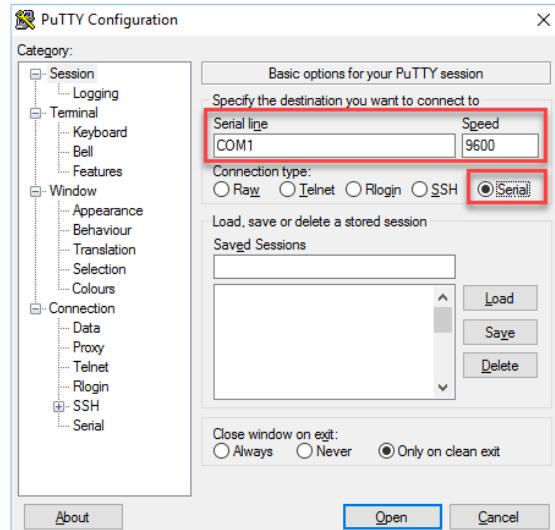


Nos exemplos abaixo, usaremos essa interface serial.

Configuração

Primeiro, precisamos conectar o cabo console ao roteador e em seguida através do Putty acessa-lo:

Certifique-se de selecionar “Serial Line” e definir a velocidade em 9600. A porta COM provavelmente será diferente, especialmente se você estiver usando um adaptador USB x serial. Verifique o número correto no gerenciador de dispositivos do Windows.



Primeiro Boot

Assim que estiver conectado à porta console e ligar o dispositivo, aparecerá a seguinte tela:

System Bootstrap, Version 12.4(13r)T, RELEASE SOFTWARE (fc1)

Technical Support: <http://www.cisco.com/techsupport>

Copyright (c) 2006 by cisco Systems, Inc.

Initializing memory for ECC

...

c2811 platform with 786432 Kbytes of main memory

Main memory is configured to 64 bit mode with ECC enabled

Readonly ROMMON initialized

program load complete, entry point: 0x8000f000, size: 0xcb80

program load complete, entry point: 0x8000f000, size: 0xcb80

program load complete, entry point: 0x8000f000, size: 0x40c7

Quando o roteador é ligado, ele primeiro inicializa o ROMMON. ROMMON é semelhante a BIOS de um computador, ele permite que o roteador execute algumas funções básicas, como carregar o sistema operacional:

```
Self decompressing the image :  
#####
#####  
#####
#####  
#####
#####  
#####
#####  
#####
##### [OK]
```

Assim que o sistema operacional for descompactado, aparecerá as informações de direitos autorais da Cisco:

```
Smart Init is disabled. IOMEM set to: 5
Using iomem percentage: 5
    Restricted Rights Legend
Use, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
Rights clause at FAR sec. 52.227-19 and subparagraph
(c) (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134-1706
Cisco IOS Software, 2800 Software (C2800NM-ADVENTERPRISEK9-M), Version
15.1(4)M10, RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2015 by Cisco Systems, Inc.

Compiled Tue 24-Mar-15 09:00 by prod_rel_team
This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.
A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wlc/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to
export@cisco.com.
```

Em seguida, aparecerá algumas informações básicas sobre o roteador:

```
Installed image archive  
  
Cisco 2811 (revision 1.0) with 747520K/38912K bytes of memory.  
  
Processor board ID FTX1145A0XN  
  
2 FastEthernet interfaces  
  
2 Serial(sync/async) interfaces  
  
1 Virtual Private Network (VPN) Module  
  
DRAM configuration is 64 bits wide with parity enabled.  
  
239K bytes of non-volatile configuration memory.  
  
3906504K bytes of ATA CompactFlash (Read/Write)
```

Acima, aparece a informação que este é um roteador Cisco da linha 2811, que possui 2 interfaces FastEthernet e 2 interfaces Serial. Também nos informa a quantidade de memória RAM que o roteador possui e quanto grande é a memória flash.

Finalmente, o roteador nos pergunta se queremos iniciar a caixa de diálogo de configuração inicial:

```
--- System Configuration Dialog ---  
  
Would you like to enter the initial configuration dialog? [yes/no]: no
```

Vamos digitar ‘não’ e seguir a configuração por nossa conta. A seguinte mensagem aparecerá e cairemos no modo de usuário:

```
Press RETURN to get started!  
  
Router>
```

Apagando a configuração inicial

Em muitas situações será preciso remover as configurações da configuração inicial, assim, conseguimos garantir que ao dispositivo está zerado e não há nenhum resquício de configuração antiga.

Primeiro entramos no modo enable:

```
Router>enable
```

Em seguida, podemos apagar a configuração:

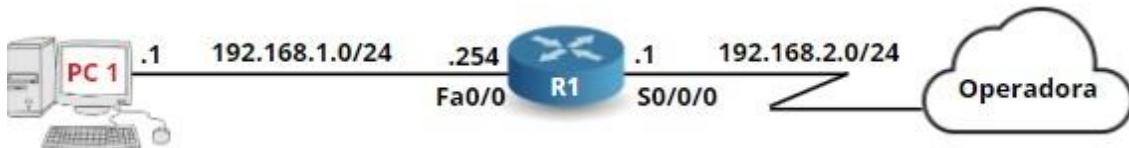
```
Router#erase startup-config  
  
Erasing the nvram filesystem will remove all configuration files! Continue?  
[confirm]  
  
[OK]  
  
Erase of nvram: complete
```

Por fim, basta reiniciar o roteador:

```
Router#reload  
  
System configuration has been modified. Save? [yes/no]: no  
  
Proceed with reload? [confirm]
```

Interfaces

Cada interface de um roteador possui um endereço IP. Vamos configurar o roteador para o cenário abaixo:



Na topologia acima, R1 está conectado ao PC1 na interface FastEthernet 0/0 com endereço IP 192.168.1.254 e a máscara de sub-rede é 255.255.255.0 (/24). Ele também está conectado a uma ‘operadora’ através da interface Serial 0/0/0.

Vamos verificar todas as interfaces do roteador:

Router#show ip interface brief					
Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	unassigned	YES	unset	administratively down	down
FastEthernet0/1	unassigned	YES	unset	administratively down	down
Serial0/0/0	unassigned	YES	unset	administratively down	down
Serial0/1/0	unassigned	YES	unset	administratively down	down

Além de informar que o roteador possui quatro interfaces, com o comando acima também é possível descobrir:

- **IP-Address:** Informa se a interface possui um endereço IP.
- **OK:** Informa se a interface está funcionando corretamente.
- **Method:** Mostra como o endereço IP foi adquirido. Por exemplo, podemos configurar um endereço IP manualmente ou por meio de DHCP.
- **Status:** Informa se a interface está ativa ou não.
- **Protocol:** Mostra se o protocolo que a interface está usando está funcionando ou não.

Todas as interfaces do roteador vêm desabilitadas por padrão. ‘Administratively down’ significa que a interface está com o comando ‘shutdown’ aplicado.

Vamos dar uma olhada de forma mais aprofundada em uma dessas interfaces:

```
Router#show interfaces FastEthernet 0/0

FastEthernet0/0 is administratively down, line protocol is down

Hardware is MV96340 Ethernet, address is 001d.a18b.36d0 (bia 001d.a18b.36d0)
MTU 1500 bytes, BW 100000 Kbit/sec, DLY 100 usec,
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation ARPA, loopback not set
Keepalive set (10 sec)
Auto-duplex, Auto Speed, 100BaseTX/FX
ARP type: ARPA, ARP Timeout 04:00:00
Last input never, output never, output hang never
```

```

Last clearing of "show interface" counters never

Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0

Queueing strategy: fifo

Output queue: 0/40 (size/max)

5 minute input rate 0 bits/sec, 0 packets/sec

5 minute output rate 0 bits/sec, 0 packets/sec

    0 packets input, 0 bytes

    Received 0 broadcasts (0 IP multicasts)

    0 runts, 0 giants, 0 throttles

    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored

    0 watchdog

    0 input packets with dribble condition detected

    0 packets output, 0 bytes, 0 underruns

    0 output errors, 0 collisions, 0 interface resets

    0 unknown protocol drops

    0 babbles, 0 late collision, 0 deferred

    29 lost carrier, 0 no carrier

    0 output buffer failures, 0 output buffers swapped out

```

Podemos ver que a interface FastEthernet 0/0 não está em uso. Vamos configura-la.

Primeiro, vamos entrar no modo de configuração global:

```
Router#configure terminal
```

Em seguida, vamos até a interface configurar o endereço IP e ativa-la:

```

Router(config)#interface FastEthernet 0/0
Router(config-if)#ip address 192.168.1.254 255.255.255.0
Router(config-if)#no shutdown

```

A interface agora está ativa e funcional. Porém, também temos uma interface serial. Uma interface serial pode exigir que configuremos a taxa de clock, se você conectar dois roteadores back-to-back com links seriais, o lado do DCE necessitará de uma taxa de clock. Vamos ver se é esse o caso:

```

Router#show controllers Serial 0/0/0
CD2430 Slot 1, Port 0, Controller 0, Channel 0, Revision 19
Channel mode is synchronous serial
idb 0x498B25D8, buffer size 1524, V.35 DCE cable

```

O router que estamos configurando é o DCE, logo, precisamos definir o ‘clock rate’:

```
Router(config)#interface Serial0/0/0
Router(config-if)#clock rate 12800
```

Podemos verificar se o ‘clock rate’ está configurado com o comando abaixo:

```
Router#show controllers Serial 0/0/0
CD2430 Slot 1, Port 0, Controller 0, Channel 0, Revision 19
Channel mode is synchronous serial
idb 0x498B25D8, buffer size 1524, V.35 DCE cable, clockrate 128000
```

Agora podemos definir o endereço IP e habilitar a interface:

```
Router(config)#interface Serial 0/0/0
Router(config-if)#no shutdown
Router(config-if)#ip address 192.168.12.1 255.255.255.0
```

Se configuramos tudo certo, o roteador deve ter nesse momento duas interfaces ativas:

```
Router#show ip interface brief
Interface                                IP-Address      OK? Method Status
Protocol

FastEthernet0/0          192.168.1.254    YES manual up
FastEthernet0/1            unassigned       YES unset administratively down down
Serial0/0/0           192.168.12.1    YES manual up
Serial0/1/0              unassigned       YES unset administratively down down
```

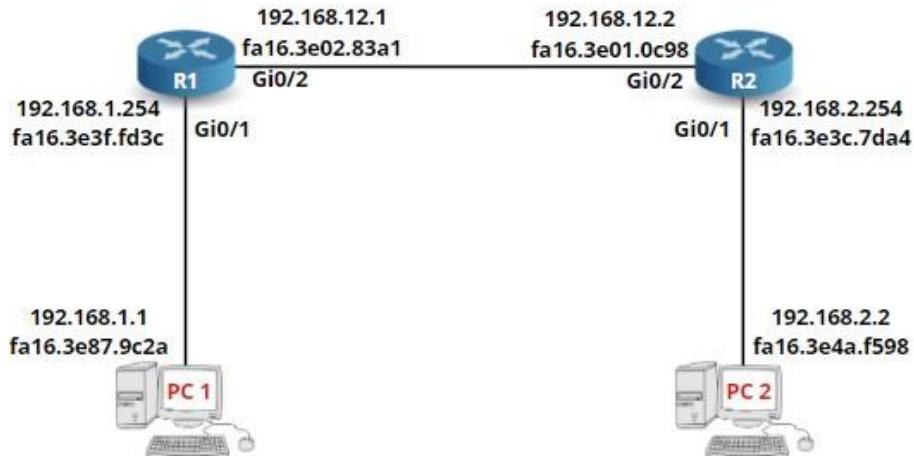
Um comando alternativo, mas que nos oferece uma saída similar é o “show protocols”:

```
Router#show protocols
Global values:
  Internet Protocol routing is enabled
FastEthernet0/0 is up, line protocol is up
    Internet address is 192.168.1.254/24
  FastEthernet0/1 is administratively down, line protocol is down
Serial0/0/0 is up, line protocol is up
    Internet address is 192.168.12.1/24
  Serial0/1/0 is administratively down, line protocol is down
```

3.1 Interpret the components of routing table

Depois de aprendermos os comandos básicos de configuração de um roteador, é hora de aprendermos os componentes que formam uma tabela de roteamento. Mas antes, você precisa entender como é formado o processo de roteamento IP.

O encaminhamento de pacotes IP por roteadores é chamado de ‘Ip routing’ (*roteamento IP*). Por enquanto não estamos falando do “aprendizado” de rotas por meio de protocolos de roteamento estáticos ou dinâmicos, mas das etapas que os roteadores devem seguir ao encaminhar um pacote IP de uma interface para outra. Vamos entender como esse processo acontece, para isso, usaremos a seguinte topologia:



Na topologia acima, temos dois computadores e dois roteadores. Mostrarei o processo passo a passo do PC1 enviando um pacote para o PC2 que terá de passar (ser roteado) pelo R1 e R2.

IP Routing Process

Vamos entender esse processo olhando todos os dispositivos:

➤ PC1

O PC1 cria um pacote IP com seu próprio endereço (192.168.1.1) como origem e PC2 (192.168.2.2) como destino. A *primeira pergunta* que PC1 fará a si mesmo é:

O destino é *local ou remoto*?

Ele responderá essa pergunta observando seu endereço IP, sua máscara de sub-rede e o endereço IP de destino:

```
C:\Users\PC1>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet 1:

  Connection-specific DNS Suffix . : luiz.local

  Link-local IPv6 Address . . . . . : fe80::88fd:962a:44d6:3a1f%4

  IPv4 Address. . . . . : 192.168.1.1
                            Subnet Mask . . . . . : 255.255.255.0
                            Default Gateway . . . . . : 192.168.1.254
```

PC1 está na rede 192.168.1.0/24, portanto, todos os endereços IPs no intervalo 192.168.1.1 a 254 são endereços locais. O destino (192.168.2.2) está fora da sub-rede local, o que significa que ele terá de usar o gateway padrão para chegar até ele.

PC1 construirá um quadro Ethernet, contendo como endereço de origem seu próprio endereço MAC e fará a si mesmo a *segunda pergunta*: Eu sei o endereço MAC de destino do gateway padrão?

Então, ele verificará sua tabela ARP para encontrar a resposta:

Internet Address	Physical Address	Type
192.168.1.254	fa-16-3e-3f-fd-3c	dynamic
192.168.1.255	ff-ff-ff-ff-ff-ff	static
224.0.0.22	01-00-5e-00-00-16	static
224.0.0.251	01-00-5e-00-00-fb	static
224.0.0.252	01-00-5e-00-00-fc	static
239.255.255.250	01-00-5e-7f-ff-fa	static

PC1 tem uma entrada ARP para 192.168.1.254. Caso não tivesse, ele teria enviado uma solicitação ARP em broadcast. Agora temos um frame Ethernet que carrega um pacote IP com os seguintes endereços:



O frame está a caminho do roteador R1.

➤ R1

Quando o frame chegar ao R1, uma série de ações será tomada:

A primeira coisa será verificar se o FCS (Frame Check Sequence) do frame Ethernet está correto ou não:



Caso o FCS esteja incorreto, o quadro será descartado imediatamente. Não há recuperação de erros para Ethernet, isso é feito por protocolos nas camadas superiores, como o protocolo TCP na camada de transporte.

Se o FCS estiver correto, o quadro será processado pelo roteador se:

- O endereço MAC de destino for o endereço da interface do roteador.
- O endereço MAC de destino é um endereço de broadcast da sub-rede à qual a interface do roteador está conectada.
- O endereço MAC de destino é um endereço multicast que o roteador ‘escuta’.

No caso em tela, o endereço MAC de destino corresponde ao endereço MAC da interface GigabitEthernet 0/1 do R1, portanto, o roteador processará o frame.

O roteador descapsulará (extrairá) o pacote IP do quadro Ethernet, que então será descartado:



O roteador agora examinará o pacote IP e a primeira coisa que fará é verificar se a soma de verificação do cabeçalho (header checksum) está OK:

Version	Header Length	Type of Service	Total Length	
Identification			IP Flags	Fragment Offset
Time to Live 255	Protocol	Header Checksum		
Source: 192.168.1.1				
Destination: 192.168.2.2				
IP Option				

Se a soma de verificação do cabeçalho não estiver correta, o pacote IP será descartado imediatamente. Também não há recuperação de erros na camada de rede, essa recuperação é realizada nas camadas superiores. Se a soma de verificação do cabeçalho estiver correta, o roteador examinará o endereço IP de destino:

Version	Header Length	Type of Service	Total Length	
Identification			IP Flags	Fragment Offset
Time to Live 255	Protocol	Header Checksum		
Source: 192.168.1.1				
Destination: 192.168.2.2				
IP Option				

R1 irá verificar sua tabela de roteamento para ver se há correspondência, podemos acompanhar com o comando ‘show ip route’:

```
R1#show ip route

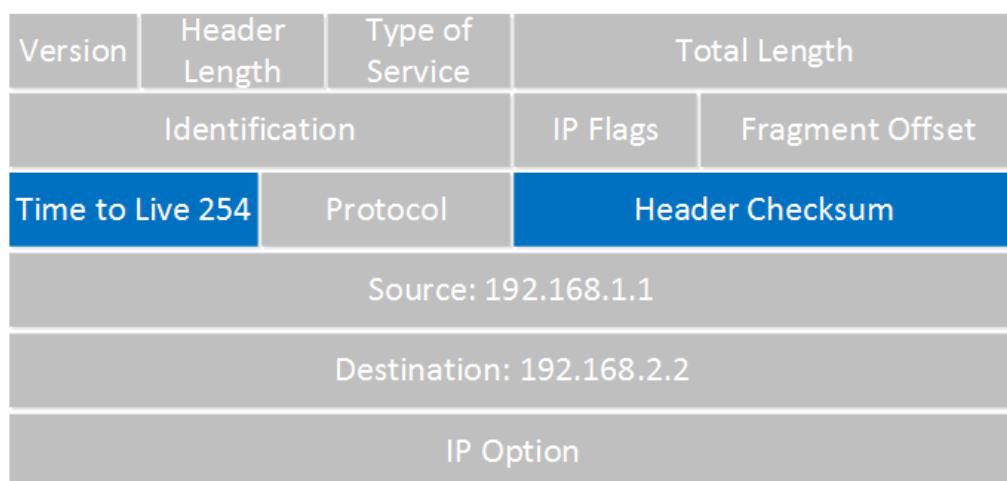
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
      a - application route
      + - replicated route, % - next hop override, p - overrides from PfR
```

```
Gateway of last resort is not set
```

```
    192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C      192.168.1.0/24 is directly connected, GigabitEthernet0/1
L      192.168.1.254/32 is directly connected, GigabitEthernet0/1
S      192.168.2.0/24 [1/0] via 192.168.12.2
    192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks
C      192.168.12.0/24 is directly connected, GigabitEthernet0/2
L      192.168.12.1/32 is directly connected, GigabitEthernet0/2
```

Observe na tabela de roteamento que R1 sabe como chegar até a rede 192.168.2.0/24, o endereço IP do próximo salto é 192.168.12.2. Agora, ele fará uma segunda pesquisa na tabela de roteamento para ver se sabe como chegar a 192.168.12.2, chamamos isso de **roteamento recursivo**. Como você pode ver, há uma entrada para 192.168.12.0/24 através da interface GigabitEthernet0/2.

Antes do pacote IP ser encaminhado, ainda necessitamos fazer mais uma coisa. Uma vez que o roteador está encaminhando o pacote IP, ele tem que diminuir o campo TTL (Time to Live) em um. Quando o roteador faz essa diminuição ele altera o cabeçalho IP, e com isso precisa calcular uma nova soma de verificação do cabeçalho.



Feito isso, R1 verificará sua tabela ARP para ver se há alguma entrada ‘arp’ para 192.168.12.2:

```
R1#show ip arp
Protocol  Address          Age (min)  Hardware Addr  Type   Interface
Internet  192.168.1.1        58     fa16.3e87.9c2a  ARPA   GigabitEthernet0/1
Internet  192.168.1.254       -      fa16.3e3f.fd3c  ARPA   GigabitEthernet0/1
Internet  192.168.12.1        -      fa16.3e02.83a1  ARPA   GigabitEthernet0/2
Internet  192.168.12.2        95     fa16.3e01.0c98  ARPA   GigabitEthernet0/2
```

O Roteador 1 tem entrada na tabela ARP. Caso não tivesse, R1 enviaria uma solicitação ARP (arp request) para encontrar o endereço MAC do 192.168.12.2.

A partir daí, o R1 construirá um novo quadro Ethernet com o endereço MAC da interface GigabitEthernet 0/2 sendo o source, e o endereço MAC da interface GigabitEthernet 0/2 do R2 como destino. O pacote IP será então encapsulado neste novo quadro Ethernet.

Origem: fa16.3e02.83a1	Destino: fa16.3e01.0c98	Origem: 192.168.1.1	Destino: 192.168.2.2
---	--	--------------------------------------	---------------------------------------

O quadro será encaminhado para o Roteador 2.

➤ R2

Quando o frame Ethernet chegar ao R2, ele seguirá os mesmos passos do R1:

- Verificará o FCS do quadro Ethernet.
- Desencapsulará o pacote IP e descartará o quadro ethernet.
- Verificará a soma de verificação do cabeçalho IP.
- Verificará o endereço IP de destino.

Na tabela de roteamento do R2, encontramos o seguinte:

```
R2#show ip route

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default, U - per-user static route
        o - ODR, P - periodic downloaded static route, H - NHRP, 1 - LISP
        a - application route
        + - replicated route, % - next hop override, p - overrides from PfR

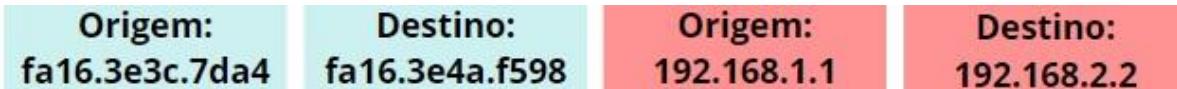
Gateway of last resort is not set

S      192.168.1.0/24 [1/0] via 192.168.12.1
      192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C          192.168.2.0/24 is directly connected, GigabitEthernet0/1
L      192.168.2.254/32 is directly connected, GigabitEthernet0/1
      192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks
C          192.168.12.0/24 is directly connected, GigabitEthernet0/2
L      192.168.12.2/32 is directly connected, GigabitEthernet0/2
```

A rede 192.168.2.0/24 está conectada diretamente ao R2 através da interface GigabitEthernet 0/1. R2 agora reduzirá o TTL do pacote IP de 254 para 253, recalculará a soma de verificação do cabeçalho IP e verificará sua tabela ARP para ver se sabe como chegar a 192.168.2.2:

R2#show ip arp						
Protocol	Address	Age (min)	Hardware Addr	Type	Interface	
Internet	192.168.2.2	121	fa16.3e4a.f598	ARPA	GigabitEthernet0/1	
Internet	192.168.2.254	-	fa16.3e3c.7da4	ARPA	GigabitEthernet0/1	
Internet	192.168.12.1	111	fa16.3e02.83a1	ARPA	GigabitEthernet0/2	
Internet	192.168.12.2	-	fa16.3e01.0c98	ARPA	GigabitEthernet0/2	

Há uma entrada ARP para o endereço 192.168.2.2, será criado um novo frame Ethernet e o pacote IP será encapsulado com os seguintes endereços:



O frame Ethernet será encaminhado para o PC2.

➤ PC2

O PC2 recebe o quadro Ethernet e:

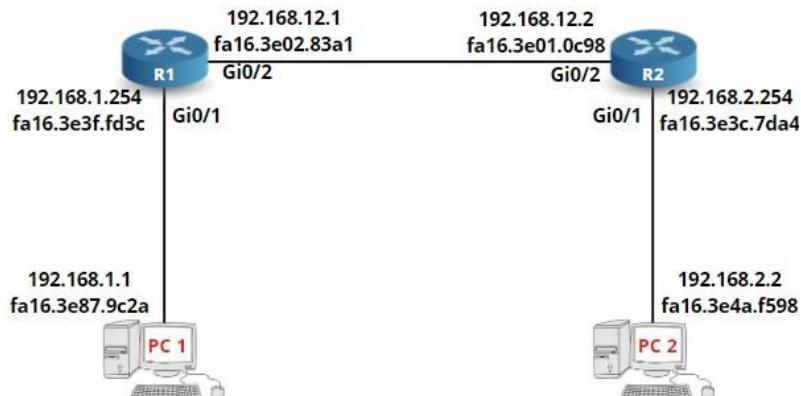
- Verifica o FCS;
- Encontrará seu próprio endereço MAC como o endereço MAC de destino;
- Desencapsulará o pacote IP do quadro;
- Encontrará seu próprio endereço IP como endereço IP de destino no pacote IP.

O PC2 então procurará o campo ‘protocol’ para descobrir qual protocolo da camada de transporte o pacote foi enviado, o que acontece a seguir depende do protocolo da camada de transporte que foi utilizado, porém, nosso tópico acaba aqui.

Agora que entendemos perfeitamente o passo a passo do encaminhamento de um pacote IP é hora de entrarmos nos códigos de uma tabela de roteamento.

3.1.a Routing protocol code

Vamos utilizar a mesma topologia anterior e aplicar o comando ‘show ip route’ para verificarmos todos os componentes e códigos de uma tabela de roteamento:



```

R1#show ip route

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
      a - application route
      + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is not set

      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.1.0/24 is directly connected, GigabitEthernet0/1
L        192.168.1.254/32 is directly connected, GigabitEthernet0/1
S        192.168.2.0/24 [1/0] via 192.168.12.2
      192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.12.0/24 is directly connected, GigabitEthernet0/2
L        192.168.12.1/32 is directly connected, GigabitEthernet0/2

```

Observe que as seguintes informações estão incluídas em uma tabela de roteamento básico:

- **Destination:** O endereço IP de destino final do pacote;
- **Next Hop:** O endereço IP para o qual o pacote está sendo encaminhado;
- **Interface:** A interface de saída que o dispositivo deve usar para encaminhar o pacote para o próximo salto ou para o destino final;
- **Metric:** Atribui um custo a cada rota disponível para que o melhor caminho seja o escolhido;
- **Routes:** Inclui sub-redes conectadas diretamente, sub-redes que não estão conectadas ao dispositivo, mas podem ser acessadas por meio de um ou mais saltos, e as rotas padrão que são usadas para certos tipos de tráfego ou quando faltam informações.

O código do protocolo de roteamento (routing protocol code) identifica por qual protocolo determinada rota foi aprendida. O routing protocol code está localizado bem no início da tabela de roteamento.

Nessa parte a Cisco foi bem gentil fornecendo até uma legenda logo no início da saída do comando para explicar o que cada valor significa:

```

R1#show ip route

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP

```

```

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
a - application route
+ - replicated route, % - next hop override, p - overrides from PfR

```

Eis os valores separados abaixo:

- L—local
- C—connected
- S—static
- R—RIP
- M—mobile
- B—BGP
- D—EIGRP
- EX—EIGRP external
- O—OSPF
- IA—OSPF inter area
- N1—OSPF NSSA external type 1
- N2—OSPF NSSA external type 2
- E1—OSPF external type 1
- E2—OSPF external type 2
- i—IS-IS
- su—IS-IS summary
- L1—IS-IS level-1
- L2—IS-IS level-2
- ia—IS-IS inter area
- *—candidate default
- U—per-user static route
- o—ODR
- P—periodic downloaded static route
- +—replicated route

3.1.b Prefix

Prefixo (Prefix) é simplesmente o endereço de rede. Em uma tabela de roteamento, o prefixo é o endereço de rede de destino. Um comprimento de prefixo (prefix-length) é apenas uma forma abreviada de expressar uma máscara de sub-rede usando a notação CIDR. Se a máscara de sub-rede for 255.255.255.0 o comprimento do prefixo será /24.

Parece bem simples, e é exatamente o que parece. Vamos voltar a tabela de roteamento que estávamos usando anteriormente para ilustrarmos com exemplos:

```

R1#show ip route
Gateway of last resort is not set

```

```

192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C      192.168.1.0/24 is directly connected, GigabitEthernet0/1
L      192.168.1.254/32 is directly connected, GigabitEthernet0/1
S      192.168.2.0/24 [1/0] via 192.168.12.2
                  192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks
C      192.168.12.0/24 is directly connected, GigabitEthernet0/2
L      192.168.12.1/32 is directly connected, GigabitEthernet0/2

```

Acompanhe comigo observando a tabela de roteamento de cima para baixo, você verá claramente os prefix e prefix-lengths para as redes de destino:

- 192.168.1.0/24 - **192.168.1.0** é o prefix e **/24** o prefix-lengths;
- 192.168.1.254/32 - **192.168.1.254** é o prefix e **/32** o prefix-lengths.

Tópico bem fácil.

3.1.c Network mask

Conforme estudado anteriormente, a Máscara de Rede (Subnet mask) nos informa quais bits do endereço IP são bits de rede e quais são os bits de identificação dos hosts. Para mais informações, retorne ao tópico **1.7 Configure and verify IPv4 addressing and subnetting**.

- 192.168.1.0/24: A network mask é 255.255.255.0 ou simplesmente /24;
- 192.168.1.254/32: A network mask é 255.255.255.255 ou simplesmente /32.

3.1.d Next hop

Quando o roteador (ou switch layer 3) precisa rotear dados para um destino específico, ele precisa ter o endereço IP do dispositivo mais próximo que conhece essa rota, esse dispositivo é chamado de ‘next hop’ (próximo salto). É para ele que o pacote é encaminhado.

Observe o comando ‘show ip route’ no Roteador 1 da topologia que estamos trabalhando:

```

R1#show ip route
Gateway of last resort is not set

      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C      192.168.1.0/24 is directly connected, GigabitEthernet0/1
L      192.168.1.254/32 is directly connected, GigabitEthernet0/1
S      192.168.2.0/24 [1/0] via 192.168.12.2
                  192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks
C      192.168.12.0/24 is directly connected, GigabitEthernet0/2
L      192.168.12.1/32 is directly connected, GigabitEthernet0/2

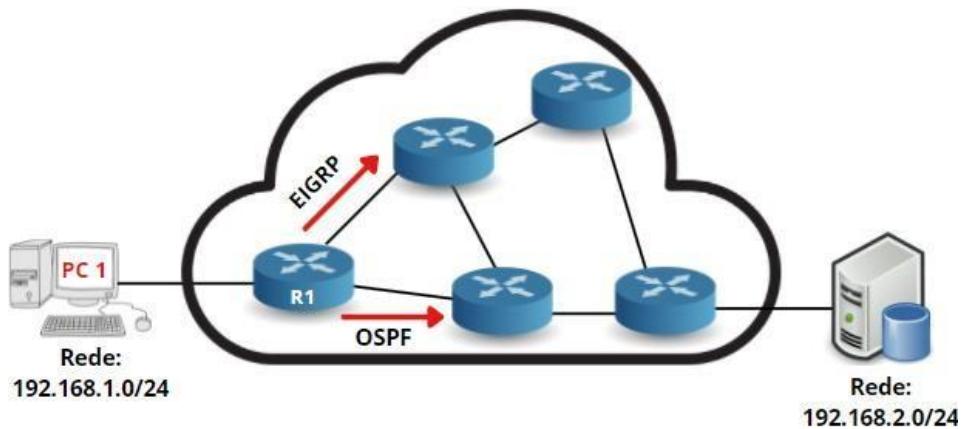
```

Ela nos informa que o R1 para chegar à rede 192.168.2.0 precisa encaminhar o pacote ‘via’ o dispositivo 192.168.12.2.

3.1.e Administrative distance

A distância administrativa é um dos conceitos de roteamento que a maioria dos alunos do CCNA tem dificuldade de entender. Mas não é difícil, basta um pouco de atenção.

Vamos começar utilizando a topologia abaixo de exemplo:



No cenário acima, temos uma rede executando **dois protocolos de roteamento simultaneamente**: OSPF e EIGRP. Ambos os protocolos de roteamento estão fornecendo informações para R1.

- O EIGRP informa que o roteador deve enviar pacotes IP usando o caminho na parte superior.
- O OSPF informa que o roteador deve enviar pacotes IP usando o caminho na parte inferior.

Que informações de roteamento vamos usar? Ambos? O OSPF ou EIGRP?

A resposta é: Quando dois ou mais protocolos de roteamento estão nos fornecendo **informações sobre o mesmo destino**, temos que fazer uma escolha, essa escolha é realizada comparando os números da **distância administrativa** ou **AD**.

Observe a tabela abaixo com as distâncias administrativas dos principais protocolos:

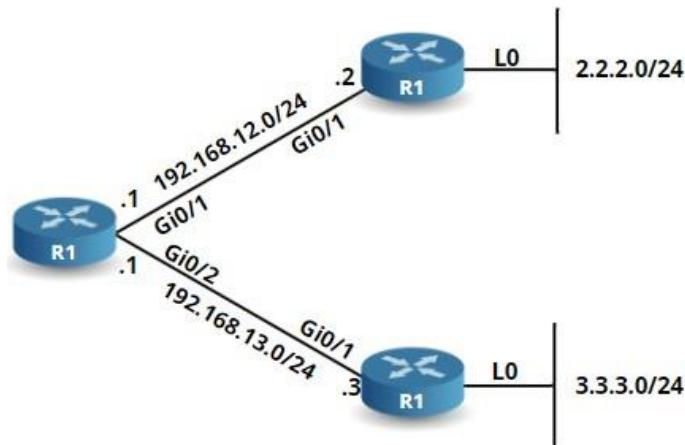
Protocolos	Distância Administrativa
Diretamente conectado	0
Rota Estática	1
EIGRP	90
OSPF	110
RIP	120

Quanto menor a distância administrativa, melhor. Como você pode ver, uma rota conectada diretamente tem um AD de 0. Isso faz sentido, pois não há nada melhor do que estar conectado diretamente ao roteador. Uma rota estática tem uma distância administrativa muito baixa de 1, o que também faz sentido, pois é algo que você configura manualmente. Às vezes, é necessário usar uma rota estática para “anular” as decisões de um protocolo de roteamento.

O EIGRP tem uma distância administrativa de 90, o que faz sentido, pois é um protocolo de roteamento Cisco. OSPF tem 110 e RIP tem 120.

No exemplo acima, o roteador R1, usará as informações que o EIGRP está informando, uma vez que a AD do EIGRP é 90, portanto, é melhor (menor) do que OSPF que é de 110.

Vamos a uma outra topologia:



O R1 está conectado ao R2 e R3. Abaixo sua tabela de roteamento:

```
R1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default, U - per-user static route
        o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
        a - application route
        + - replicated route, % - next hop override, p - overrides from PfR

Gateway of last resort is not set

2.0.0.0/24 is subnetted, 1 subnets
R    2.2.2.0 [120/1] via 192.168.12.2, 00:00:21, GigabitEthernet0/1
3.0.0.0/24 is subnetted, 1 subnets
S    3.3.3.0 [1/0] via 192.168.13.3
192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks
C      192.168.12.0/24 is directly connected, GigabitEthernet0/1
L      192.168.12.1/32 is directly connected, GigabitEthernet0/1
192.168.13.0/24 is variably subnetted, 2 subnets, 2 masks
C      192.168.13.0/24 is directly connected, GigabitEthernet0/2
L      192.168.13.1/32 is directly connected, GigabitEthernet0/2
```

Observe que o R1 aprendeu sobre a rede 2.2.2.0/24 por meio do RIP. Entre os colchetes, encontramos:

[120/1]

120 é a distância administrativa, 1 é a métrica (Se você não sabe o que é métrica não se preocupe, é o nosso próximo assunto). No caso específico do RIP, é a contagem de saltos.

R1 também tem uma rota estática para 3.3.3.0/24 a R3. Entre os colchetes encontramos:

[1/0]

1 é a distância administrativa, pois esta é uma rota estática, não há métrica, então é 0.

3.1.f Metric

Métrica é o valor do melhor caminho para uma rede de destino, calculado pelo protocolo de roteamento. A métrica varia de acordo com o protocolo de roteamento dinâmico envolvido.

Podemos dizer que é uma medida da “distância” para chegar à rede de destino. Por exemplo, para o protocolo RIP a melhor métrica é a que tem menos saltos (ou roteadores no caminho), portanto a métrica do RIP indica quantos roteadores você deve cruzar para alcançar o prefixo de destino. Protocolos diferentes têm matrizes diferentes, conforme descrito na tabela abaixo:

Protocolos	Distância Administrativa
Rota Estática	0
RIP	Contagem de saltos
EIGRP	Valores de ‘K’
OSPF	Largura de banda

O OSPF leva em conta a largura da banda, já o EIGRP faz uma conta bem complicada... Importante esclarecer que a métrica é utilizada somente por aquele protocolo de roteamento.

3.1.g Gateway of last resort

Se não houver rotas específicas na tabela de roteamento para um destino particular, o Gateway of last resort (gateway de último recurso ou rota padrão) será usado.

Vamos aprender como um host sabe quando usar o Gateway of last resort e como ele funciona nos bastidores.

Quando um host deseja enviar um pacote IP para outro host, ele verifica se o destino está dentro ou fora da sua própria rede. Se o destino estiver na mesma rede, ele usará o ARP para encontrar o endereço MAC do destino e poderá encaminhar o pacote IP.

Quando o host não está na mesma rede e o roteador não possui uma rota específica para aquela rede, é utilizado o Gateway of last resort. Os motivos para termos um Gateway of last resort configurado são variados, mas comumente é utilizado em redes stub ou como saída para a Internet (afinal, é impossível ter todas as rotas necessárias para navegar na internet no roteador). Qualquer um dos comandos abaixo serve para configurar o Gateway of last resort:

1. ip default-gateway a.b.c.d
2. ip default-network a.b.c.d
3. ip route 0.0.0.0 0.0.0.0 a.b.c.d

3.2 Determine how a router makes a forwarding decision by default

Descobriremos agora como os roteadores Cisco selecionam a melhor rota a ser usada para o encaminhamento de pacotes. O exame CCNA inclui os seguintes tópicos, que abordaremos em detalhes:

- 3.2.a Longest match
- 3.2.b Administrative distance
- 3.2.c Routing protocol metric

Acima, temos a lista de critérios de seleção, na ordem de preferência, que um roteador usa para escolher o melhor caminho entre as várias opções disponíveis. No entanto, para compilar a tabela de roteamento, o processo é invertido.

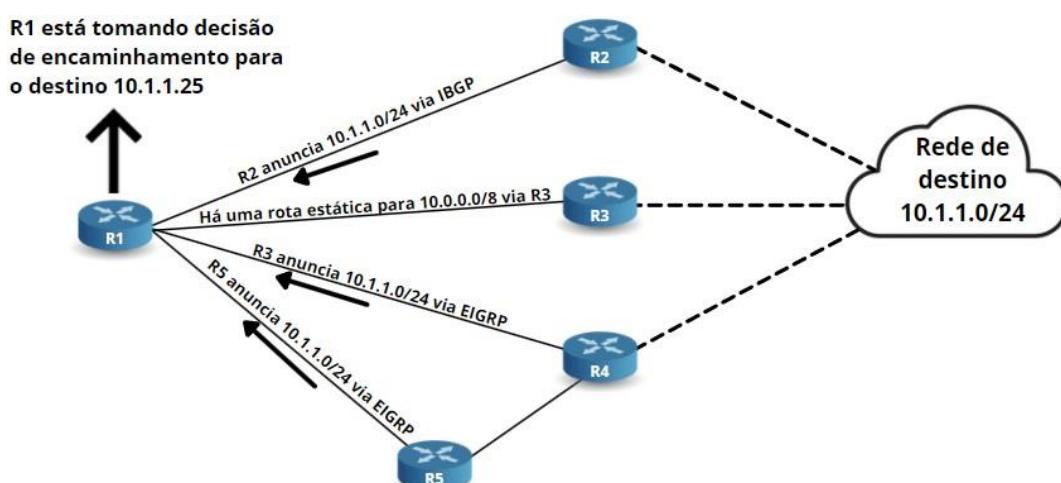
Em primeiro lugar, cada protocolo de roteamento seleciona a melhor rota usando seu próprio procedimento de comparação de métricas. Se houver mais de uma rota candidata de protocolos diferentes, as distâncias administrativas são comparadas e apenas um protocolo instala sua rota na tabela de roteamento. Em alguns casos, um protocolo de roteamento, em vez de preferir um único caminho, pode usar vários próximos saltos para a mesma rede dividindo assim a carga (load balance) entre vários links.

A decisão de encaminhamento é então baseada apenas na correspondência mais longa (longest match), pois a tabela de roteamento já filtrou todas as outras rotas, deixando apenas as melhores. O roteador procura a correspondência mais longa (longest match) para um destino e prefere rotas de prefixo IP mais específicas em vez de rotas mais amplas.

Por questões didáticas, adotarei a abordagem ascendente para explicar todo esse processo, ou seja, começaremos com a escolha do protocolo, passando para a escolha Inter protocolo e, finalmente, realizando a comparação de correspondência mais longa (longest match).

Observe a rede abaixo:

A topologia abaixo mostra um diagrama com um roteador selecionando o melhor caminho para encaminhar o tráfego para um host com o endereço IP 10.1.1.25.



Determine como um roteador toma uma decisão de encaminhamento

3.2 c Routing Protocol Metric

Os protocolos de roteamento dinâmico calculam e usam um valor numérico para descrever o custo de um caminho para um destino. Esse número é chamado de métrica e é específico para cada protocolo de roteamento. Os valores da métrica de dois protocolos de roteamento diferentes não são comparados entre si. Todos os protocolos de roteamento usam propriedades diferentes do caminho ou cálculos diferentes.

Por exemplo, alguns protocolos usam uma métrica simples, como o número de roteadores ou saltos que um pacote precisa cruzar para chegar à rede remota. Se dois roteadores anunciam rotas para tal rede, aquele que tiver o número menor de saltos será escolhido. Alguns outros protocolos usam largura de banda para definir o custo de determinado caminho.

A Tabela abaixo lista os diferentes protocolos de roteamento e a métrica que eles usam.

Protocolo	Métrica
RIP	Número de saltos entre a origem e o destino.
OSPF, IS-IS	Valor cumulativo baseado na largura de banda dos vários links entre origem e destino que formam o custo.
EIGRP	Métrica composta baseada em vários parâmetros: Delay, Largura de banda, confiabilidade, etc.
BGP	Número de Sistemas Autônomos para chegar ao destino, entre outros.

Métrica Internal gateway protocol (IGP)

As métricas IGP (protocolos internos), com exceção do RIP, fornecem uma mensuração do desempenho dos links até determinado destino. Eles são baseados em parâmetros fixos, como largura de banda e delay. Apenas a fórmula de cálculo da métrica do protocolo EIGRP pode incluir recursos dinâmicos, como utilização e confiabilidade, no entanto, eles não são usados por padrão.

Os protocolos IGP também preferem rotas que foram injetadas no protocolo internamente, ou seja, rotas nas quais os roteadores têm interfaces. As rotas externas são representadas por rotas injetadas por redistribuição de outros protocolos ou através de uma rota estática. Por exemplo, o OSPF seleciona rotas intra-áreas, depois rotas inter-áreas e, finalmente, rotas externas. Esta seleção ocorre antes das comparações das métricas (veremos sobre áreas OSPF mais adiante).

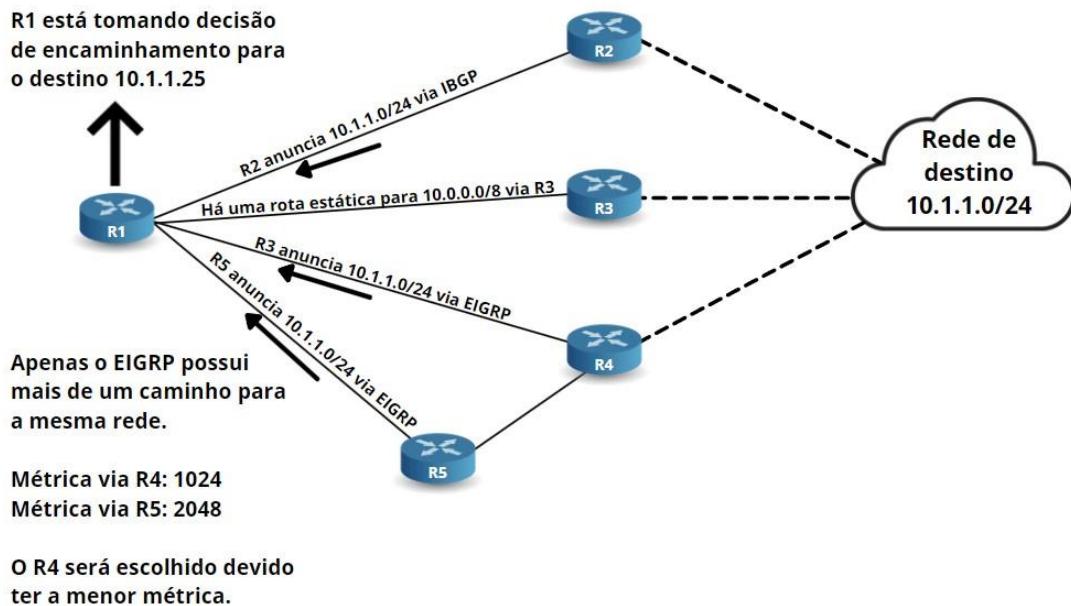
Métrica Exterior Gateway Protocols

O BGP tem um objetivo diferente ao escolher o melhor caminho. Como é um protocolo usado entre diferentes organizações, ele foi projetado para incluir vários atributos que podem ser usados para influenciar e direcionar o fluxo de tráfego para o caminho desejado. O processo de seleção do caminho do BGP consiste em mais de 10 etapas. Muitos dos atributos são configurados estaticamente para influenciar o processo de seleção e representam uma visão administrativa do custo do caminho, ao contrário do cálculo do IGP baseado em algumas avaliações objetivas do desempenho do caminho.

Exemplo de melhor seleção de rota usando métricas dos protocolos de roteamento

Na topologia abaixo, o roteador R1 enxerga 2 caminhos para a rede 10.1.1.0/24 através da tabela de topologia EIGRP. O caminho via R4 possui um custo de 1024 e o caminho via R5 de 2048. O processo EIGRP no roteador R1 escolherá o caminho via roteador R4 como rota candidata.

No próximo tópico, veremos o processo pelo qual o roteador decide se o EIGRP tem permissão para instalar sua rota na tabela de roteamento do roteador.



Seleção usando métrica do protocolo de roteamento

3.2.b Administrative distance

Distância administrativa é um fator de desempate usado quando há duas ou mais rotas candidatas aprendidas por meio de protocolos de roteamento diferentes. Apenas uma dessas rotas será instalada na tabela de roteamento.

A distância administrativa é um valor numérico pré-configurado da confiabilidade de uma fonte de informações de roteamento. Os protocolos “mais confiáveis” têm o número de distância administrativa menor.

Protocolo	Distância Administrativa	Comentário
Rede diretamente conectada	0	Redes nas quais o roteador está diretamente conectado, não pode ser alterada.
Estática	1	Rotas criadas manualmente, valor pode ser alterado para criar rotas estáticas flutuantes.
Eigrp (Rota Sumarizada)	5	Propriedade da Cisco, as rotas desse tipo são visíveis apenas para o roteador que as criou. Funciona como um mecanismo de prevenção de loops.
eBGP	20	Atribuído a rotas que são aprendidas de vizinhos BGP externos. Protocolo não proprietário.
EIGRP	90	Protocolo proprietário da Cisco.
IGRP	100	Protocolo obsoleto, não é mais utilizado, também era proprietário da Cisco.
OSPF	110	Protocolo interno, não proprietário.
IS-IS	115	Protocolo interno, não proprietário.
RIP	120	Protocolo interno, não proprietário.
EIGRP (Externo)	170	Protocolo proprietário da Cisco. São as rotas redistribuídas para o EIGRP. Também serve como mecanismo de prevenção de loops.
iBGP	200	Rotas aprendidas de pares BGP internos. Protocolo não proprietário.
OMP	251	Protocolo proprietário da Cisco utilizado em SD-Wan.

Na tabela acima falamos de redistribuição, um conceito que veremos mais à frente.

Redes conectadas diretamente e rotas estáticas

Como mostra a tabela, as redes conectadas diretamente têm a menor distância administrativa. Um roteador possui uma interface em cada uma das redes conectadas a ele.

As rotas estáticas, por padrão, possuem preferência em cima de qualquer rota aprendida dinamicamente com o mesmo ‘comprimento de prefixo’ (prefix length). Alterar a distância administrativa de uma rota estática para ela ter uma A.D maior que o do protocolo de roteamento dinâmico é uma maneira comum de realizar backup. Nesta configuração, se a rota dinâmica não estiver mais disponível, a rota estática a substituirá e fornecerá um caminho secundário. Essa rota é chamada de **rota estática flutuante**.

Distância administrativa dos internal gateway protocol (IGP)

Os valores padrão de distância administrativa classificam os protocolos IGP na seguinte ordem de preferência: EIGRP, OSPF, IS-IS e RIP. Com exceção do RIP, a prioridade não significa que um protocolo seja mais confiável ou preciso do que outro.

Na maioria das redes, há um único protocolo IGP, nesses casos, não faz diferença o AD. No entanto, em algumas situações, como fusão de redes ou transição para um protocolo diferente, o administrador pode ter de trabalhar com vários IGPs ao mesmo tempo. Em tais redes, as distâncias administrativas padrão podem ser ajustadas para que um protocolo tenha preferência maior que outro, de acordo com o planejado pelo administrador. É recomendável testar as configurações em um laboratório, pois configurações incorretas podem causar diversos problemas, como loops de rede.

Distância administrativa do Exterior Gateway Protocol (EGP)

Há apenas um único Exterior Gateway Protocol que não está obsoleto, o Border Gateway Protocol (BGP). Se um roteador não executa nenhum outro protocolo de roteamento dinâmico, então, como acontece com os IGPs, o valor da distância administrativa não afeta o processo de seleção da rota.

No entanto, em redes corporativas, é comum executar o BGP junto com um dos IGPs. Por exemplo, uma empresa pode trocar rotas via BGP com seus provedores de Internet ao mesmo tempo, ele pode rodar internamente OSPF ou EIGRP. Em tais cenários, as rotas podem ser divididas em externas e internas. O BGP tem autoridade para as rotas externas, e IGP para as rotas internas.

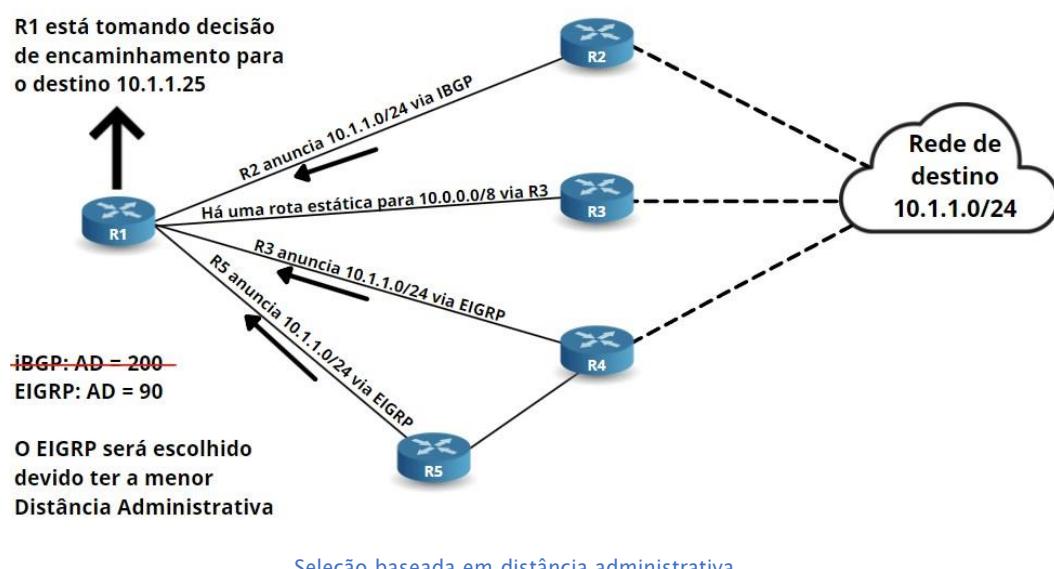
A distância administrativa padrão de rotas BGP externas garante que um roteador não comece a preferir uma rota para redes externas por meio de um roteador IGP adjacente, que muitas vezes pode anunciar essa rede de volta se houver mais de um roteador executando a redistribuição entre os protocolos.

Diferentemente, os pares (peers) do IBGP possuem uma distância administrativa de 200, que é maior do que qualquer rota AD do IGP. Isso faz com que um roteador use IGP como a fonte para os destinos internos.

Curiosamente, em alguns casos, o BGP interno pode ultrapassar as rotas de BGP externas, mesmo se o último tiver uma distância administrativa menor. A razão para isso é que o processo BGP realiza sua própria avaliação ao selecionar o melhor caminho antes de colocar a rota na tabela de roteamento. Por exemplo, as rotas BGP recebidas via peer interno podem ter um valor melhor de ‘local preference’ e, como resultado, terá maior preferência sobre a mesma rota aprendida via peer externo, que seria instalada com AD de 20. Como a melhor rota é de um peer BGP interno, ele será instalado na tabela de roteamento com AD de 200.

Seguindo nosso exemplo anterior, após o EIGRP selecionar o caminho via roteador R4, ainda temos 3 caminhos possíveis, via roteadores R2, R3 e R4. Ambos os roteadores R2 e R3 desejam instalar a mesma rede - 10.1.1.0/24 na tabela de roteamento. A distância administrativa será usada agora para decidir qual é o melhor caminho.

Como o EIGRP possui melhor distância administrativa (90) que o BGP interno (200), o caminho via roteador R2 será o selecionado. Essa decisão ocorre antes que qualquer encaminhamento de pacotes seja realizado, pois ainda estamos na parte da convergência do protocolo de roteamento. Ambas as rotas (10.0.0.0/8 e 10.1.1.0/24) agora estão instaladas na tabela de roteamento. A próxima etapa do roteador é realizar a seleção com base na correspondência mais longa (Longest match);



3.2.a Longest match

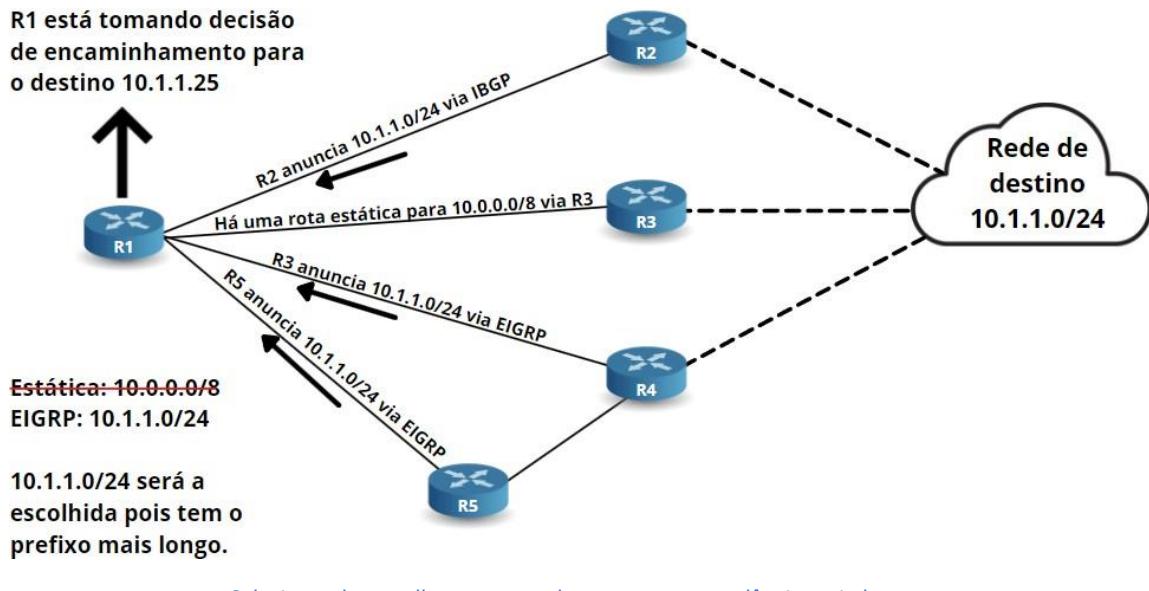
Todos os protocolos de roteamento unicast definem as rotas usando o endereço IP de destino de um pacote. A correspondência mais longa (Longest match) refere-se ao processo de identificação da rota para a rede mais específica à qual o pacote corresponde.

Por exemplo, a rota padrão (default route) ou rota para 0.0.0.0/0 corresponde a todos os pacotes. O próximo salto dessas rotas é frequentemente chamado de ‘gateway de último recurso’ porque é a rota de menor preferência, usada apenas se nenhuma outra rota correspondente existir.

A rota mais específica é uma rota de host com um comprimento de prefixo de 32 (ou máscara de sub-rede 255.255.255.255). Por exemplo, 192.168.100.25/32 é uma rota de host e os pacotes enviados a esse host específico sempre seguirão essa rota.

A diferença importante entre a correspondência mais longa e as outras duas etapas é que o roteador compara duas rotas diferentes, sendo uma super conjunto da outra. Ambas as redes aparecerão na tabela de roteamento. Essa situação geralmente existe quando há sumarização sendo realizada na rede, que é o processo de combinar várias rotas em uma única.

Observe o exemplo abaixo, o roteador R1 precisa escolher entre 2 rotas: 10.0.0.0/8 configurado estaticamente e 10.1.1.0/24 aprendido dinamicamente. A rota /24 é mais específica e possui a correspondência mais longa.



3.3 Configure and verify IPv4 and IPv6 static routing

Vamos entrar nas configurações das rotas estáticas em um roteador, tanto no IPv4 quanto IPv6. Vocês irão notar que algum desses assuntos já foram tratados anteriormente, mas temos que passar por eles novamente, dessa vez utilizarei uma nova abordagem.

3.3.a Default route

Uma rota padrão (default route) é a rota que o roteador usará se não houver nenhuma rota específica disponível para a rede de destino. Nos roteadores Cisco a veremos como ‘Gateway of Last Resort’.

O comando para configurar uma rota default IPv4 é:

```
router(config)# ip route 0.0.0.0 0.0.0.0 {interface | next-hop}
```

As informações entre chaves, significam que, você tem a opção de colocar a interface que o próximo salto para determinada rede está configurada ou simplesmente o próximo salto. Exemplo do comando acima com o next-hop configurado:

```
router(config)# ip route 0.0.0.0 0.0.0.0 10.0.255.2
```

Para o IPv6 o comando é:

```
router(config)# ipv6 route ::/0 {interface | next-hop}
```

3.3.b Network route

Quando uma rota é criada para uma rede (como a maioria das entradas de rota fazem), ela é chamada de rota de rede. Isso significa simplesmente que a rota aponta para um grupo de hosts e não para um host específico:

O comando para configurar uma rota para uma rede no IPv4 é:

```
router(config)# ip route 10.0.0.0 255.255.255.0 {interface | next-hop}
```

O comando para configurar uma rota para uma rede no IPv6 é:

```
router(config)# ipv6 route 2001::/64 {interface | next-hop}
```

3.3.c Host route

Na maioria das vezes, criamos rotas para redes, mas por algum motivo, pode ser necessário criar uma rota que leve a um único host. Observe que neste tipo de configuração a máscara que acompanha a rota tem 32 bits de comprimento, o que significa que é uma rota para um único endereço IP.

O comando para configurar uma rota para um único host no IPv4 é:

```
router(config)# ip route 10.0.0.21 255.255.255.255 {interface | next-hop}
```

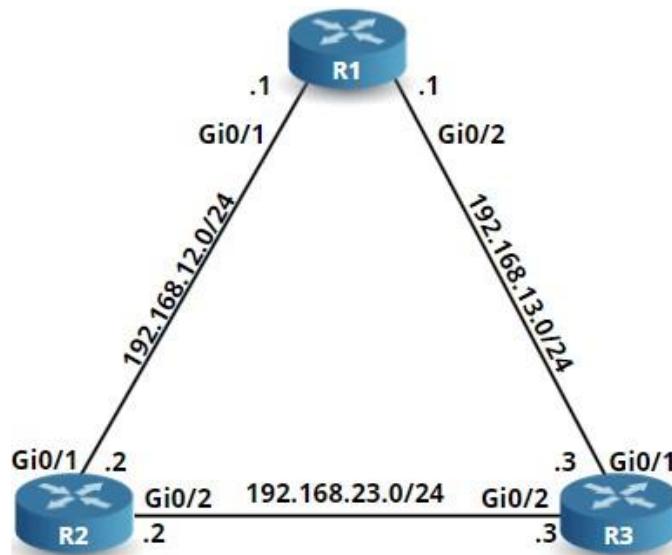
O comando para configurar uma rota para um único host no IPv6 é:

```
router(config)# ipv6 route 2001::21/128 {interface | next-hop}
```

3.3.d Floating static

As rotas estáticas têm uma distância administrativa muito baixa, precisamente de 1, isso significa que o roteador preferirá uma rota estática em vez de quaisquer outras rotas que tenham sido aprendidas por meio de protocolos de roteamento dinâmico. Se quisermos usar uma rota estática como rota de backup, teremos que alterar sua distância administrativa. Isso é chamado de **rota estática flutuante** (floating static).

Vamos usar a topologia abaixo:



Nessa topologia o R1 pode usar tanto o R2 como o R3 para chegar à rede 192.168.23.0/24.

Vamos configurar R1 e R2 para utilizarem o protocolo RIP para alcançar esta rede, observe que não há nada de muito especial em configurar um protocolo de roteamento:

```
R1(config)#router rip  
R1(config-router)#version 2  
R1(config-router)#no auto-summary  
R1(config-router)#network 192.168.12.0
```

```
R2(config-router)#version 2
R2(config-router)#no auto-summary
R2(config-router)#network 192.168.12.0
R2(config-router)#network 192.168.23.0
```

Para configurar o RIP, basta usar os comandos acima, informando as redes que farão parte. O R1 deve ser capaz de chegar na rede 192.168.23.0/24 através do R2:

```
R1#show ip route | begin 192.168.23.0
R      192.168.23.0/24 [120/1] via 192.168.12.2, 00:00:22, GigabitEthernet0/1
```

R1 agora possui uma rota para a rede 192.168.23.0/24. Mas, e se quisermos usar R3 como backup?

R3 não está executando nenhum protocolo de roteamento, então temos que usar uma rota estática. Essa é uma possibilidade quando não temos gerência sobre o outro roteador. Vamos criar uma rota estática para 192.168.23.0/24 através do R3:

```
R1(config)#ip route 192.168.23.0 255.255.255.0 192.168.13.3
```

A rota estática está funcionando, mas há um problema, vamos verificar novamente a tabela de roteamento:

```
R1#show ip route | begin 192.168.23.0
S      192.168.23.0/24 [1/0] via 192.168.13.3
```

Devido a rota estática ter a distância administrativa menor, o roteador adotou-a e retirou a rota do RIP da tabela. Vamos remover esta rota estática:

```
R1(config)#no ip route 192.168.23.0 255.255.255.0 192.168.13.3
```

Para que a rota gerada pelo protocolo de roteamento permaneça mesmo com uma rota estática é necessário que alteremos AD no momento em que estamos configurando a rota estática, observe:

```
R1(config)#ip route 192.168.23.0 255.255.255.0 192.168.13.3 ?
<1-255>      Distance metric for this route
multicast      multicast route
name          Specify name of the next hop
permanent     permanent route
tag           Set tag for this route
track         Install route depending on tracked item
<cr>
```

A distância administrativa do RIP é 120, portanto, se escolhermos um número maior, a rota estática será usada como backup. Vamos tentar 121:

```
R1(config)#ip route 192.168.23.0 255.255.255.0 192.168.13.3 121
```

Vamos verificar a tabela de roteamento novamente:

```
R1#show ip route | begin 192.168.23.0
```

```
R      192.168.23.0/24 [120/1] via 192.168.12.2, 00:00:26, GigabitEthernet0/1
```

A entrada da rota introduzida pelo protocolo RIP está sendo usada novamente. A rota estática ainda está em algum lugar nos bastidores. Vamos ver como ela atua como uma rota backup. Para simular um erro, darei ‘shutdown’ na interface do R2 que se conecta ao R1:

```
R2(config)#interface GigabitEthernet 0/1  
R2(config-if)#shutdown
```

O RIP é um protocolo de roteamento muito lento, então teremos que esperar um pouco para que a rota desapareça. Depois de alguns segundos, acontecerá a mudança na tabela de roteamento de R1:

```
R1#show ip route | begin 192.168.23.0  
  
S      192.168.23.0/24 [121/0] via 192.168.13.3
```

Concluímos, essa é a rota estática flutuante. Observe que a AD dela é 121.

Resumindo: Quando configuramos rotas estáticas com uma distância administrativa maior, as transformamos em rotas estáticas flutuantes. Ela pode ser usada como backup para rotas que foram aprendidas por meio de protocolos de roteamento dinâmico (ou outras rotas estáticas com uma distância administrativa menor).

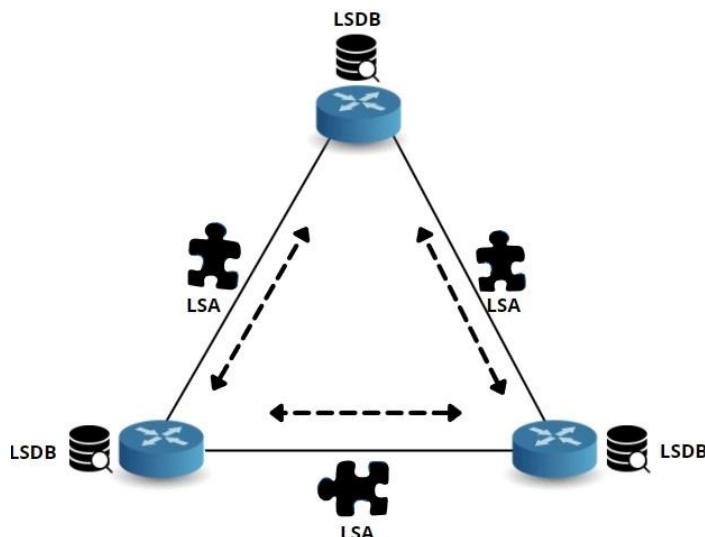
3.4 Configure and verify single area OSPFv2

Entraremos agora na parte de roteamento dinâmico, veremos OSFP. O blueprint da prova cobra quatro pontos específicos, porém é impossível falar sobre eles sem dar uma passada completa sobre o protocolo.

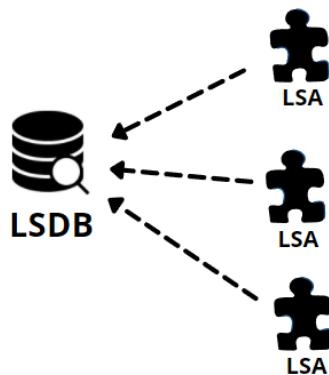
Introdução ao OSPF

A primeira coisa a saber sobre OSPF é que ele é um protocolo de roteamento link-state. Os protocolos de roteamento link-state são como GPS, eles têm um mapa completo da rede. Com um mapa completo é possível calcular o caminho mais curto para todos os destinos. Isso é interessante porque como o OSPF conhece todos os caminhos é impossível haver loops na rede! A desvantagem é que esse processo exige mais da CPU do que um protocolo de roteamento de vetor de distância (como é o caso do protocolo RIP).

Vamos entender o que significa o link-state:



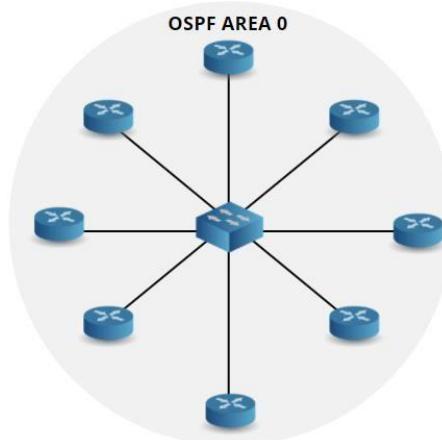
- Link: Interface do roteador.
- State: Descrição da interface e como ela está conectada aos roteadores vizinhos.



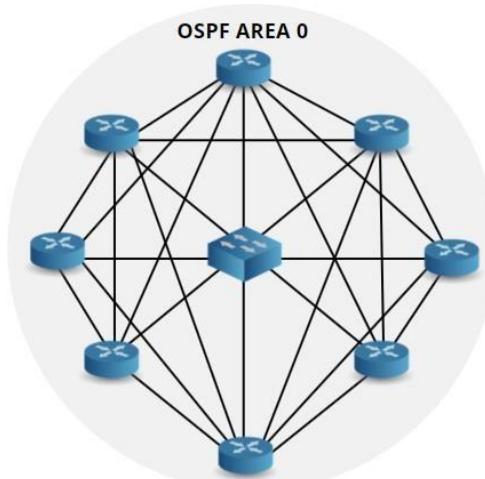
Os protocolos de roteamento link-state operam enviando **link-state advertisements (LSA)** para todos os outros roteadores link-state.

Todos os roteadores precisam receber esses anúncios **link-state advertisements** para que possam construir o **banco de dados de estado de link (link-state database - LSDB)**. Basicamente, todos os anúncios de link-state são uma peça do quebra-cabeça que constrói o LSDB.

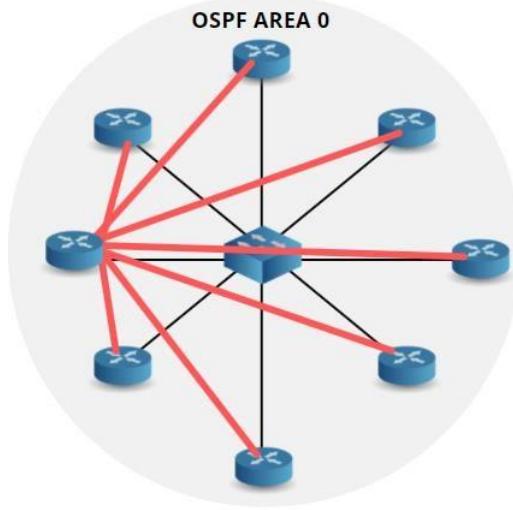
Se houver muitos roteadores OSPF, não será interessante que cada roteador OSPF envie seus LSAs para todos os outros roteadores OSPF. Observe o exemplo abaixo:



Na topologia temos uma rede formada por 8 roteadores rodando OSPF conectados em um switch. Cada um desses roteadores formará uma vizinhança OSPF com todos os outros roteadores, enviando ‘hello packets’, LSAs e construindo o LSDB. Isso é o que vai acontecer:

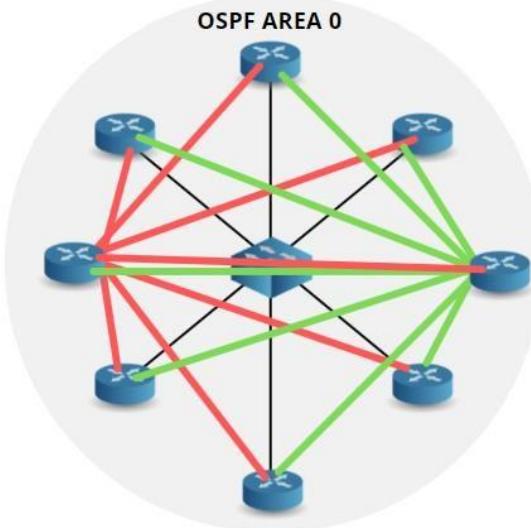


Obteremos uma rede full-mesh de vizinhos OSPF. Cada roteador irá inundar (floodar) LSAs para todos os outros roteadores, portanto, teremos uma quantidade elevada de tráfego OSPF. Porém, existe uma maneira de tornar esse processo mais eficiente!



Para evitar todo esse tráfego desnecessário, ficou definido um processo em que todos os roteadores OSPF enviam suas informações para um único roteador, que fica responsável por encaminhar para todos os outros roteadores as informações sobre a rede e os outros roteadores.

Esse roteador responsável por agrupar todas essas informações é chamado de '**DR (Designated Router)**'. Todos os roteadores OSPF formarão adjacência “completa” (full) com o DR.

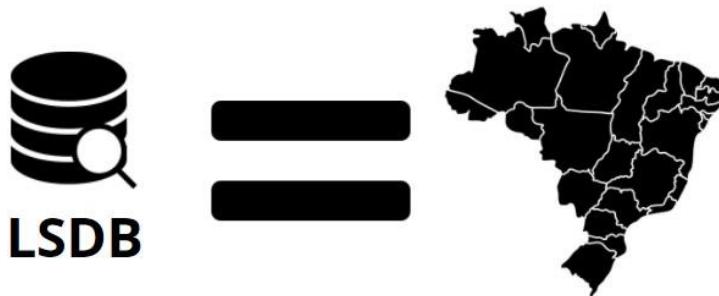


Como imprevistos podem acontecer, é necessário ter um roteador backup para o DR. Esse roteador é chamado de **BDR** (**Backup Designated Router**), e em caso de falha no BR ele que assumirá o controle. Todos os roteadores OSPF formarão ‘full neighbor adjacencies’ (adjacências vizinhas completas) apenas com o DR e BDR (essa é apenas a introdução, ainda entraremos mais a fundo no em DR\BDR, full neighbor adjacencies’, etc.)

Usamos apenas um DR/BDR em uma rede multiacesso como essa rede formada por um switch. Não há necessidade da utilização de DR/BDR em links **ponto a ponto**, afinal, há apenas um outro roteador do outro lado, não havendo razão para selecionar um DR/BDR.

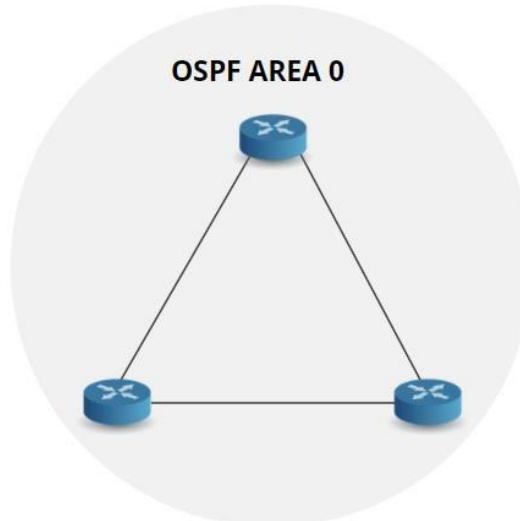
O LSDB possui a imagem completa da rede ou a **topologia** completa.

Podemos comparar o LSDB com o mapa completo do Brasil.

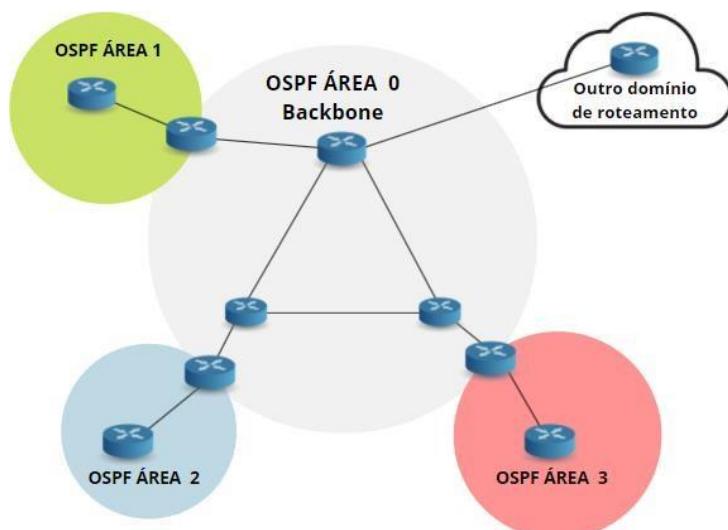


Quando cada roteador tiver um mapa completo da rede, eles começam a realizar os cálculos para descobrir o caminho ‘mais curto’ para todos os destinos, esses cálculos são efetuados usando o algoritmo **shortest-path first (SPF)** (**caminho mais curto primeiro**). As MELHORES informações vão para a tabela de roteamento. A fórmula que o OSPF utiliza é semelhante a de um GPS, ele olha todo o mapa, as diferentes formas de chegar ao destino e mostra apenas a melhor maneira de chegar até lá.

Agora que temos uma visão geral e superficial do OSPF, vamos entrar nos detalhes de funcionamento. Observe a topologia abaixo:



OSPF trabalha com os conceitos de **áreas**, e por padrão sempre teremos uma única área, normalmente é a **área 0**, também chamada de área de **backbone**.



É possível ter várias áreas, como na topologia acima onde há as áreas 1,2 e 3. Todas essas áreas **devem se conectar** à área de **backbone**. Se quisermos ir da área 1 para a área 2, teremos que passar pela área do backbone para chegar até lá. É impossível ir de uma área para outra sem passar pela área de backbone.

O OSPF trabalha com áreas para otimizar a capacidade de processamento dos roteadores, pois eles não precisam ter a imagem completa de toda topologia da rede, apenas da área em que estão conectados. Tenha em mente que o algoritmo SPF é dos anos 70, e o protocolo OSPF foi inventado nos anos 80, naquela época não tínhamos processadores potentes como o I5, I7, etc.

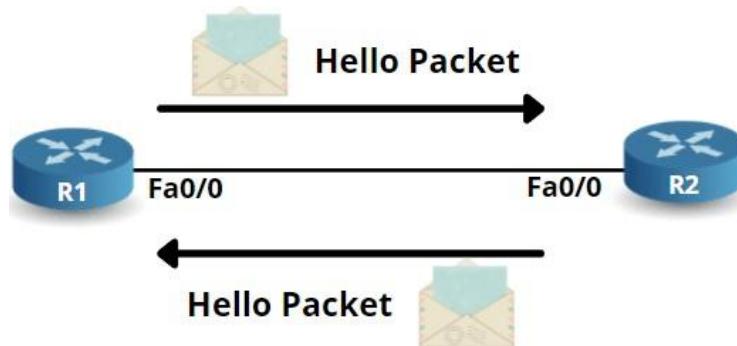
Lembre-se, quanto menor a topologia, mais rápido o algoritmo SPF funciona!

Na topologia, é possível ver no canto superior direito, um roteador em uma área chamada “outro domínio de roteamento”. Esta poderia ser outra rede executando outro protocolo de roteamento (como o RIP ou EIGRP), é possível importar e exportar rotas do RIP (ou outro protocolo de roteamento) para o OSPF ou vice-versa, isso é chamado de **redistribuição**.

- Os roteadores na área de backbone (área 0) são chamados de roteadores de backbone.
- Os roteadores na fronteira de 2 áreas (como aquele entre as áreas 0 e 1) são chamados de area border routers (roteadores de borda de área) ou ABR.
- Os roteadores que executam OSPF e estão conectados a outra rede que executa outro protocolo de roteamento são chamados de “autonomous system border routers” (roteadores de borda de sistema autônomo ou ASBR).

O OSPF não sai enviando link-state-advertisements aleatoriamente. Antes disso, os roteadores precisam se tornar vizinhos; e só depois de se tornarem vizinhos que eles começam a trocar link-state-advertisements.

Assim que configuramos o OSPF no roteador, ele começa a enviar hello packets (pacotes de saudação). Se por acaso também receber hello packets de outro roteador, eles se tornam vizinhos (neighbors).



No entanto, não basta só receber ‘hello packets’. O ‘hello packet’ é composto por vários campos e muitos deles precisam ser iguais nos roteadores, caso contrário, eles não se tornaram vizinhos. Vamos examinar os campos do ‘hello packet’:



- **Router ID:** Cada roteador OSPF precisa ter um ID exclusivo, que é o endereço IP mais alto de qualquer interface ativa. Falaremos mais sobre isso.
- **Hello / Dead Interval:** A cada X segundos, o roteador envia um pacote hello, se não houver nenhuma resposta a esse pacote hello por X segundos, ele declara que o outro roteador está ‘down’ e encerra a vizinhança. Esses valores devem ser iguais em ambos os lados para que os roteadores se tornem vizinhos.
- **Neighbors:** Todos os roteadores vizinhos são especificados no pacote hello.
- **Area ID:** Esta é a área em que o roteador está. Esses valores devem ser iguais em ambos os lados para que os roteadores se tornem vizinhos.
- **Router Priority:** É usado para determinar quem será o roteador designado ou o roteador designado backup.
- **DR e BDR IP Address:** Endereço IP do roteador designado e do roteador backup designado.
- **Authentication password:** Pode-se usar texto não criptografado e autenticação MD5 para OSPF, o que significa que todos os pacotes serão autenticados. Obviamente, é necessário a mesma senha e modo de autenticação em ambos os roteadores.
- **Stub area flag:** OSPF tem diferentes tipos de área. Ambos os roteadores precisam concordar com o tipo de área para se tornarem vizinhos.

Cada roteador OSPF precisa ter um router ID único, esse router ID é baseado no endereço IP mais alto que esteja configurado em qualquer interface ativa. Atente-se para o seguinte detalhe: Em roteadores Cisco, é possível criar interfaces de loopback que são como uma interface virtual, nessas interfaces é possível configurar endereços IP, essas interfaces estarão sempre on, tanto que se você fizer o teste de ping, sempre obterá uma resposta.

Portanto, se houver uma interface loopback no roteador OSPF, este endereço IP será usado como o ID do roteador, mesmo quando este endereço IP não for o mais alto. Isso faz todo sentido, porque como dito anteriormente, a interface de loopback sempre estará On, a menos que todo o roteador trave.

Nesse momento o OSPF está configurado, o roteador já possui vários vizinhos, e todos eles estão trocando pacotes LSA. Os roteadores construirão seu LSDB e possuem a imagem completa da topologia da rede. O próximo passo é executar o algoritmo SPF para determinar qual é o caminho mais curto para os mais diversos destinos.

Aqui entramos nas métricas que os protocolos de roteamento usam para determinar o melhor caminho. O OSPF usa uma métrica chamada **custo** que é baseada na largura de banda de uma interface:

$$\text{Custo} = \text{largura de banda de referência} / \text{largura de banda de interface}$$

A largura de banda de referência nos roteadores Cisco é uma interface de 100 Mbit. O cálculo do custo é realizado dividindo a largura de banda de referência pela largura de banda da interface.

Exemplo: Observe abaixo o cálculo para descobrir o custo em uma interface 100 Mbit:

$$\text{Custo} = \text{largura de banda de referência/largura de banda da interface.}$$

$$100 \text{ Mbit}/100 \text{ Mbit} = \text{Custo } 1$$

Exemplo: Se você tiver uma interface de 10 Mbit, qual será o custo?

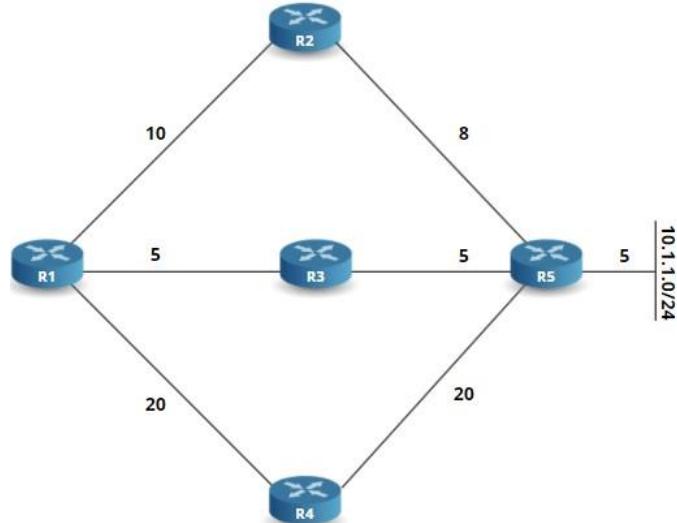
$$100 \text{ Mbit}/10 \text{ Mbit} = \text{Custo } 10$$

Exemplo: Se você tiver uma interface de 1 Mbit, qual será o custo?

$$100 \text{ Mbit}/1 \text{ Mbit} = \text{Custo } 100$$

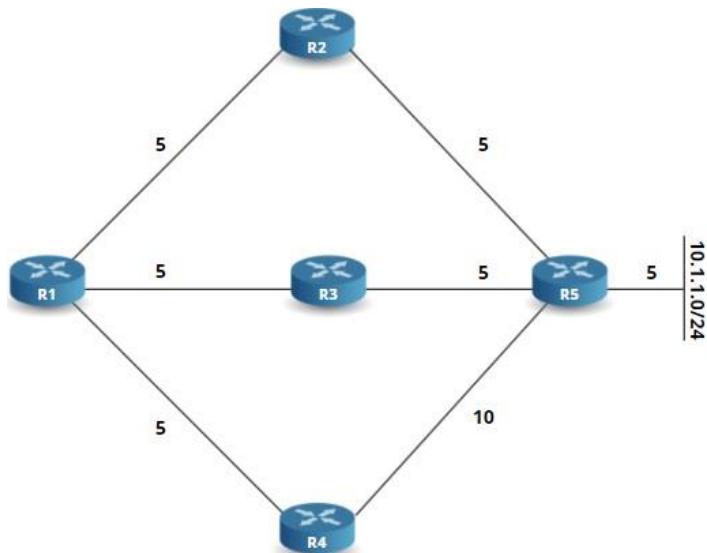
Quanto **menor** o custo, melhor é o caminho.

Observe a topologia abaixo, O R1 está executando o algoritmo SPF procurando o caminho mais curto para a rede de 10.1.10/24. Qual caminho ele usará?



Se utilizar o R2, teríamos um custo de $10 + 8 + 5 = 23$. O caminho no meio através do R3 é $5 + 5 + 5 = 15$. O caminho por R4 tem um custo de $20 + 20 + 5 = 45$. Portanto, o caminho do meio pelo roteador R3 tem o custo mais baixo, sendo esse o caminho escolhido.

Vejamos outro cenário:



Observe que o caminho através do R2 e R3 tem o mesmo custo ($5 + 5 + 5 = 15$). Vamos descobrir qual ação o SPF toma quando os custos são iguais.

A resposta é **balanceamento de carga**: OSPF utilizará os dois caminhos e fará o balanceamento de carga entre eles 50/50. Alguns detalhes sobre o balanceamento de carga OSPF:

- Os caminhos devem ter um custo igual.
- OSPF adiciona caminhos com um custo igual na tabela de roteamento.
- O número máximo de caminhos de custo igual suportados por um roteador é 32 (no entanto, isso pode depender da plataforma e/ou versão do IOS)
- Para tornar os custos dos caminhos iguais, basta alterar o “custo” de um link.

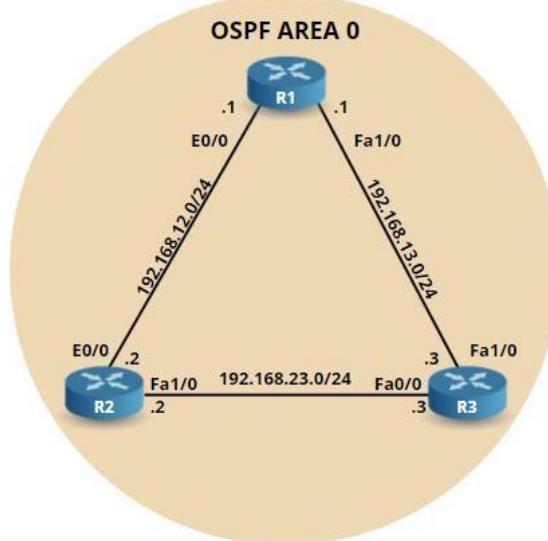
Se um caminho não for igual, podemos fazer isso alterando manualmente o custo ou a largura de banda de uma interface.

Para encerrar essa introdução, falarei o básico sobre autenticação no OSPF:

- OSPF pode fazer autenticação MD5.
- OSPF pode fazer autenticação de texto não criptografado.
- Autenticação pode ser configurada em toda a área ou em uma única interface.

Configuração OSPF

Após a introdução, é hora de partirmos para a parte prática. Vamos configurar o OSPF na topologia abaixo:



Nessa topologia todos os roteadores estão na área 0. Observe que o link entre R2 e R1 é um link Ethernet (10 Mb/s). Todos os outros links são FastEthernet (100Mb/s).

Começaremos com a configuração entre os roteadores R2 e R3:

```
R2(config)#router ospf 1
R2(config-router)#network 192.168.23.0 0.0.0.255 area 0
```

```
R3(config)#router ospf 1
R3(config-router)#network 192.168.23.0 0.0.0.255 area 0
```

Para entrarmos na configuração do OSPF, precisamos usar o comando ‘router ospf’. O número “1” é o número de identificação do processo OSPF, podemos escolher qualquer número para essa identificação. Esse ID só tem significado local, ou seja, podemos escolher outros números de processos nos demais roteadores.

A segunda etapa é usar o comando ‘network’, vamos dividir o comando para ficar mais claro a explicação:

```
network 192.168.23.0 0.0.0.255
```

O comando network faz duas coisas:

- Anuncia as redes que devem fazer parte do OSPF.
- Ativa o OSPF na (s) interface (s) que se enquadram neste intervalo de rede. Isso significa que o OSPF enviará pacotes ‘hello’ nessas interfaces.

Observando o comando, você verá que após o endereço IP 192.168.23.0 há o endereço 0.0.0.255. Esta não é uma máscara de sub-rede, mas uma **máscara curinga**. Uma máscara curinga é uma **máscara de sub-rede reversa**:

Máscara de sub-rede	255	255	255	0
Máscara curinga	11111111	11111111	11111111	00000000
	0	0	0	255
	00000000	00000000	00000000	11111111

Quando falamos em máscara de sub-rede reversa, na verdade queremos dizer que os 1s e 0s binários da máscara curinga são invertidos em comparação com a máscara de sub-rede. Uma máscara de sub-rede 255.255.255.0 é igual à máscara curinga 0.0.0.255. Não se preocupe muito com isso por enquanto, pois explicarei as máscaras curinga quando falarmos sobre listas de acesso!

OSPF usa áreas, então é necessário especificar a área:

```
area 0
```

No exemplo que estamos estudando, configuramos o OSPF de área única, em que todos os roteadores pertencem à área 0.

Depois de digitar esses comandos nos roteadores, veremos as seguintes mensagens no console:

```
R3# %OSPF-5-ADJCHG: Process 1, Nbr 192.168.23.2 on FastEthernet0/0 from LOADING
to FULL, Loading Done
```

```
R2# %OSPF-5-ADJCHG: Process 1, Nbr 192.168.23.3 on FastEthernet1/0 from LOADING
to FULL, Loading Done
```

Os roteadores R3 e R2 se tornaram vizinhos. Há alguns comandos que podemos utilizar para verificação:

```
R3#show ip ospf neighbor
Neighbor ID Pri State Dead Time Address Interface
192.168.23.2 1 FULL/BDR 00:00:36 192.168.23.2 FastEthernet0/0
```

```
R2#show ip ospf neighbor
Neighbor ID Pri State Dead Time Address Interface
192.168.23.3 1 FULL/DR 00:00:32 192.168.23.3 FastEthernet1/0
```

O comando ‘show ip ospf neighbor’ é fundamental para descobrir se o roteador tem vizinhos OSPF. O campo ‘state’ em ‘full’, indica que os roteadores se tornaram vizinhos com sucesso.

Cada roteador OSPF tem um ID de roteador (router ID) e podemos verificar com o comando ‘show ip protocols’:

```
R2#show ip protocols
Routing Protocol is "ospf 1"
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Router ID 192.168.23.2
```

```
R3#show ip protocols
```

```

Routing Protocol is "ospf 1"
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Router ID 192.168.23.3
```

Acima conseguimos ver o ID do roteador R2 e R3. Conforme explicado anteriormente, eles estão utilizando o endereço IP ativo mais alto como a “router ID”. Vamos criar uma interface loopback no R2 para mudarmos sua “router ID”:

```

R2(config)#interface loopback 0
R2(config-if)#ip address 2.2.2.2 255.255.255.0
```

É dessa forma que se cria uma interface de loopback. Pode-se escolher qualquer número para a interface.

```

R2#show ip protocols
Routing Protocol is "ospf 1"
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Router ID 192.168.23.2
```

A “router ID” ainda é a mesma. Precisamos “resetar” o processo OSPF para que a alteração tenha efeito:

```

R2#clear ip ospf process
Reset ALL OSPF processes? [no]: yes
```

O comando “clear ip ospf process” reseta o processo OSPF. Agora sim podemos verificar o router ID.

```

R2#show ip protocols
Routing Protocol is "ospf 1"
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Router ID 2.2.2.2
```

Também podemos alterar o router ID manualmente. Vamos mostrar esse processo no R3:

```

R3#show ip protocols
Routing Protocol is "ospf 1"
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Router ID 192.168.23.3
```

Por enquanto o Router ID é 192.169.23.3.

```

R3(config-router)#router-id 3.3.3.3
Reload or use "clear ip ospf process" command, for this to take effect
```

```
R3#clear ip ospf process  
Reset ALL OSPF processes? [no]: yes
```

O roteador é amigável o suficiente para nos avisar que precisamos resetar o processo OSPF para que a configuração tenha efeito:

```
R3#show ip protocols  
Routing Protocol is "ospf 1"  
Outgoing update filter list for all interfaces is not set  
Incoming update filter list for all interfaces is not set  
Router ID 3.3.3.3
```

Observe, agora o router ID é 3.3.3.3.

No momento, temos uma adjacência OSPF entre R2 e R3. Vamos acabar de configurar o OSPF nos demais links para que R2/R1 e R1/R3 também se tornem vizinhos OSPF:

```
R2(config)#router ospf 1  
R2(config-router)#network 192.168.12.0 0.0.0.255 area 0
```

```
R1(config)#router ospf 1  
R1(config-router)#network 192.168.12.0 0.0.0.255 area 0  
R1(config-router)#network 192.168.13.0 0.0.0.255 area 0
```

```
R3(config)#router ospf 1  
R3(config-router)#network 192.168.13.0 0.0.0.255 area 0
```

Anunciamos todas as redes dentro do OSPF. Antes de verificarmos a tabela de roteamento, vamos verificar se os roteadores realmente se tornaram vizinhos OSPF:

```
R2#show ip ospf neighbor  
Neighbor ID      Pri      State        Dead Time      Address          Interface  
192.168.13.1      1      FULL/BDR    00:00:31      192.168.12.1  Ethernet0/0  
3.3.3.3            1      FULL/DR     00:00:38      192.168.23.3  FastEthernet1/0
```

```
R1#show ip ospf neighbor  
Neighbor ID      Pri      State        Dead Time      Address          Interface  
3.3.3.3            1      FULL/BDR    00:00:33      192.168.13.3  FastEthernet1/0  
2.2.2.2            1      FULL/DR     00:00:30      192.168.12.2  Ethernet0/0
```

```
R3#show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
192.168.13.1	1	FULL/DR	00:00:37	192.168.13.1	FastEthernet1/0
2.2.2.2	1	FULL/BDR	00:00:30	192.168.23.2	FastEthernet0/0

Tudo está como esperado, os roteadores tornaram-se vizinhos OSPF e o ‘state’ está ‘full’, o que significa que a troca de informações terminou. Hora de verificar as tabelas de roteamento:

```
R2#show ip route ospf
```

```
0    192.168.13.0/24 [110/2] via 192.168.23.3, 00:09:45, FastEthernet1/0
```

R2 tem uma entrada, essa entrada é para a rede 192.168.13.0/24. Vamos decifrar a saída desse comando:

- O “O” significa OSPF. Ou seja, essa entrada foi aprendida por meio do OSPF.
- 192.168.13.0/24 é a rede que aprendemos. Este é o link entre os roteadores R1 e R3.
- O “110” é a distância administrativa do OSPF.
- O “2” é a métrica. O OSPF usa o custo como métrica. Para alcançar esta rede, temos um custo total de 2.
- “Via” é o endereço IP do próximo salto para onde enviaremos o tráfego. Para a rede específica, é o roteador R3.

Vamos dar uma olhada detalhada em como o OSPF calculou esse custo:

```
R2#show ip ospf interface fa1/0
```

```
FastEthernet1/0 is up, line protocol is up
```

```
Internet Address 192.168.23.2/24, Area 0
```

```
Process ID 1, Router ID 2.2.2.2, Network Type BROADCAST, Cost: 1
```

Podemos usar o comando ‘show ip ospf interface’ para verificar o custo de uma determinada interface. Como você pode ver, uma interface FastEthernet tem um custo de 1.

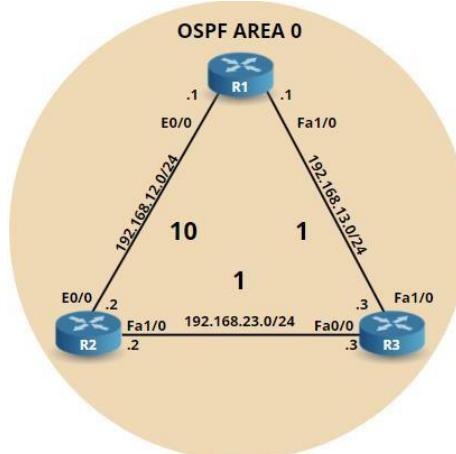
```
R2#show ip ospf interface e0/0
```

```
Ethernet0/0 is up, line protocol is up
```

```
Internet Address 192.168.12.2/24, Area 0
```

```
Process ID 1, Router ID 2.2.2.2, Network Type BROADCAST, Cost: 10
```

Uma interface Ethernet é mais lenta, por isso, tem um custo de 10. Vamos ilustrar isso:



Na imagem acima, adicionei o custo da interface ao lado de cada link. A partir do R2, podemos alcançar a rede 192.168.13.0/24 passando pelo R3 ou R1. Eis os custos:

- Por meio de R3: $1 + 1 = \text{custo } 2$.
- Por meio de R1: $10 + 1 = \text{custo } 11$.

Obviamente o caminho pelo R3 possui o menor custo. Como um experimento, podemos dar shutdown na interface FastEthernet0/0 do R3 para verificarmos se o R2 encontrara o outro caminho:

```
R3(config)#interface fastEthernet 0/0
R3(config-if)#shutdown

R2#show ip route ospf
0    192.168.13.0/24 [110/11] via 192.168.12.1, 00:01:20, Ethernet0/0
```

Agora, o roteador R2 alcançará a rede 192.168.13.0/24 por meio do roteador R1, que possui um custo total de 11. Antes de continuar, vamos habilitar a interface novamente:

```
R3(config)#interface fastEthernet 0/0
R3(config-if)#no shutdown
```

Vamos verificar nossa tabela de roteamento:

```
R2#show ip route ospf
0    192.168.13.0/24 [110/2] via 192.168.23.3, 00:00:01, FastEthernet1/0
```

O R2 está novamente usando a interface FastEthernet1/0. É possível forçar o OSPF a usar a interface Ethernet0/0, mesmo ela sendo mais lenta, fazemos isso alterando manualmente o custo:

```
R2(config)#interface fastEthernet 1/0
R2(config-if)#ip ospf cost 50
```

O comando “ip ospf cost” altera o custo da interface. Quando mudamos o curso da interface para 50, a interface FastEthernet deixou de ser a mais atraente.

```
R2#show ip ospf interface fastEthernet 1/0
FastEthernet1/0 is up, line protocol is up
Internet Address 192.168.23.2/24, Area 0
Process ID 1, Router ID 2.2.2.2, Network Type BROADCAST, Cost: 50
```

Observe que agora o custo da interface é de 50.

```
R2#show ip route ospf
0    192.168.13.0/24 [110/11] via 192.168.12.1, 00:01:20, Ethernet0/0
```

E como resultado, o OSPF irá preferir a interface Ethernet apesar de ela ser mais lenta. Vamos retirar essa mudança de custo para verificarmos as outras tabelas de roteamento:

```
R2(config)#interface fastEthernet 1/0
```

```
R2(config-if)#no ip ospf cost 50
```

```
R1#show ip route ospf
```

```
0    192.168.23.0/24 [110/2] via 192.168.13.3, 00:00:15, FastEthernet1/0
```

R1 tem uma única entrada para a rede 192.168.23.0/24 por meio do R3. Este é o caminho mais curto com o custo de 2.

```
R3#show ip route ospf
```

```
0    192.168.12.0/24 [110/11] via 192.168.23.2, 00:01:14, FastEthernet0/0
```

```
[110/11] via 192.168.13.1, 00:01:14, FastEthernet1/0
```

O R3 possui duas entradas. Ele aprendeu sobre a rede 192.168.12.0/24 e pode alcançá-la por meio de 192.168.23.2 (R2) ou de 192.168.13.1 (R1). O custo de ambos os caminhos é 11, lembre-se quando há custos iguais o OSPF faz balanceamento de carga (load balance).

Você se lembra da nossa interface loopback0 no R2? Nós a usamos para a identificação do roteador (router ID), mas também podemos anunciar-a no OSPF ou em qualquer outro protocolo de roteamento:

```
R2#show ip interface loopback 0
```

```
Loopback0 is up, line protocol is up
```

```
Internet address is 2.2.2.2/24
```

Interface loopback é uma interface normal com endereço IP e uma máscara de sub-rede. Vamos anunciar-a através do comando ‘network’:

```
R2(config)#router ospf 1
```

```
R2(config-router)#network 2.2.2.0 0.0.0.255 area 0
```

Rede anunciada, é hora de verificarmos as tabelas de roteamento:

```
R1#show ip route ospf
```

```
2.0.0.0/32 is subnetted, 1 subnets
```

```
0      2.2.2.2 [110/3] via 192.168.13.3, 00:00:26, FastEthernet1/0
```

```
0      192.168.23.0/24 [110/2] via 192.168.13.3, 00:00:26, FastEthernet1/0
```

R1 alcançará a interface loopback passando pelo R3. O custo total será 3:

- 1 (FastEthernet) + 1 (FastEthernet) + 1 (Loopback) = 3.

```
R3#show ip route ospf
```

```
0      192.168.12.0/24 [110/11] via 192.168.23.2, 00:01:42, FastEthernet0/0
```

```
[110/11] via 192.168.13.1, 00:01:42, FastEthernet1/0
```

```
2.0.0.0/32 is subnetted, 1 subnets
```

```
0      2.2.2.2 [110/2] via 192.168.23.2, 00:01:42, FastEthernet0/0
```

R3 tem um custo total de 2:

➤ 1 (FastEthernet) + 1 (Loopback) = 2.

A grande vantagem das interfaces de loopback é que elas podem ser acessadas da mesma forma que as interfaces normais, por exemplo, elas respondem ao comando ping como qualquer interface:

```
R3#ping 2.2.2.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:
!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/8 ms
```

Também podemos anunciar uma rota padrão (default route) no OSPF. Isso pode ser útil se o roteador estiver conectado à Internet e quisermos anunciar isso para os outros roteadores:

```
R2(config)#router ospf 1
R2(config-router)#default-information originate always
```

É necessário usar o comando “default-information originate”. Caso ainda não haja uma rota padrão na tabela de roteamento, será necessário adicionar a palavra-chave “Always”.

Vamos ver se a rota padrão foi anunciada:

```
R1#show ip route ospf
      2.0.0.0/32 is subnetted, 1 subnets
0        2.2.2.2 [110/3] via 192.168.13.3, 00:00:50, FastEthernet1/0
0        192.168.23.0/24 [110/2] via 192.168.13.3, 00:00:50, FastEthernet1/0
*E2 0.0.0.0/0 [110/1] via 192.168.13.3, 00:00:50, FastEthernet1/0
```

```
R3#show ip route ospf
0        192.168.12.0/24 [110/11] via 192.168.23.2, 00:00:45, FastEthernet0/0
                  [110/11] via 192.168.13.1, 00:00:45, FastEthernet1/0
      2.0.0.0/32 is subnetted, 1 subnets
0        2.2.2.2 [110/2] via 192.168.23.2, 00:00:45, FastEthernet0/0
*E2 0.0.0.0/0 [110/1] via 192.168.23.2, 00:00:45, FastEthernet0/0
```

Como você pode ver, R1 e R3 aprenderam a rota padrão a partir do R2.

Vamos continuar nossa configuração OSPF realizando autenticação de texto simples e MD5. Vamos começar configurando a autenticação de texto simples entre R2 e R3:

```
R2(config)#interface fastEthernet 1/0
R2(config-if)#ip ospf authentication
```

```
R2(config-if)#ip ospf authentication-key Senha123
```

```
R3(config)#interface fastEthernet 0/0
R3(config-if)#ip ospf authentication
R3(config-if)#ip ospf authentication-key Senha123
```

Primeiro, precisamos usar o comando “ip ospf authentication” para habilitar a autenticação de texto simples na interface. Em segundo lugar, precisamos configurar uma senha usando o comando “ip ospf authentication-key”.

Depois de configurar a autenticação em um roteador, você verá a adjacência do vizinho caindo até que você configure o outro roteador.

Há um comando, chamado “debug”, que é muito útil em situações como essa. Com ele é possível verificar se a autenticação foi ativa ou não:

```
R3#debug ip ospf packet
OSPF packet debugging is on
```

O comando “Debug ip ospf packet” irá mostrar uma visão geral dos pacotes OSPF que você está recebendo, a saída é semelhante a esta:

```
R3#
OSPF: rcv. v:2 t:1 1:48 rid:192.168.13.1
aid:0.0.0.0 chk:7D95 aut:0 auk: from FastEthernet1/0
```

Este é um pacote que recebemos do roteador R1. O “aut: 0” significa que este pacote não está autenticado, o que está certo, afinal, ainda não configuramos a autenticação entre R3 e R1.

```
OSPF: rcv. v:2 t:1 1:48 rid:2.2.2.2
aid:0.0.0.0 chk:3339 aut:1 auk: from FastEthernet0/0
```

Este pacote veio do roteador R2 e você pode ver que diz “aut:1”. Isso significa que a autenticação de texto simples está habilitada. Vamos desativar a depuração antes de continuar:

```
R3#no debug all
All possible debugging has been turned off
```

Vamos configurar a autenticação MD5 entre R1 e R3:

```
R3(config)#interface fastEthernet 1/0
R3(config-if)#ip ospf authentication message-digest
R3(config-if)#ip ospf message-digest-key 1 md5 Senha123
```

```
R1(config)#interface fastEthernet 1/0
R1(config-if)#ip ospf authentication message-digest
R1(config-if)#ip ospf message-digest-key 1 md5 Senha123
```

Primeiro, dizemos ao OSPF para usar MD5 com o comando “ip ospf authentication message-digest”. Em segundo lugar, o comando “ip ospf message-digest-key” que diz ao OSPF para usar a chave MD5 ‘1’ (você pode escolher qualquer número que desejar, desde que seja o mesmo em ambos os roteadores) e a senha “Senha123”.

Se você habilitar o ‘debug’, verá se está funcionando:

```
R1#debug ip ospf packet  
OSPF packet debugging is on
```

```
R1#  
  
OSPF: rcv. v:2 t:1 1:48 rid:3.3.3.3  
aid:0.0.0.0 chk:0 aut:2 keyid:1 seq:0x3C7EE6DC from FastEthernet1/0
```

```
R1#no debug all  
All possible debugging has been turned off
```

Na saída acima, observe o “auth: 2”, que significa autenticação MD5. Também é possível ver a ID da chave.

Nos exemplos acima, habilitei a autenticação por interface. Também é possível fazer isso para toda a área. Habilitar a autenticação em toda área pode economizar tempo se o roteador possuir muitas interfaces. A maneira de habilitar a configuração em toda a área é:

```
R3(config-if)#router ospf 1  
R3(config-router)#area 0 authentication
```

Caso a opção seja pela autenticação MD5:

```
R3(config-if)#router ospf 1  
R3(config-router)#area 0 authentication message-digest
```

Também é possível alterar os temporizadores OSPF para que ele responda mais rapidamente às mudanças na rede:

```
R1#show ip ospf interface fastEthernet 1/0  
  
FastEthernet1/0 is up, line protocol is up  
  
Internet Address 192.168.13.1/24, Area 0  
  
Process ID 1, Router ID 192.168.13.1, Network Type BROADCAST, Cost: 1  
  
Transmit Delay is 1 sec, State DR, Priority 1  
  
Designated Router (ID) 192.168.13.1, Interface address 192.168.13.1  
  
Backup Designated router (ID) 3.3.3.3, Interface address 192.168.13.3  
  
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
```

Observe acima os temporizadores (timer) padrões. A cada 10 segundos, um pacote hello é enviado, e se não recebemos nenhum pacote hello por 40 segundos, o roteador declara o vizinho como “morto”. Se for necessário, podemos alterar esses sinalizadores. Vamos alterar os timers entre R2 e R3:

```
R2(config-if)#interface fastEthernet 1/0
R2(config-if)#ip ospf hello-interval 5
R2(config-if)#ip ospf dead-interval 15
```

```
R3(config)#interface fastEthernet 0/0
R3(config-if)#ip ospf hello-interval 5
R3(config-if)#ip ospf dead-interval 15
```

Os comandos “ip ospf hello-interval” e ‘ip ospf dead-interval’ alteram esses temporizadores. Conforme explicado anteriormente, esses valores devem corresponder nas duas extremidades, ou não formaremos uma adjacência OSPF!

3.4.a Neighbor adjacencies

Falaremos agora sobre o processo de formação de vizinhança OSPF.

Pacotes OSPF e processo de descoberta de vizinho

OSPF usa seu próprio protocolo, não usa um protocolo de transporte como TCP ou UDP, o OSPF utiliza o **protocolo ID 89** para todos os seus pacotes.



Se usarmos o comando ‘**debug ip ospf packet**’ poderemos examinar os pacotes OSPF. Vejamos os diferentes campos que temos:

```
R2#debug ip ospf packet
OSPF packet debugging is on
OSPF: rcv. v:2 t:1 l:48 rid:1.1.1.1
aid:0.0.0.0 chk:4D40 aut:0 auk: from FastEthernet0/0
```

- **V:2** - Significa OSPF versão 2. Se estivermos executando o IPv6, a versão seria a V:3.
- **T:1** - Significa pacote OSPF número 1, que é o ‘hello packet’. Daqui a pouco veremos os diferentes tipos de pacotes.
- **L:48** - O comprimento do pacote em bytes. Este pacote de saudação (hello packet) possui 48 bytes.
- **RID 1.1.1.1** - O ID do roteador.
- **AID** – Aqui está o ID da área em pontos decimais. O ID da área pode ser escrito tanto em formato decimal (área 0) como decimal pontuado (área 0.0.0.0).
- **CHK 4D40** - É a soma de verificação (checksum) do pacote OSPF. Através do Checksum o roteador verifica se o pacote está corrompido ou não.
- **AUT:0** – Refere-se ao tipo de autenticação. O OSFP oferece 3 opções:
 - 0 = sem autenticação
 - 1 = texto não criptografado
 - 2 = MD5
 - **AUK:** Caso autenticação esteja habilitada, este campo terá algumas informações.

Vamos continuar examinando os diferentes tipos de pacote OSPF:

1. Hello
2. Database Description (DBD)
3. Link-State Request (LSR)
4. Link-State Update (LSU)
5. Link-State Acknowledgment (LSAck)

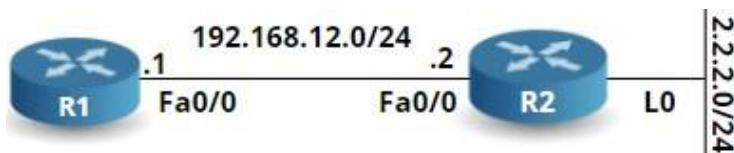
Aqui estão todos os **tipos de pacote OSPF** que temos. No ‘**debug ip ospf packet**’ na página anterior, você viu que o campo T:1 significa o tipo de pacote 1. Aqui você vê que ele corresponde a um pacote hello. Abaixo a função de cada pacote OSPF:

- **Hello:** Utilizado para descoberta de vizinhos, construção e manutenção de adjacências.
- **DBD:** Este pacote é usado para verificar se o LSDB entre 2 roteadores é o mesmo. O DBD é um **resumo do LSDB**.
- **LSR:** Solicita registros link-state específicos de um vizinho OSPF.
- **LSU:** Envia registros link-state específicos que foram solicitados. Este pacote é como um envelope com vários LSAs.
- **LSAck:** OSPF é um protocolo confiável, portanto ele utiliza pacotes de confirmação.

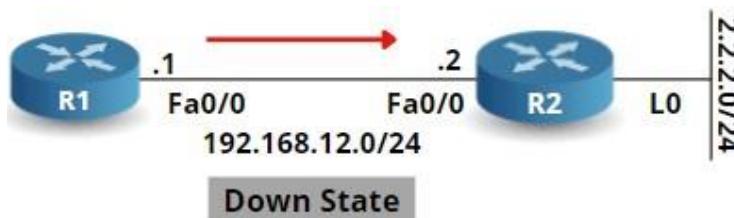
Os roteadores rodando o protocolo OSPF passam por 7 estados antes de estabelecer vizinhança:

- **Down:** Não há nenhum vizinho OSPF detectado.
- **Init:** O roteador entra nesse estado quando recebe “hello packet”.
- **Two-way:** O roteador encontra seu próprio ID no ‘hello packet’ que ele recebeu.
- **Exstart:** Determina qual roteador exercerá a função de mestre e a de escravo.
- **Exchange:** Os pacotes de descrição do banco de dados (**DBD**) são enviados.
- **Loading:** troca de pacotes LSRs (solicitação do estado do link) e LSUs (atualização do estado do link).
- **Full:** Os roteadores OSPF formam adjacência.

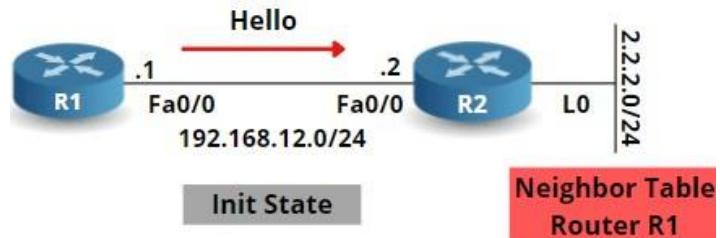
Vamos detalhar um pouco mais este processo, para isso, usaremos a topologia abaixo:



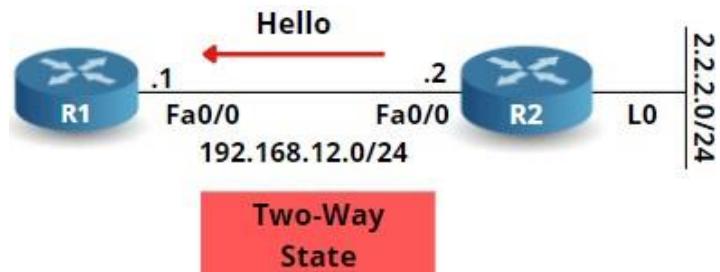
R1 e R2 estão conectados usando um único link, veremos todo o processo de aprendizado da rede 2.2.2.0/24 através do OSPF pelo roteador R1.



Assim que o OSPF é configurado no roteador R1, ele começa a enviar ‘Hello packets’. Nesse momento, R1 ainda sabe da existência de outros roteadores OSPF, então está no “**down state**”. O pacote hello é enviado ao **endereço multicast 224.0.0.5**.



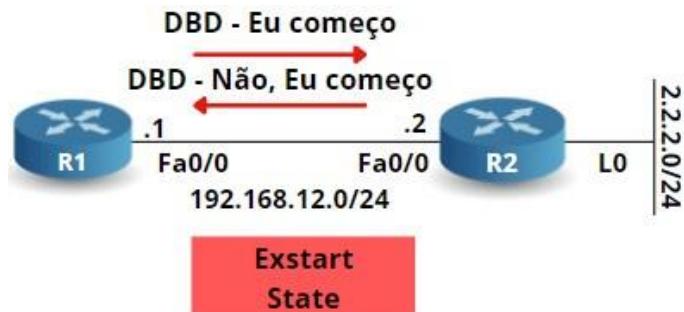
R2 recebe o “hello packet” e adiciona uma entrada para o roteador R1 na ‘OSPF neighbor table’ (tabela de vizinhos OSPF). Agora estamos no **init State**.



R2 tem que responder ao R1 com um ‘hello packet’. Este pacote não é enviado como **multicast**, mas como **unicast** e no “neighbor field” (campo vizinho) incluirá **todos os vizinhos OSPF** que R2 possui. R1 verá **seu próprio nome** no ‘campo vizinho’ do ‘hello packet’.

R1 receberá este hello packet e verá sua própria ID. Estamos agora no ‘**two-way state**’ (estado de mão dupla).

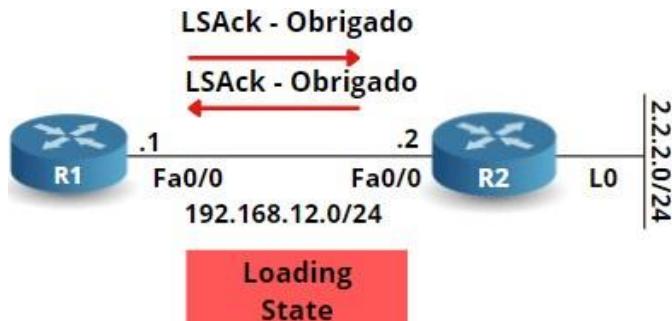
Tenho que fazer uma pequena observação: Se o link que estamos usando é uma **rede multiacesso**, o OSPF elegerá um **DR (roteador designado)** e um **BDR (roteador designado de backup)**. Só depois disso acontecerá que o roteador continuará com o processo OSPF.



Entramos agora no “**Exstart State**”. Nesse momento, os roteadores estão prontos para sincronizar seu LSDB. Agora, é selecionado qual roteador exercerá a função de “**master e slave**” (mestre e escravo). O roteador com a “router ID” mais alta se tornará o mestre. No caso em tela, R2 tem o ‘router ID’ mais alto e se tornará o mestre.



No **Exchange State (estado de troca)**, os roteadores enviam um DBD com um resumo do LSDB. Dessa forma, os roteadores descobrem quais redes eles não conhecem.



Quando os roteadores recebem o pacote DBD, eles executam os seguintes passos:

- Enviam uma confirmação usando o pacote LSAck.
- Comparam as informações no DBD com as que ele já possui:
 - Se o vizinho tiver informações novas ou mais recentes, ele enviará um pacote LSR (Link State Request) para solicitar essas informações.
 - Quando os roteadores começam a enviar o LSR (Link State Request), eles ficam no “Loading state” (estado de carregamento).
 - O outro roteador responderá com uma LSU (Link State Update) com as informações solicitadas.



Quando R1 solicitou informações sobre a rede 2.2.2.0/24, ele usou um LSR. R2 enviará um LSU com as informações sobre a rede solicitada, e o R1 enviará uma confirmação usando o pacote LSAck como confirmação.

Nesse momento, os roteadores estão em **full state** (estado pleno, completo). Ambos os roteadores têm os LSDBs sincronizados e estão prontos para rotear os pacotes.

Vamos fazer um debug no roteador para identificar esse passo a passo em um ambiente real! Usaremos o comando **debug ip ospf adj**:

```
R2#debug ip ospf adj
OSPF adjacency events debugging is on
```

```
R2#clear ip ospf process
Reset ALL OSPF processes? [no]: yes
```

Debug ligado e processo OSPF ‘resetado’, é hora de vermos a depuração ‘ao vivo’:

```
R2#  
  
OSPF: Interface Loopback0 going Down  
  
OSPF: 2.2.2.2 address 2.2.2.2 on Loopback0 is dead, state DOWN  
  
OSPF: Interface FastEthernet0/0 going Down  
  
OSPF: 2.2.2.2 address 192.168.12.2 on FastEthernet0/0 is dead, state DOWN  
  
OSPF: Neighbor change Event on interface FastEthernet0/0  
  
OSPF: DR/BDR election on FastEthernet0/0  
  
OSPF: Elect BDR 0.0.0.0  
  
OSPF: Elect DR 1.1.1.1  
  
OSPF: Elect BDR 0.0.0.0  
  
OSPF: Elect DR 1.1.1.1  
  
    DR: 1.1.1.1 (Id)    BDR: none  
  
OSPF: 1.1.1.1 address 192.168.12.1 on FastEthernet0/0 is dead, state DOWN  
  
%OSPF-5-ADJCHG: Process 1, Nbr 1.1.1.1 on FastEthernet0/0 from FULL to DOWN,  
Neighbor Down: Interface down or detached  
  
OSPF: Neighbor change Event on interface FastEthernet0/0  
  
OSPF: DR/BDR election on FastEthernet0/0  
  
OSPF: Elect BDR 0.0.0.0  
  
OSPF: Elect DR 0.0.0.0  
  
    DR: none    BDR: none  
  
OSPF: Remember old DR 1.1.1.1 (id)  
  
OSPF: Interface Loopback0 going Up  
  
OSPF: Interface FastEthernet0/0 going Up  
  
OSPF: 2 Way Communication to 1.1.1.1 on FastEthernet0/0, state 2WAY  
  
OSPF: Backup seen Event before WAIT timer on FastEthernet0/0  
  
OSPF: DR/BDR election on FastEthernet0/0  
  
OSPF: Elect BDR 2.2.2.2  
  
OSPF: Elect DR 1.1.1.1  
  
OSPF: Elect BDR 2.2.2.2  
  
OSPF: Elect DR 1.1.1.1  
  
    DR: 1.1.1.1 (Id)    BDR: 2.2.2.2 (Id)  
  
OSPF: Send DBD to 1.1.1.1 on FastEthernet0/0 seq 0x1E09 opt 0x52 flag 0x7 len 32
```

```

OSPF: Rcv DBD from 1.1.1.1 on FastEthernet0/0 seq 0x886 opt 0x52 flag 0x7 len 32
mtu 1500 state EXSTART

OSPF: First DBD and we are not SLAVE

OSPF: Rcv DBD from 1.1.1.1 on FastEthernet0/0 seq 0x1E09 opt 0x52 flag 0x2 len 72
mtu 1500 state EXSTART

OSPF: NBR Negotiation Done. We are the MASTER

OSPF: Send DBD to 1.1.1.1 on FastEthernet0/0 seq 0x1E0A opt 0x52 flag 0x1 len 32

OSPF: Rcv DBD from 1.1.1.1 on FastEthernet0/0 seq 0x1E0A opt 0x52 flag 0x0 len 32
mtu 1500 state EXCHANGE

OSPF: Exchange Done with 1.1.1.1 on FastEthernet0/0

OSPF: Send LS REQ to 1.1.1.1 length 24 LSA count 2

OSPF: Rcv LS UPD from 1.1.1.1 on FastEthernet0/0 length 108 LSA count 2

OSPF: Synchronized with 1.1.1.1 on FastEthernet0/0, state FULL

%OSPF-5-ADJCHG: Process 1, Nbr 1.1.1.1 on FastEthernet0/0 from LOADING to FULL,
Loading Done

```

Destaquei alguns campos: Observe a comunicação bidirecional (two way communication), a luta pelo poder para determinar quem será o mestre e o escravo (master e slave), a troca de resumos LSDBs usando pacotes DBD, e finalmente os pacotes LSQ e LSU.

Largura de banda de referência OSPF

Já falamos sobre esses cálculos na introdução geral sobre o OSPF, mas agora, é hora de aprofundarmos um pouco mais. Como dito anteriormente, o OSPF usa uma fórmula simples para calcular o custo para uma interface, a fórmula é:

$$\text{Custo} = \frac{\text{largura de banda de referência}}{\text{largura de banda de interface}}$$

A largura de banda de referência é um valor em Mbps, e pode até ser definida manualmente. Por padrão, o valor em roteadores Cisco é de 100 Mbps.

Vamos dar uma olhada em um exemplo:



O roteador acima tem duas interfaces, uma FastEthernet e uma interface serial:

R1#show ip interface brief					
Interface	IP-Address	OK?	Method	Status	
FastEthernet0/0	192.168.1.1	YES	manual	up	
Serial0/0	192.168.2.1	YES	manual	up	

Vamos habilitar o OSPF nestas interfaces:

```
R1(config)#router ospf 1  
R1(config-router)#network 192.168.1.0 0.0.0.255 area 0  
R1(config-router)#network 192.168.2.0 0.0.0.255 area 0
```

Após habilitarmos o OSPF, podemos verificar qual a largura de banda de referência:

```
Router#show ip ospf | include Reference  
Reference bandwidth unit is 100 mbps
```

Como esperado, por padrão o valor é 100 Mbps. Vamos ver quais valores de custo o OSPF calculou para as duas interfaces:

```
Router#show interfaces FastEthernet 0/0 | include BW  
MTU 1500 bytes, BW 100000 Kbit/sec, DLY 100 usec
```

```
Router#show ip ospf interface FastEthernet 0/0 | include Cost  
Process ID 1, Router ID 192.168.1.1, Network Type BROADCAST, Cost: 1
```

A interface FastEthernet tem uma largura de banda de 100.000 kbps (100 Mbps) e o custo OSPF é 1. A fórmula para calcular o custo é a seguinte:

$$\text{100.000 kbps reference bandwidth} / \text{100.000 interface bandwidth} = 1$$

E quanto à interface serial? Vamos descobrir:

```
R1#show interfaces Serial 0/0 | include BW  
MTU 1500 bytes, BW 1544 Kbit/sec, DLY 20000 usec,
```

```
R1#show ip ospf interface Serial 0/0 | include Cost  
Process ID 1, Router ID 192.168.2.1, Network Type POINT_TO_POINT, Cost: 64
```

A interface serial tem largura de banda de 1,544 kbps (1,5 Mbps) e custo de 64. Foi calculada assim:

$$\text{100.000 kbps reference bandwidth} / \text{1.544 kbps interface bandwidth} = 64,76$$

O custo foi arredondado para 64.

A largura de banda de referência padrão é de 100 Mbps, o que pode causar problemas caso estejamos usando interfaces Gigabit ou superior. O menor valor de custo possível é 1, portanto, com a largura de banda de referência padrão, uma interface FastEthernet, Gigabit ou superior teria o custo OSPF de 1.

Se você usar interfaces Gigabit ou superior, é melhor alterar a largura de banda de referência:

```
Router(config-router)#auto-cost reference-bandwidth ?  
<1-4294967> The reference bandwidth in terms of Mbits per second
```

Com o comando **auto-cost reference-bandwidth** podemos especificar o valor que desejamos em Mbps. Vamos configura-lo para 1.000 Mbps:

```

Router(config-router)#auto-cost reference-bandwidth 1000
% OSPF: Reference bandwidth is changed.

Please ensure reference bandwidth is consistent across all routers

```

O IOS informará que devemos nos preocupar com essa configuração em todos os roteadores OSPF. Vamos verificar:

```

Router#show ip ospf | include Reference
Reference bandwidth unit is 1000 mbps

```

A largura de banda de referência agora é de 1.000 Mbps, vamos ver qual é o custo do link FastEthernet agora:

```

Router#show ip ospf interface FastEthernet 0/0 | include Cost
Process ID 1, Router ID 192.168.1.1, Network Type BROADCAST, Cost: 10
Topology-MTID      Cost      Disabled      Shutdown      Topology Name

```

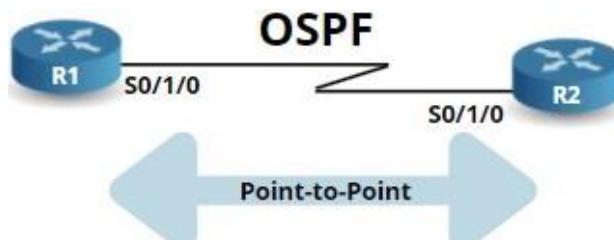
A interface agora tem um custo de 10, o que significa que uma interface Gigabit teria um custo de 1.

3.4.b Point-to-point

Quando o OSPF opera em links seriais ponto a ponto usando protocolos WAN de camada dois como HDLC e PPP, ele é executado como um ‘tipo’ de rede ponto a ponto.

Neste modo, as funções DR/BDR não são necessárias, pois não é uma conexão multi acesso.

Configuração da interface ponto a ponto OSPFv2:



Vamos configurar o Roteador 1 e 2 como point-to-point. Primeiro, vamos configurar o OSPF no R1 e R2 nas interfaces S1/0/1:

```

R1(config-router)#router ospf 1
R1(config-router)#network 0.0.0.0 0.0.0.0 area 0

```

A seguir, configuraremos o link serial para point-to-point. O comando é ‘**ip ospf network point-to-point**’

```

R1(config)#int se0/1/0
R1(config-if)#ip ospf network point-to-point

```

Vamos utilizar o comando ‘**show ip ospf interface s0/1/0**’ para verificarmos se o ‘network type’ (tipo de rede), e como esperado deve ser ‘point to point’:

```

Serial0/1/0 is up, line protocol is up
Internet Address 192.168.13.1/30, Area 0
Process ID 1, Router ID 111.111.111.111, Network Type POINT_TO_POINT, Cost: 64

```

```

Transmit Delay is 1 sec, State POINT_TO_POINT,
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
oob-resync timeout 40
Hello due in 00:00:02
Supports Link-local Signaling (LLS)
Index 3/3, flood queue length 0
Next 0x0(0)/0x0(0)
Last flood scan length is 1, maximum is 1
Last flood scan time is 4 msec, maximum is 4 msec
Neighbor Count is 1, Adjacent neighbor count is 1
Adjacent with neighbor 133.133.133.133
Suppress hello for 0 neighbor(s)

```

Depois de configurarmos o OSPF no outro roteador, as adjacências se formaram. Observe que não há informações de DR na saída **show ip ospf neighbour**. Isso ocorre porque o tipo de rede não suporta eleições de DR/BDR.

```

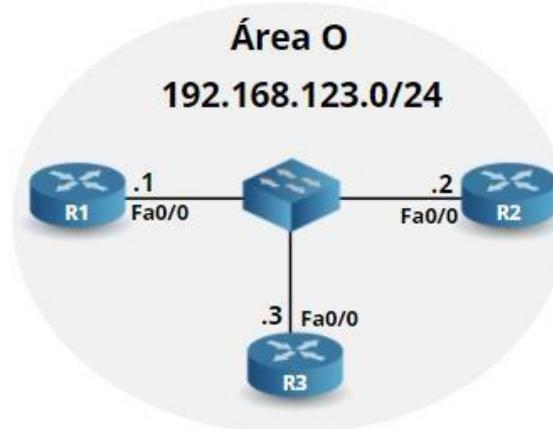
R1(config-if)#do show ip ospf neighbor
Neighbor ID Pri State Dead Time Address Interface
133.133.133.133 0 FULL/ - 00:00:30 192.168.13.2 Serial0/1/0

```

3.4.c Broadcast (DR/BDR selection)

O OSPF usa DR (Roteador designado) e BDR (Roteador designado de backup) em cada rede multiacesso. Uma rede multiacesso é um segmento (rede) que possui mais de dois roteadores. O OSPF descobre isso observando o tipo de interface. Por exemplo, uma interface Ethernet é considerada uma rede multiacesso e uma interface serial é considerada uma interface ponto a ponto.

A maioria dos alunos do CCNA pensa que esta eleição do DR/BDR é feita por área, mas isso está errado. Vamos aprender como a eleição é realizada e como podemos influenciá-la. Esta é a topologia que usaremos:



Neste exemplo temos 3 roteadores em uma rede Ethernet rodando o protocolo OSPF. Eles estão conectados ao mesmo switch (rede multiacesso), portanto, haverá a escolha do DR/BDR. O OSPF foi configurado para que todos os roteadores se tornassem vizinhos OSPF, vamos dar uma olhada:

R1#show ip ospf neighbor					
Neighbor ID	Pri	State	Dead Time	Address	Interface
192.168.123.2	1	FULL/BDR	00:00:32	192.168.123.2	FastEthernet0/0
192.168.123.3	1	FULL/DR	00:00:31	192.168.123.3	FastEthernet0/0

Da perspectiva do R1, R2 é o BDR e R3 é o DR.

R3#show ip ospf neighbor					
Neighbor ID	Pri	State	Dead Time	Address	Interface
192.168.123.1	1	FULL/DROTHER	00:00:36	192.168.123.1	FastEthernet0/0
192.168.123.2	1	FULL/BDR	00:00:39	192.168.123.2	FastEthernet0/0

Quando um roteador não é o DR ou BDR, ele é chamado de DROTHER. Observe que R1 é um DROTHER.

R2#show ip ospf neighbor					
Neighbor ID	Pri	State	Dead Time	Address	Interface
192.168.123.1	1	FULL/DROTHER	00:00:31	192.168.123.1	FastEthernet0/0
192.168.123.3	1	FULL/DR	00:00:32	192.168.123.3	FastEthernet0/0

Já o roteador R2 (o BDR) vê o DR e o DROTHER.

Claro que podemos mudar qual roteador será o DR/BDR alterando as prioridades. Vamos transformar o R1 no DR:

```
R1(config)#interface fastEthernet 0/0
R1(config-if)#ip ospf priority 200
```

É possível alterar a prioridade usando o comando ‘**ip ospf priority**’:

- A prioridade padrão é 1.
- Uma prioridade de 0 significa que o roteador nunca será eleito como DR ou BDR.
- É preciso usar o comando ‘**clear ip ospf process**’ para ‘resetar’ o processo OSPF e as alterações tenham efeito.

R1#show ip ospf neighbor					
Neighbor ID	Pri	State	Dead Time	Address	Interface
192.168.123.2	1	FULL/BDR	00:00:31	192.168.123.2	FastEthernet0/0
192.168.123.3	1	FULL/DR	00:00:32	192.168.123.3	FastEthernet0/0

Observe que o R3 ainda é o DR, precisamos resetar as adjacências dos vizinhos OSPF, como dito anteriormente, para eleger o novo DR e BDR.

```
R3#clear ip ospf process
Reset ALL OSPF processes? [no]: yes
```

```
R2#clear ip ospf process
Reset ALL OSPF processes? [no]: yes
```

Com todas as adjacências redefinidas, é hora de verificarmos novamente:

R1#show ip ospf neighbor					
Neighbor ID	Pri	State	Dead Time	Address	Interface
192.168.123.2	1	FULL/DROTHER	00:00:36	192.168.123.2	FastEthernet0/0
192.168.123.3	1	FULL/BDR	00:00:30	192.168.123.3	FastEthernet0/0

Agora, o R1 se tornou o DR, sabemos disso porque os outros roteadores são DROTHER e BDR. Podemos confirmar no R3 que R1 é o DR e que a prioridade dele é 200:

R3#show ip ospf neighbor					
Neighbor ID	Pri	State	Dead Time	Address	Interface
192.168.123.1	200	FULL/DR	00:00:30	192.168.123.1	FastEthernet0/0
192.168.123.2	1	FULL/DROTHER	00:00:31	192.168.123.2	FastEthernet0/0

Lembre-se: A **escolha do DR/BDR é por segmento multi-acesso e não por área!**

3.4.d Router ID

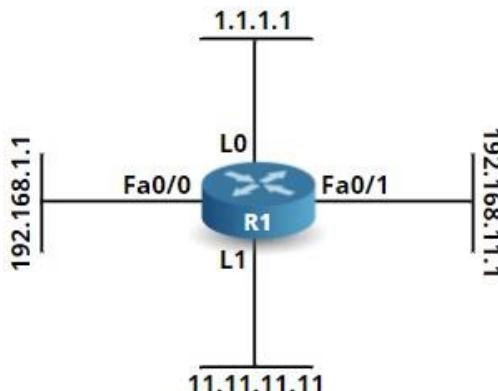
Cada roteador OSPF seleciona um router ID (ID de roteador (RID)) que deve ser exclusivo na rede. O OSPF armazena a topologia da rede em seu LSDB (Banco de dados de estado de link) e cada roteador é identificado com sua router ID exclusiva. Se houver router IDs duplicados na rede, haverá problemas de acessibilidade.

Por causa disso, dois roteadores OSPF com a mesma router ID não se tornarão vizinhos, mas ainda é possível ter router IDs duplicados na rede desde que com roteadores que não estejam diretamente conectados um ao outro.

O OSPF usa os seguintes critérios para selecionar a ID do roteador:

1. Configuração manual do router ID.
2. Endereço IP mais alto em uma interface de loopback.
3. Endereço IP mais alto em uma interface sem loopback.

Observe o exemplo abaixo, usarei o seguinte roteador para esta demonstração:



Existem duas interfaces físicas e duas interfaces de loopback. Todas as interfaces estão ativas:

R1#show ip interface brief					
Interface	Protocol	IP-Address	OK?	Method	Status
Fa0/0					

FastEthernet0/0	192.168.1.1	YES manual up	up
FastEthernet0/1	192.168.11.1	YES manual up	up
Loopback0	1.1.1.1	YES manual up	up
Loopback1	11.11.11.11	YES manual up	up

Vamos iniciar um processo OSPF:

```
R1(config)#router ospf 1
R1(config-router)#exit
```

Vamos checar qual router ID foi selecionado:

```
R1#show ip protocols | include Router ID
Router ID 11.11.11.11
```

Foi selecionado o 11.11.11.11, que é o endereço IP mais alto de uma interface de loopback. Vamos apagar as interfaces loopbacks e ver o que acontecerá:

```
R1(config)#no interface loopback 0
R1(config)#no interface loopback 1
```

Observe novamente:

```
R1#show ip protocols | include Router ID
Router ID 11.11.11.11
```

A router ID permanece a mesmo, isso ocorre porque a seleção do router ID é realizada apenas uma vez. Após essa primeira eleição, é necessário ‘resetar’ o processo OSPF para que haja uma nova eleição:

```
R1#clear ip ospf process
Reset ALL OSPF processes? [no]: yes
```

Agora sim, podemos olhar novamente que veremos a diferença:

```
R1#show ip protocols | include Router ID
Router ID 192.168.11.1
```

A router ID agora é o endereço IP mais alto das interfaces físicas. Caso seja necessário, é possível definir manualmente a router ID como o comando ‘router-id’:

```
R1(config)#router ospf 1
R1(config-router)#router-id 111.111.111.111
```

Vamos verificar nosso trabalho:

```
R1#show ip protocols | include Router ID
Router ID 111.111.111.111
```

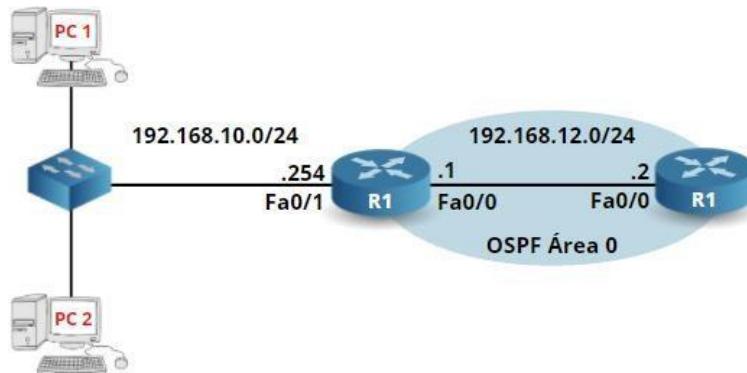
Quando o roteador ainda não possui nenhuma adjacência formada, o OSPF alterara imediatamente a ID do roteador (caso tenha sido utilizado o comando router-id), quando houver vizinhança estabelecida, é necessário resetar o processo OSPF.

OSPF Passive Interface

Quando aplicamos o comando ‘network’ no OSPF, duas coisas acontecem:

- Todas as interfaces que possuem uma rede dentro do intervalo do comando ‘network’ serão anunciadas no OSPF.
- Os ‘hello packets’ OSPF serão enviados por essas interfaces.

Às vezes, é indesejável enviar ‘hello packets’ em certas interfaces. Dê uma olhada na imagem abaixo:



Os roteadores R1 e R2 estão configurados com OSPF. R1 está conectado à rede 192.168.10.0/24 e possui alguns computadores conectados a um switch. R1 deseja anunciar esta rede para R2.

Assim que usarmos o comando ‘network’ para anunciar a rede 192.168.10.0/24 no OSPF, R1 também enviará “hello packets” para o switch. Isso é uma má ideia, não só porque não há roteadores nesta rede, mas também porque é um risco à segurança. Explico, se alguém em algum computador iniciar um aplicativo que responda os ‘hellos packets’ do OSPF, R1 tentará formar vizinhança. Um invasor pode anunciar rotas falsas usando essa técnica.

Para evitar que isso aconteça, podemos usar o comando’ **passive-interface**’. Este comando diz ao OSPF para não enviar “hello packets” em certas interfaces. Vamos ver como isso funciona:

Observe a configuração OSPF no R1 e R2:

```
R1(config)#router ospf 1
R1(config-router)#network 192.168.12.0 0.0.0.255 area 0
R1(config-router)#network 192.168.10.0 0.0.0.255 area 0
```

```
R2(config)#router ospf 1
R2(config-router)#network 192.168.12.0 0.0.0.255 area 0
```

Com a configuração acima, R2 aprenderá sobre rede 192.168.10.0/24:

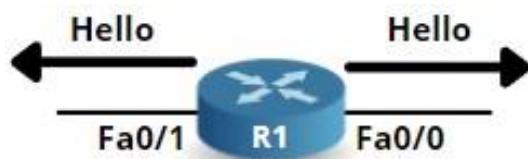
```
R2#show ip route ospf
0    192.168.10.0/24 [110/20] via 192.168.12.1, 00:03:21, FastEthernet0/0
```

Isso é ótimo, mas há um efeito colateral dessa configuração: R1 enviará hello packets pela interface FastEthernet 0/1. Podemos ver isso com o debug:

```
R1#debug ip ospf hello
OSPF hello events debugging is on
OSPF: Send hello to 224.0.0.5 area 0 on FastEthernet0/1 from 192.168.10.254
```

```
OSPF: Send hello to 224.0.0.5 area 0 on FastEthernet0/0 from 192.168.12.1
```

Acima você pode ver que os hello packets são enviados em ambas as direções.



Vamos consertar isso configurando o OSPF para interromper o envio de hello packets em direção ao switch:

```
R1(config)#router ospf 1
R1(config-router)#passive-interface FastEthernet 0/1
```

O comando ‘**passive-interface**’, mas a interface desejada, interrompe o processo. Podemos verificar através do comando “show ip protocols”:

```
R1#show ip protocols

Routing Protocol is "ospf 1"

    Outgoing update filter list for all interfaces is not set
    Incoming update filter list for all interfaces is not set
    Router ID 192.168.12.1
    Number of areas in this router is 1. 1 normal 0 stub 0 nssa
    Maximum path: 4
    Routing      for      Networks:
        192.168.10.0 0.0.0.255 area 0
        192.168.12.0 0.0.0.255 area 0
    Reference bandwidth unit is 100 mbps

Passive Interface(s):
FastEthernet0/1

    Routing Information Sources:
        Gateway          Distance      Last Update
        Distance: (default is 110)
```

O comando ‘show ip protocols’ informará quais interfaces estão configuradas como interface (s) passiva (s). É possível também observar pelo debug.

```
R1#
OSPF: Send hello to 224.0.0.5 area 0 on FastEthernet0/0 from 192.168.12.1
```

Observe que agora ele só está enviando pela interface fa0/0.



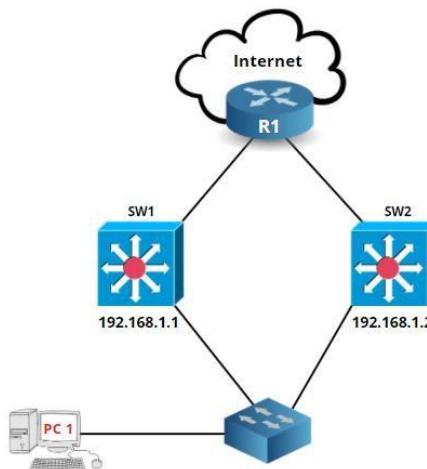
É possível configurar o roteador para que a interface no modo passivo seja o modo default do roteador, essa configuração é bastante útil caso tenhamos várias interfaces para bloquear e apenas algumas para liberar os ‘hello packets’:

```
R1(config)#router ospf 1
R1(config-router)#passive-interface default
R1(config-router)#no passive-interface FastEthernet 0/0
```

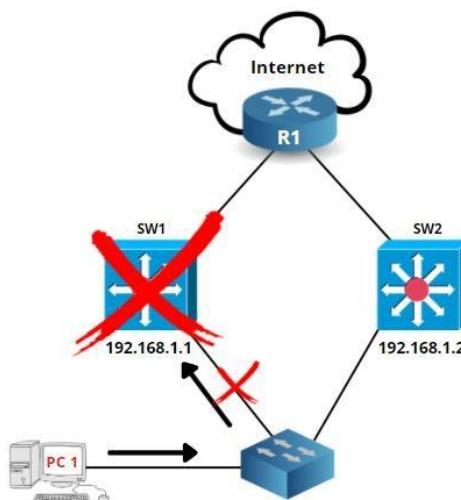
No exemplo acima, colocamos todas as interfaces como passiva, com exceção da fa0/0.

3.5 Describe the purpose of first hop redundancy protocol

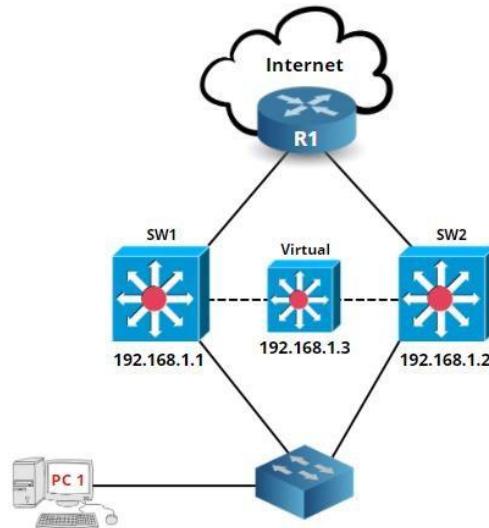
Neste tópico falaremos sobre o que é FHRP (First hop redundancy protocol), ou protocolo de redundância de gateway. Vamos começar com um exemplo!



A topologia acima é bem simples. Há um computador conectado a um switch. Esse switch está ligado a outros dois switches multilayer (SW1 e SW2). Esses switches layer 3 possuem endereços IPs que podem ser usados como o default gateway do computador. Mais acima do SW1 e SW2 há um roteador conectado à Internet. Qual gateway deverá ser configurado no dispositivo? SW1 ou SW2? Afinal, só pode haver um gateway configurado no dispositivo.



Caso o escolhido seja o SW1 e ele travar por qualquer motivo, o computador não conseguirá sair da sua própria sub-rede, porque ele conhece apenas um default gateway. Para resolver este problema, criamos um **gateway virtual**:



Entre os switches SW1 e SW2, criaremos um gateway virtual, que terá seu próprio endereço IP, no exemplo acima o endereço escolhido foi o 192.168.1.3.

O computador usará 192.168.1.3 como seu default gateway. Sendo assim, um dos switches será o gateway ativo e, em caso de falha, o outro assumirá.

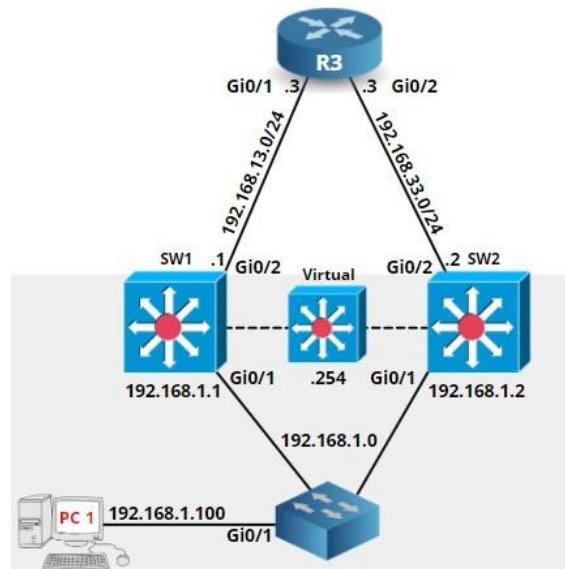
Existem três protocolos diferentes que podemos usar para criar um gateway virtual:

- **HSRP** (Hot Standby Routing Protocol)
- **VRRP** (Virtual Router Redundancy Protocol)
- **GLBP** (Gateway Load Balancing Protocol)

Embora dois desses protocolos sejam proprietários, o HSRP e GLPB, a Cisco só cobra no CCNA o HSRP. Já o VRRP foi criado na RFC 5798.

HSRP (Hot Standby Routing Protocol)

Vamos entender como o HSRP (Hot Standby Routing Protocol) funciona nos mínimos detalhes e como devemos configura-lo. Para isso, usaremos a topologia abaixo:



Eis o que temos:

- SW1 e SW2 são switches multicomadas. A sub-rede 192.168.1.0/24 pertence à VLAN 1, e há um host conectado nessa rede.
- O endereço IP 192.168.1.254 será usado como o endereço do gateway virtual.
- Os switches multicomadas estão conectados às interfaces de camada três ao roteador R3.

O primeiro passo, é habilitar o HSRP. Faremos isso nas interfaces VLAN 1 do SW1 e SW2:

```
SW1 & SW2
(config)#interface Vlan 1
(config-if)#standby 1 ip 192.168.1.254
```

O comando para habilitar o HSRP é o ‘**standby**’. O número “1” é o número do grupo HSRP. Podemos escolher qualquer número, contanto que ele seja igual em ambos os dispositivos que farão parte daquele grupo. E por último, colocamos o endereço 192.168.1.254 como endereço do gateway virtual. Após configurarmos o HSRP em ambos os switches a seguinte mensagem aparecerá na tela:

```
SW1#
%HSRP-5-STATECHANGE: Vlan1 Grp 1 state Standby -> Listen
%HSRP-5-STATECHANGE: Vlan1 Grp 1 state Speak -> Standby
```

```
SW2#
%HSRP-5-STATECHANGE: Vlan1 Grp 1 state Standby -> Active
```

Um dos switches será o gateway ativo (active) e o outro entrará em modo de espera (standby). Vamos ver se temos conectividade com gateway virtual a partir do computador:

```
PC1#ping 192.168.1.254
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.254, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 6/206/1007 ms
```

Como você pode ver, conseguimos pingar com sucesso o endereço IP do gateway virtual.

Apenas um comando em cada switch é suficiente para o HSRP funcionar! No entanto, há algumas outras coisas que devemos examinar. Usamos 192.168.1.254 como endereço IP virtual, mas qual endereço MAC ele usará?

```
R1#show ip arp | include 1.254
Internet 192.168.1.254          1  0000.0c07.ac01  ARPA  GigabitEthernet0/1
```

Você pode ver o endereço MAC de 192.168.1.254 na tabela ARP, mas, de onde veio esse endereço MAC?

0000.0c07.ac01 é o endereço MAC que encontramos. O HSRP usa o endereço MAC **0000.0c07.acXX** onde **XX** é o número do grupo HSRP. No exemplo acima, configuramos o grupo com o número 1. Há algumas outras coisas interessantes a serem verificadas, dê uma olhada abaixo:

```

SW1#show standby

Vlan1 - Group 1

State is Standby

    3 state changes, last state change 00:03:33

Virtual IP address is 192.168.1.254

Active virtual MAC address is 0000.0c07.ac01 (MAC Not In Use)

    Local virtual MAC address is 0000.0c07.ac01 (v1 default)

Hello time 3 sec, hold time 10 sec

    Next hello sent in 0.144 secs

Preemption disabled

Active router is 192.168.1.2, priority 100 (expires in 7.776 sec)

Standby router is local

Priority 100 (default 100)

Group name is "hsrp-Vl1-1" (default)

```

```

SW2#show standby

Vlan1 - Group 1

State is Active

    2 state changes, last state change 00:04:25

Virtual IP address is 192.168.1.254

Active virtual MAC address is 0000.0c07.ac01 (MAC In Use)

    Local virtual MAC address is 0000.0c07.ac01 (v1 default)

Hello time 3 sec, hold time 10 sec

    Next hello sent in 0.992 secs

Preemption disabled

Active router is local

Standby router is 192.168.1.1, priority 100 (expires in 10.640 sec)

Priority 100 (default 100)

Group name is "hsrp-Vl1-1" (default)

```

Com o comando **show standby** é possível verificar a configuração do HSRP, inclusive algumas informações importantes:

- Ele informa qual o endereço IP virtual (192.168.1.254).
- Mostra o endereço MAC virtual (0000.0c07.ac01).

- É possível ver o status, se o switch está ativo ou em modo de espera.
- Os temporizadores, por exemplo o ‘hello’ é de 3 segundos e o tempo de espera (hold timer) é de 10 segundos.
- Se a preempção (preemption) está desativada ou não.

O switch que estiver no modo ativo **responderá às solicitações ARP** dos hosts e encaminhará os pacotes enviados pelos computadores. Ele também é o responsável por enviar mensagens de ‘hello’ para os switches que estão no modo de espera (standby). Os dispositivos em modo de espera ouvirão as mensagens ‘hello’; se não receberem nada do switch ativo, eles aguardarão o **tempo de espera expirar** antes de assumir o controle. O tempo de espera por padrão é de 10 segundos, o que é considerado lento para os padrões atuais; veremos como podemos modificá-lo em seguida.

Cada switch HSRP passará por vários estados antes de terminar como ativo ou em espera:

Estado:	Explicação:
Initial	Este é o primeiro estado, ocorre quando o HSRP é iniciado.
Listen	Neste estágio, o switch conhece o endereço de IP virtual e está ouvindo mensagens ‘hello’ dos outros dispositivos HSRP.
Speak	O switch encaminha mensagens de ‘hello’ e participa da eleição para definir qual dispositivo será o ‘active’ e ‘standby’.
Standby	O switch não se tornou o ‘active’, porém, ele continuará encaminhando pacotes ‘hello’. Caso haja alguma falha com o switch ‘ativo’ ele tomará o controle.
Active	O switch encaminhará os pacotes vindo dos clientes e continuará encaminhando mensagens ‘hello’ para os switches que estejam no modo de standby.

Podemos acompanhar todas essas etapas com o comando debug. Vamos derrubar a interface VLAN 1, para que possamos reiniciar o HSRP com o debug ativado:

```
SW1 & SW2
#debug standby events
HSRP Events debugging is on
```

```
SW1 & SW2
(config)#interface Vlan 1
(config-if)#shutdown
```

Vamos habilitar a interface VLAN 1 no SW1 primeiro:

```
SW1
(config)#interface Vlan 1
(config-if)#no shutdown
```

Eis as informações de debug que teremos:

```
SW1#
HSRP: V11 Interface UP
HSRP: V11 Starting minimum intf delay (1 secs) - uptime 997
HSRP: V11 Intf min delay expired - uptime 998
HSRP: V11 Grp 1 Init: a/HSRP enabled
HSRP: V11 Grp 1 Init -> Listen
```

```

HSRP: Vl1 Grp 1 Redundancy "hsrp-Vl1-1" state Init -> Backup
HSRP: Vl1 IP Redundancy "hsrp-Vl1-1" update, Init -> Backup
HSRP: Vl1 Grp 1 Listen: d/Standby timer expired (unknown)
HSRP: Vl1 Grp 1 Listen -> Speak
HSRP: Vl1 Grp 1 Redundancy "hsrp-Vl1-1" state Backup -> Speak
HSRP: Vl1 IP Redundancy "hsrp-Vl1-1" update, Backup -> Speak
HSRP: Vl1 Grp 1 Speak: d/Standby timer expired (unknown)
HSRP: Vl1 Grp 1 Standby router is local
HSRP: Vl1 Grp 1 Speak -> Standby
HSRP: Vl1 Grp 1 Redundancy "hsrp-Vl1-1" state Speak -> Standby
HSRP: Vl1 IP Redundancy "hsrp-Vl1-1" standby, unknown -> local
HSRP: Vl1 IP Redundancy "hsrp-Vl1-1" update, Speak -> Standby
HSRP: Vl1 Grp 1 Standby: c/Active timer expired (unknown)
HSRP: Vl1 Grp 1 Active router is local
HSRP: Vl1 Grp 1 Standby router is unknown, was local
HSRP: Vl1 Grp 1 Standby -> Active
HSRP: Vl1 Grp 1 Redundancy "hsrp-Vl1-1" state Standby -> Active
HSRP: Vl1 Grp 1 Added 192.168.1.254 to ARP (0000.0c07.ac01)
HSRP: Vl1 Grp 1 Activating MAC 0000.0c07.ac01
HSRP: Vl1 Grp 1 Adding 0000.0c07.ac01 to MAC address filter - resetting the
interface
HSRP: Vl1 IP Redundancy "hsrp-Vl1-1" standby, local -> unknown
HSRP: Vl1 IP Redundancy "hsrp-Vl1-1" update, Standby -> Active
HSRP: Vl1 IP Redundancy "hsrp-Vl1-1" update, Active -> Active

```

É possível identificar claramente os diferentes estados pelos quais o switch passou antes de terminar no estado ‘ativo’. No momento, SW1 é o único switch que está executando HSRP, vamos habilitar a interface VLAN 1 no SW2:

```

SW2(config)#interface Vlan 1
SW2(config-if)#no shutdown

```

Eis o resultado do debug:

```

SW2#
HSRP: Vl1 Grp 1 Active router is 192.168.1.1
HSRP: Vl1 Nbr 192.168.1.1 created
HSRP: Vl1 Nbr 192.168.1.1 active for group 1

```

```

HSRP: Vl1 Interface UP

HSRP: Vl1 Starting minimum intf delay (1 secs) - uptime 1089

HSRP: Vl1 Intf min delay expired - uptime 1090

HSRP: Vl1 Grp 1 Init: a/HSRP enabled

HSRP: Vl1 Grp 1 Init -> Listen

HSRP: Vl1 Grp 1 Redundancy "hsrp-Vl1-1" state Init -> Backup

HSRP: Vl1 IP Redundancy "hsrp-Vl1-1" update, Init -> Backup

HSRP: Vl1 Grp 1 Listen: d/Standby timer expired (unknown)

HSRP: Vl1 Grp 1 Listen -> Speak

HSRP: Vl1 Grp 1 Redundancy "hsrp-Vl1-1" state Backup -> Speak

HSRP: Vl1 IP Redundancy "hsrp-Vl1-1" update, Backup -> Speak

HSRP: Vl1 Grp 1 Speak: d/Standby timer expired (unknown)

HSRP: Vl1 Grp 1 Standby router is local

HSRP: Vl1 Grp 1 Speak -> Standby

HSRP: Vl1 Grp 1 Redundancy "hsrp-Vl1-1" state Speak -> Standby

HSRP: Vl1 IP Redundancy "hsrp-Vl1-1" standby, unknown -> local

HSRP: Vl1 IP Redundancy "hsrp-Vl1-1" update, Speak -> Standby

```

Acima, podemos ver que o SW2 está vendo 192.168.1.1 (SW1) como o dispositivo ativo, sendo assim, ele entra no estado de espera.

Processo de eleição para o Active Gateway

Por que SW2 entrou no modo standby em vez de SW1?

Por padrão, o switch com a **prioridade mais alta** se tornará o dispositivo HSRP ativo. Se a prioridade for a mesma, o **endereço IP mais alto** será o fator de desempate. Vamos dar uma olhada nas prioridades:

```

SW1#show standby | include Priority

Priority 100 (default 100)

```

```

SW2#show standby | include Priority

Priority 100 (default 100)

```

A prioridade é a mesma em ambos os switches, porém, o SW2 possui o endereço IP mais alto, então ele deveria se tornar o dispositivo ativo, mas mesmo assim não se tornou. Vamos aumentar sua prioridade:

```

SW2(config)#interface Vlan 1

SW2(config-if)#standby 1 priority 150

```

Vamos verificar novamente a prioridade:

```
SW2#show standby | include Priority
```

```
Priority 150 (configured 150)
```

```
SW2#show standby | include Active
```

```
Active virtual MAC address is 0000.0c07.ac01 (MAC Not In Use)
```

```
Active router is 192.168.1.1, priority 100 (expires in 9.232 sec)
```

Mesmo o SW2 com a prioridade mais alta, SW1 permaneceu como o ativo, vamos entender o porquê em instantes. Outro comando útil para verificar qual roteador está ativo ou em modo de espera é o comando **show standby brief**:

```
SW1#show standby brief
```

```
P indicates configured to preempt.
```

```
|
```

Interface	Grp	Pri	P	State	Active	Standby	Virtual IP
Vl1	1	100		Active	local	192.168.1.2	192.168.1.254

```
SW2#show standby brief
```

```
P indicates configured to preempt.
```

```
|
```

Interface	Grp	Pri	P	State	Active	Standby	Virtual IP
Vl1	1	150		Standby	192.168.1.1	local	192.168.1.254

É possível confirmar que o SW2 tem a prioridade mais alta, mas SW1 ainda está com estado de ativo. Depois que o HSRP decide qual dispositivo é o ativo, ele permanecerá como ativo até que haja alguma falha e ele fique ‘down’. Vamos aprender como podemos mudar esse comportamento.

Preemption

Quando habilitamos o preemption, o switch com a prioridade mais alta (ou endereço IP, caso a prioridade seja a mesma) sempre se tornará o dispositivo ativo. Veja como habilitar, mas antes a definição do dicionário para o que é **preempção** na área de TI: “*Em um ambiente multitarefa, ação ou evento que causa mudança do processamento de uma aplicação para outra.*”

```
SW1 & SW2
```

```
(config)#interface Vlan 1
```

```
(config-if)#standby 1 preempt
```

Vamos ver se a aplicação desse comando fez realmente a diferença:

```
SW1#show standby brief
```

```
P indicates configured to preempt.
```

```
|
```

Interface	Grp	Pri	P	State	Active	Standby	Virtual IP
-----------	-----	-----	---	-------	--------	---------	------------

```
Vl1           1     100 P Standby 192.168.1.2      local          192.168.1.254
```

```
SW2#show standby brief
```

P indicates configured to preempt.

|

Interface	Grp	Pri	P	State	Active	Standby	Virtual IP
Vl1	1	150	P	Active	local	192.168.1.1	192.168.1.254

Agora, depois do comando aplicado, o SW2 está em modo ativo e SW1 foi para o modo de espera!

Por padrão, a preempção entrará em vigor imediatamente, mas pode ser uma boa ideia colocar um pequeno delay. Se um switch for reinicializado, pode levar algum tempo para “convergir”. Talvez o OSPF ou o EIGRP precisem formar adjacências vizinhas ou o STP ainda não esteja pronto para desbloquear as portas. Por isso, vamos adicionar um pequeno delay:

```
SW1 & SW2  
(config)#interface Vlan 1  
(config-if)#standby 1 preempt delay minimum 60
```

Com o comando ‘standby 1 preempt delay minimum 60’ atrasaremos a preempção em 60 segundos.

Autenticação

O HSRP também oferece suporte à autenticação. Temos dois modos: Autenticação de texto simples ou MD5. Veja como configurar o MD5:

```
SW1 & SW2  
(config)#interface Vlan 1  
(config-if)#standby 1 authentication md5 key-string Senha123
```

Isso garante que todos os pacotes enviados entre os dois switches sejam autenticados, evitando que alguém não autorizado, que esteja na sub-rede 192.168.1.0/24, entre na configuração do HSRP.

Temporizadores HSRP

Por padrão, o HSRP é muito lento. SW1 é o nosso dispositivo em espera, e ele aguardará 10 segundos (hold time) antes de se tornar ativo quando houver alguma falha no SW2. Isso significa que teremos 10 segundos de tempo de inatividade, vamos ver como podemos acelerar isso:

```
SW1(config-if)#standby 1 timers ?  
<1-254> Hello interval in seconds  
msec      Specify hello interval in milliseconds
```

Podemos acelerar as coisas alterando os temporizadores com o comando **standby timers**. Podemos até usar valores de milissegundos:

```
SW1 & SW2  
(config)#interface Vlan 1  
(config-if)#standby 1 timers msec 100 msec 300
```

Definimos o tempo de ‘hello’ para 100 milissegundos e o hold time para 300 milissegundos. As boas práticas recomendam que o tempo de espera seja de pelo menos **três vezes o temporizador do hello**. Vamos verificar:

```
SW1#show standby | include time  
Hello time 100 msec, hold time 300 msec
```

```
SW2#show standby | include time  
Hello time 100 msec, hold time 300 msec
```

HSRP Versão 1 e 2

Existem duas versões do HSRP dependendo do modelo do roteador ou switch. Podemos alterar a versão usando o comando ‘**standby version**’. Vamos mudar nossos dispositivos para a versão 2:

```
SW1 & SW2  
(config)#interface Vlan 1  
(config-if)#standby version 2
```

```
SW1#show standby | include version  
Vlan1 - Group 1 (version 2)
```

Quadro comparativo entre as duas versões do HSRP:

	HSRPv1	HSRPv2
Número dos grupos	0 - 255	0 – 4095
Endereço virtual do MAC	0000.0c07.acXX (XX = número do grupo)	0000.0c9f.fxxx (XX = número do grupo)
Endereço Multicast	224.0.0.2	224.0.0.102

Com isso encerramos o terceiro grande bloco que a Cisco dividiu o novo CCNA, passamos por todos os tópicos abordados nesse bloco.

Exercícios:

1. Um engenheiro configurou um 'vizinho OSPF' como um 'roteador designado'. Qual status indica se o 'roteador designado' está no modo adequado?
 - a) Exchange
 - b) 2-way
 - c) Full
 - d) Init
2. Um administrador de redes configurou OSPF entre dois roteadores utilizando interface serial, tanto o R1 quanto R2 estão utilizando PPP. Por default, qual 'network type' será mostrado quando o comando 'show ip ospf interface' for aplicado nos roteadores?
 - a) port-to-multipoint
 - b) broadcast
 - c) point-to-point
 - d) nonbroadcast
3. Um administrador de redes configurou OSPF na interface gigabit Ethernet de dois roteadores. Por default, qual a 'network type' dessa interface?
 - a) point-to-multipoint
 - b) point-to-point
 - c) nonbroadcast
 - d) broadcast
4. Quais os dois resultados abaixo são comportamentos esperado do HSRP? (Escolha dois)
 - a) Dois ou mais roteadores compartilham um endereço IP virtual que será usado como default gateway para os dispositivos da LAN.
 - b) Dois ou mais roteadores negociam para que um roteador se torne o 'active router' e o outro 'standby router'.
 - c) Cada roteador HSRP possui um endereço IP diferente, todos eles atuam como default gateway da LAN e o tráfego será平衡ado entre eles.
 - d) Dois ou mais roteadores sincronizam suas configurações para fornecer encaminhamento de pacotes consistente.
 - e) Dois ou mais roteadores compartilham o mesmo endereço IP e há balanceamento de carga no tráfego para o default gateway.
5. Quando uma rota estática flutuante (floating static route) é configurada, qual ação garante que a rota de backup seja usada quando a rota primária falhar?
 - a) A rota estática flutuante deve ter uma distância administrativa maior que a rota principal, desse modo ela será utilizada como backup.
 - b) A distância administrativa deve ser maior na rota principal para que a rota de backup se torne secundária
 - c) A rota estática flutuante (floating static route) deve ter uma distância administrativa menor do que a rota principal, para que seja usada como backup.
 - d) O comando 'default-information originate' deve ser aplicado no roteador para a rota a ser instalada na tabela de roteamento.
6. Um roteador executando EIGRP aprendeu a mesma rota através de dois caminhos diferentes. Qual parâmetro o roteador usará para selecionar o melhor caminho?
 - a) Cost
 - b) administrative distance
 - c) metric
 - d) as-path
7. R1 aprendeu a rota 192.168.12.0/24 via IS-IS, OSPF, RIP e 'Internal EIGRP'. Em condições normais de operação, qual protocolo de roteamento está instalado na tabela de roteamento?
 - a) IS-IS
 - b) RIP
 - c) Internal EIGRP
 - d) OSPF
8. Qual endereço MAC é reconhecido como endereço virtual do VRRP?
 - a) 0000.5E00.010^a
 - b) 0005.3711.0975
 - c) 0000.0C07.AC99
 - d) 0007.C070/AB01
9. Quando o OSPF aprende vários caminhos diferentes para uma mesma rede, como ele seleciona a rota que irá usar?
 - a) Multiplica o valor do K por 256 para calcular a rota com a métrica mais baixa.
 - b) Para cada interface existente, ele adiciona a métrica do roteador de origem até o destino para calcular a rota com a largura de banda mais baixa.

- c) Divide a largura de banda de referência (100 Mbps) pela largura de banda real da interface, calculando assim o roteador com o custo mais baixo.
- d) Conta o número de saltos entre o roteador de origem e o destino para determinar o roteador com a métrica mais baixa.
10. Qual atributo um roteador usa para selecionar o melhor caminho quando duas ou mais rotas diferentes para o mesmo destino existem vindo de dois protocolos de roteamento diferentes?
- a) dual algorithm
- b) metric (métrica)
- c) administrative distance (distância administrativa)
- d) hop count (contagem de saltos)
11. R1 aprendeu a mesma rota de dois vizinhos diferentes, um dos roteadores vizinhos é um vizinho OSPF e o outro é um vizinho EIGRP. Qual a distância administrativa da rota que será instalada na tabela de roteamento?
- a) 20
- b) 90
- c) 110
- d) 115
12. Qual opção abaixo é um endereço IPv6 válido?
- a) 2001:0000:130F::099a::12^a
- b) 002:7654:A1AD:61:81AF:CCC1
- c) FEC0:ABCD:WXYZ:0067::2A4
- d) 2004:1:25A4:886F::1
13. Quais são os dois recursos do IPv6?
- a) Anycast
- b) Broadcast
- c) Multicast
- d) Podcast
- e) Allcast
14. Qual protocolo abaixo é um protocolo de roteamento de vetor de distância?
- a) IS-IS
- b) OSPF
- c) BGP
- d) EIGRP
15. Qual comando você deve ser aplicado para garantir que um roteador HSRP com prioridade mais alta se torne o roteador HSRP primário após ser reiniciado?
- a) standby 10 preempt
- b) standby 10 version 1
- c) standby 10 priority 150
- d) standby 10 version 2
16. Qual comando deve ser aplicado em um roteador HSRP para que sua interface local se torne ativa se todos os outros roteadores do grupo falharem?
- a) standby 1 track ethernet
- b) não é necessário nenhum comando adicional
- c) standby 1 preempt
- d) standby 1 priority 250

Resposta: 1) c, 2) c, 3) d, 4) a, b 5) a, 6) c, 7) c, 8) a, 9) c, 10) c, 11) b, 12) d, 13) a, c 14) d 15) a 16) b

4.0 IP Services

Hora de entrarmos nos serviços IPs. Esse bloco trata de assuntos e protocolos que são bem diferentes entre si, aqui nós vamos desde NAT até QOS, é um grande balaio de gato!

4.1 Configure and verify inside source NAT using static and pools

Falaremos sobre tradução de endereços IPs, essa explicação se conecta com o início do livro, quando falamos de endereços públicos e privados.

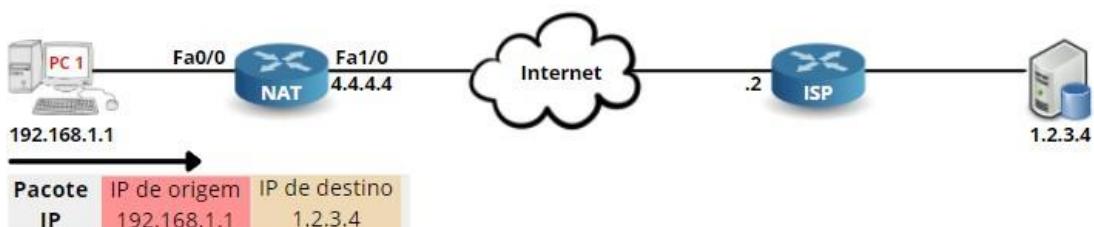
Introdução ao NAT

Sem a tradução dos endereços de rede (Network address translation - NAT), provavelmente não conseguiríamos acessar a Internet da nossa casa, ou pelo menos seríamos o único dentro de casa com acesso à Internet! Neste tópico vamos entender o porquê e como usamos NAT para acessar à Internet.

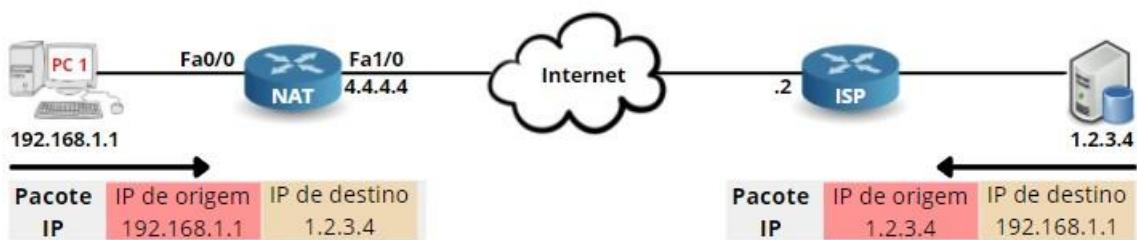
Vamos usar a topologia abaixo para exemplificar a necessidade do NAT:



No lado esquerdo temos um computador com o endereço IP 192.168.1.1 conectado a um roteador. O ISP entregou o endereço IP 4.4.4.4, e há um servidor na Internet usando o endereço IP 1.2.3.4. Se o computador enviar algo para esse servidor, qual será o endereço IP de origem e de destino do pacote IP que ele enviará?



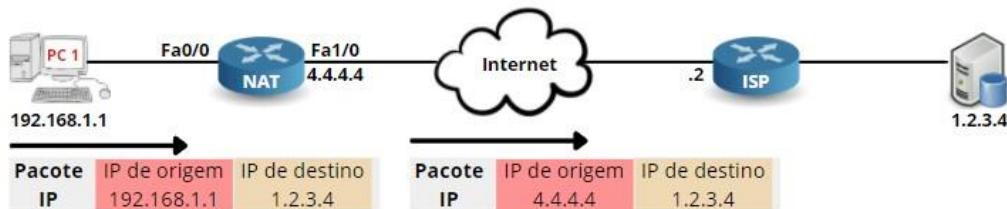
O endereço IP de origem será o endereço do computador e o endereço IP de destino será o endereço do servidor, como você pode ver no pacote IP da imagem acima.



Assim que o servidor responder, ele criará um pacote IP especificando o endereço IP do computador como destino, e o endereço IP de origem será seu próprio endereço IP.

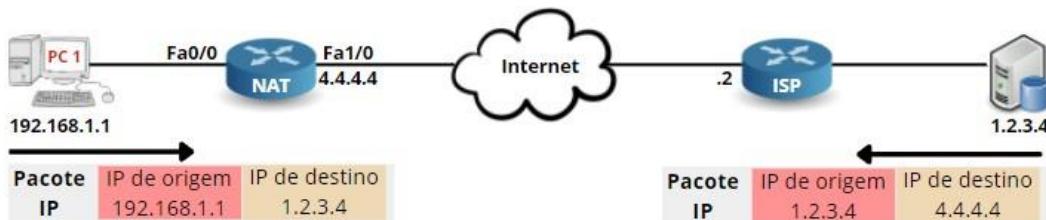
Há algo de errado com este exemplo? Não, está perfeitamente bem, exceto por um detalhe! O endereço IP do computador é um endereço privado. Você deve lembrar que os endereços IPs privados são destinados apenas às LANs e os endereços IPs públicos à Internet.

Para que essa comunicação seja possível, precisamos configurar o NAT (Network Address Translation), observe abaixo:



A história é praticamente igual, o computador irá enviar um pacote para o servidor, mas agora o roteador foi configurado para realizar o NAT. De forma que o endereço IP 192.168.1.1 seja traduzido para o endereço IP 4.4.4.4.

Eis o que acontece: O roteador NAT irá reescrever o endereço IP de origem de 192.168.1.1 para 4.4.4.4 como você pode ver no pacote IP acima.



O servidor pensa que está se comunicando com o endereço IP 4.4.4.4, razão pela qual vemos esse endereço IP como o destino no pacote IP que ele está enviando.

Assim que esse pacote IP chegar ao roteador, ele examinará novamente sua tabela NAT e traduzirá o endereço IP 4.4.4.4 de volta para 192.168.1.1 e o enviará para o computador.

O exemplo que acabei de mostrar é chamado de **NAT estático**. Existe uma relação 1:1 entre o endereço IP do computador na LAN e o endereço IP que obtivemos do ISP. Então, o que acontece se tivermos mais computadores na LAN? Nesse caso, podemos usar o chamado **NAT dinâmico**.

O NAT dinâmico é diferente do NAT estático porque:

- Você pode usar um pool de endereços IP para tradução.
- Você pode usar uma lista de acesso para corresponder aos hosts na LAN que devem ser traduzidos.

Vamos exemplificar: Na topologia anterior, utilizamos NAT estático para traduzir o endereço do computador para o endereço IP 4.4.4.4 fornecido pelo ISP. Porém, vamos imaginar que o ISP invés de nos dar um único endereço IP, forneceu toda a sub-rede 4.4.4.0/24.

Além do nosso computador 192.168.1.1, existem outros 10 computadores que precisam de acesso à Internet. O que acontecerá agora? Afinal, temos um pool de endereços IPs cedidos pelo ISP que podemos usar para fazer a tradução.

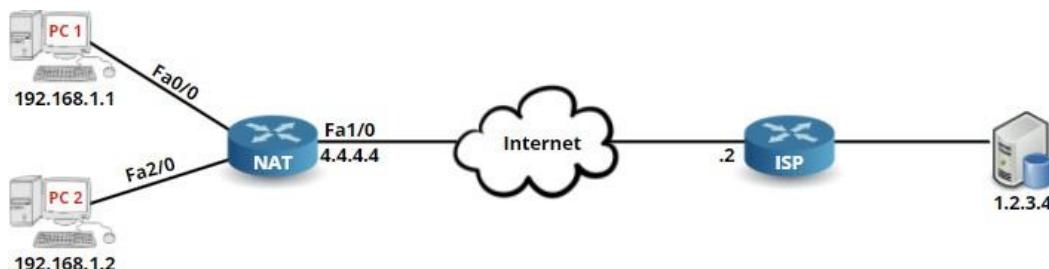
Vamos discutir com um exemplo:

1. O computador com IP 192.168.1.1 está visitando um site na Internet, o roteador NAT irá traduzir este endereço IP para o primeiro endereço IP do pool: 4.4.4.1.
2. O próximo computador com IP 192.168.1.2 está agora visitando um outro site na Internet, o roteador NAT irá traduzir este endereço IP para o segundo endereço IP do pool, 4.4.4.2.
3. O terceiro computador com IP 192.168.1.3 também está visitando algum site na Internet, o roteador NAT irá traduzir este endereço IP para o terceiro endereço IP do pool, 4.4.4.3.
4. Etc.

Isso é o que chamamos de NAT dinâmico.

Se você nunca viu NAT, talvez esteja tudo meio confuso até aqui. Afinal, você provavelmente tem mais de um dispositivo em sua casa acessando a Internet e você só obteve um único endereço IP do seu provedor.

É aqui que introduzimos o **PAT** ou **Port Address Translation** (**tradução de endereço por porta**). O NAT nos dá apenas uma relação 1:1 entre dois endereços IPs. Se tivermos vários computadores em nossa LAN e apenas um único endereço IP fornecido pelo ISP, precisaremos traduzir os números das portas. Dessa forma, podemos ter vários computadores ‘atrás’ de um único endereço IP público. Vamos dar uma olhada em um exemplo:



Observe a rede acima, temos dois computadores na LAN com os endereços IP 192.168.1.1 e 192.168.1.2. O roteador está configurado para NAT:

A seguinte situação está acontecendo:

1. O computador com endereço IP 192.168.1.1 vai se conectar ao servidor.
2. O roteador NAT irá traduzir 192.168.1.1 para 4.4.4.4.
3. O outro computador com endereço IP 192.168.1.2 também irá se conectar ao servidor.
4. O roteador NAT agora tem um problema, pois 192.168.1.1 já está traduzido para 4.4.4.4. Você não pode ter dois endereços IPs iguais.

É aqui que o PAT entra em ação, eis o que vai acontecer:

1. O computador com endereço IP 192.168.1.1 vai se conectar ao servidor.
2. O roteador NAT irá traduzir 192.168.1.1 para 4.4.4.4, mas também manterá o controle das portas de origem e destino!
3. O outro computador com endereço IP 192.168.1.2 também irá se conectar ao servidor.
4. Uma vez que nosso roteador NAT também faz PAT, ele irá traduzir 192.168.1.2 para 4.4.4.4, mas irá usar outro número de porta de origem.

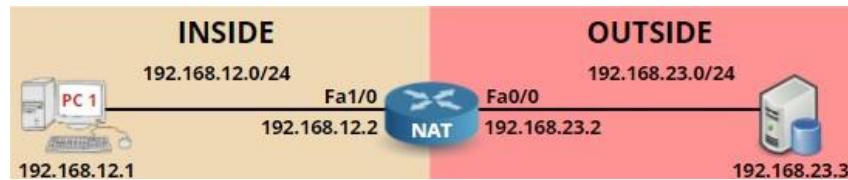
E é assim que é possível ter vários computadores na LAN e fazer com que todos eles acessem a Internet por meio de um único endereço IP público.

O servidor pensa que está se comunicando apenas com o 4.4.4.4, então não tem ideia de que existe um computador com o endereço IP 192.168.1.1 ou 192.168.1.2. Isso significa que podemos dizer que o NAT ou PAT é um protocolo de segurança? Este é um grande debate, mas na minha opinião não é um mecanismo de segurança. Não ver os verdadeiros hosts na LAN não significa que você não consegue acessá-los. Assim que seu roteador estiver fazendo a tradução de rede e / ou endereço de porta, esses hosts estarão acessíveis. Segurança é algo que implementamos usando listas de acesso, firewalls, sistemas de prevenção de intrusão e políticas de segurança.

Como o NAT e/ou PAT mudam o pacote IP, acaba gerando alguma incompatibilidade com alguns aplicativos que não lidam muito bem com tradução de endereços IP e portas, o IPSEC é um exemplo, o FTP também é problemático quando há um IP por trás de um roteador NAT.

Configuração do NAT Estático

Vamos aprender como configurar o NAT estático em um roteador Cisco. Esta é a topologia que usaremos:



Na topologia acima, temos três dispositivos chamados PC1, Roteador NAT e Servidor_Web. Imagine que nosso host está na LAN e o servidor da web está em algum lugar da Internet. O roteador no meio é nossa conexão com a Internet.

O computador é capaz de alcançar o roteador Web:

```
PC1#ping 192.168.23.3

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.23.3, timeout is 2 seconds:
!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 8/8/12 ms
```

Observe essa demonstração:

```
Servidor_Web#debug ip packet

IP packet debugging is on
```

Se utilizarmos o comando **debug ip packet** conseguimos ver os pacotes IP que estamos recebendo no roteador Servidor_Web. **NÃO** faça isso em um ambiente de produção pois você sobrecarregará o roteador! Agora vamos enviar o ping novamente:

```
Servidor_Web#
IP: s=192.168.12.1 (FastEthernet0/0), d=192.168.23.3, len 100, rcvd 1
```

Observe que o roteador recebeu um pacote IP com endereço de origem 192.168.12.1 e endereço IP de destino 192.168.23.3.

Ele responderá com um pacote IP que possui o endereço de origem 192.168.23.3 e o endereço de destino 192.168.12.1.

Agora vamos configurar o NAT para vermos ver a diferença:

```
NAT(config)#interface fastEthernet 1/0
NAT(config-if)#ip nat inside
```

```
NAT(config)#interface fastEthernet 0/0
NAT(config-if)#ip nat outside
```

A configuração do Nat não é complexa: Primeiro, temos que configurar e informar ao roteador qual é a interface interna (inside) e externa (outside). A interface ‘inside’ é a que fica do lado da “LAN”, por isso fica do lado de dentro (inside). O servidor está “na Internet”, portanto, está fora de nossa rede (outside). Agora podemos configurar nossa regra de NAT estático:

```
NAT(config)#ip nat inside source static 192.168.12.1 192.168.23.2
```

Usamos o comando **ip nat inside** para traduzir um endereço IP interno (192.168.12.1) para um endereço IP externo (192.168.23.2).

```
NAT#show ip nat translations  
Pro Inside global      Inside local      Outside local      Outside global  
--- 192.168.23.2      192.168.12.1      ---      ---
```

Podemos verificar a configuração com o comando ‘**show ip nat translations**’. Vamos enviar outro ping e verificarmos como o pacote chega até o Servidor_Web:

```
PC1#ping 192.168.23.3  
  
Type escape sequence to abort.  
  
Sending 5, 100-byte ICMP Echos to 192.168.23.3, timeout is 2 seconds:  
!!!!!  
  
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/8/12 ms
```

```
Servidor_Web#  
IP: s=192.168.23.2 (FastEthernet0/0), d=192.168.23.3, len 100, rcvd 1
```

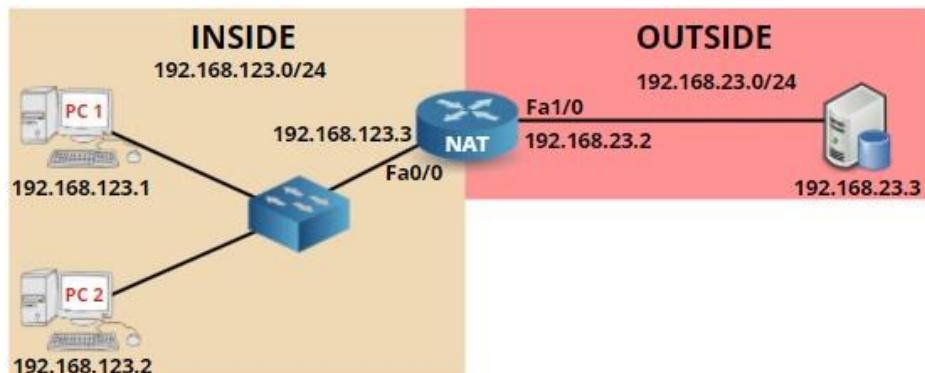
Agora, o pacote que o Servidor_Web recebe do PC1 tem o endereço IP de origem 192.168.23.2.

```
Servidor_Web#  
IP: tableid=0, s=192.168.23.3 (local), d=192.168.23.2 (FastEthernet0/0), routed  
via RIB
```

E quando ele responde, o endereço IP de destino é 192.168.23.2. Sabemos com certeza que o NAT estático está funcionando.

NAT Dinâmico

É hora de configurarmos o NAT dinâmico, onde usamos um pool de endereços IP para tradução. Usaremos uma topologia bem simples com dois hosts e 1 roteador que executará o NAT:



A primeira etapa é configurar o NAT informando e configurando no roteador a interface interna e externa:

```
NAT(config)#interface fastEthernet 0/0
NAT(config-if)#ip nat inside
NAT(config)#interface fastEthernet 1/0
NAT(config-if)#ip nat outside
```

Agora criaremos um pool com endereços IP que podemos usar para a tradução:

```
NAT(config)#ip nat pool MEU_POOL 192.168.23.10 192.168.23.20 prefix-length 24
```

O comando ‘**ip nat pool**’ nos permite criar um pool. Coloquei o nome de “MEU_POOL” e estamos usando o endereço IP 192.168.23.10 até 192.168.23.20. Agora podemos selecionar os hosts que queremos traduzir:

```
NAT(config)#access-list 1 permit 192.168.123.0 0.0.0.255
```

A lista de acesso acima corresponde à rede 192.168.123.0/24. É a rede onde o PC1 e o PC2 estão localizados. A última etapa é unir a lista de acesso e o pool juntos:

```
NAT(config)#ip nat inside source list 1 pool MEU_POOL
```

O comando acima seleciona a ‘access-list 1’ como fonte dos endereços que serão traduzidos para o “MEU_POOL”. Isso garante que o PC1 e o PC2 sejam convertidos para um endereço IP do pool. Agora vamos verificar nossa configuração!

```
PC1#ping 192.168.23.3
Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.23.3, timeout is 2 seconds:
!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 8/12/28 ms
```

PC1 obteve sucesso ao executar ping para o Servidor_Web, agora vamos dar uma olhada nas traduções no roteador NAT:

```
NAT#show ip nat translations
Pro Inside global      Inside local        Outside local       Outside global
icmp 192.168.23.10:3   192.168.123.1:3    192.168.23.3:3    192.168.23.3:3
--- 192.168.23.10      192.168.123.1       ---             ---
```

Como você pode ver, o PC1 foi traduzido para o endereço IP 192.168.23.10. Agora vamos enviar algum tráfego do PC2 para ver se haverá alguma diferença na tabela NAT:

```
PC2#ping 192.168.23.3
Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.23.3, timeout is 2 seconds:
!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 8/9/16 ms
```

NAT#show ip nat translations			
Pro	Inside global	Inside local	Outside local
			Outside global
icmp	192.168.23.10:4	192.168.123.1:4	192.168.23.3:4
---	192.168.23.10	192.168.123.1	---
icmp	192.168.23.11:2	192.168.123.2:2	192.168.23.3:2
---	192.168.23.11	192.168.123.2	---

O PC2 foi traduzido para o endereço IP 192.168.2.11. Excelente, nosso NAT dinâmico está funcionando!

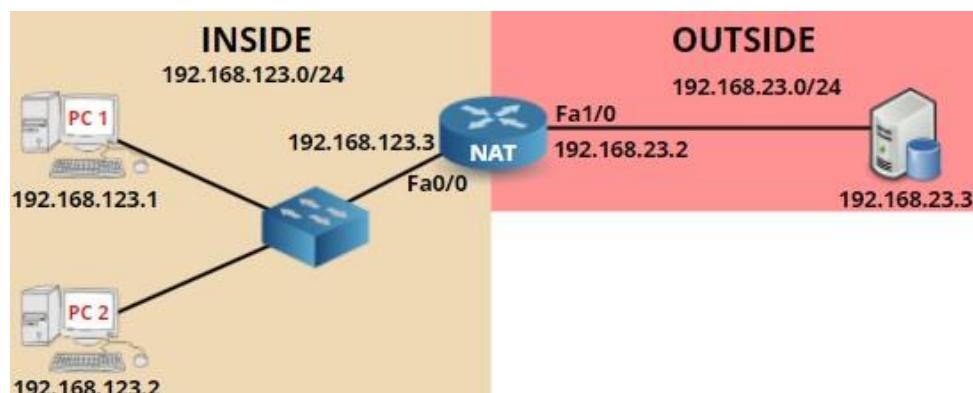
Observe que existem alguns endereços na lista com o nome de: Inside global, Inside local, Outside local, Outside global, vamos entender o que significa cada um deles:

- Inside Global é o endereço IP na interface externa do seu roteador executando o NAT.
- Inside local é o endereço IP de um de seus hosts internos que é traduzido com NAT.
- Outside local é o endereço IP do dispositivo que você está tentando acessar, em nosso exemplo, o Servidor_Web.
- Outside global também é o endereço IP do dispositivo que você está tentando acessar, em nosso exemplo, o Servidor_Web.

Por que os endereços outside local e outside global são iguais? Bem, esta resposta está fora do escopo do exame CCNA, mas com NAT é possível traduzir não só de “dentro” para “fora”. É possível criar uma entrada no roteador para que sempre que um dos hosts enviar um ping para um endereço IP, digamos 5.5.5.5, ele seja encaminhado para Servidor_Web. Neste exemplo, o “servidor web outside” é “localmente” visto por nossos hosts como 5.5.5.5, não 192.168.23.3.

Port Address Translation

Usaremos a topologia abaixo para falar do PAT:



A primeira etapa é configurar o NAT definindo quais são as interfaces ‘inside’ e ‘outside’

```
NAT(config)#interface fastEthernet 0/0
NAT(config-if)#ip nat inside
```

```
NAT(config)#interface fastEthernet 1/0
NAT(config-if)#ip nat outside
```

Vamos criar uma lista de acesso que corresponda a ambos os hosts:

```
NAT(config)#access-list 1 permit 192.168.123.0 0.0.0.255
```

E finalmente configurar o PAT:

```
NAT(config)#ip nat inside source list 1 interface fastEthernet 1/0 overload
```

Selecionamos a Access-list 1 como fonte interna e vou traduzi-los para o endereço IP da FastEthernet 1/0. A palavra mágica aqui é **overload (Sobrecarga)**. Basta adicionarmos ela para habilitarmos o PAT!

Vamos fazer um teste!

Para verificarmos o número da porta, não vamos usar o ping, vamos nos conectar à porta TCP 80 do Servidor_Web, mas antes, vamos habilitar o servidor WEB no roteador (sim, estou usando um roteador como servidor web):

```
Servidor_Web(config)#ip http server
```

Agora, podemos usar telnet para conectar à porta 80:

```
PC1#telnet 192.168.23.3 80
Trying 192.168.23.3, 80 ... Open
```

```
PC2#telnet 192.168.23.3 80
Trying 192.168.23.3, 80 ... Open
```

Como você pode ver, está escrito “open”, o que significa que conectamos com sucesso à porta 80.

Vamos ver como esta nossa tabela NAT/PAT:

```
NAT#show ip nat translations
      Pro   Inside global        Inside local        Outside local        Outside global
tcp  192.168.23.2:46369  192.168.123.1:46369  192.168.23.3:80  192.168.23.3:80
tcp  192.168.23.2:50669  192.168.123.2:50669  192.168.23.3:80  192.168.23.3:80
```

O roteador mantém o registro do número da porta em que ambos os hosts são traduzidos para o endereço IP 192.168.23.2. Missão cumprida!

Telnet é um ótimo comando para se conectar a diferentes portas TCP. Você pode usá-lo para testar listas de acesso ou conectividade, ou como no exemplo, para testar NAT/PAT.

4.2 Configure and verify NTP operating in a client and server mode

O NTP (Network Time Protocol) é usado para permitir que os dispositivos de rede sincronizem seus relógios com um relógio central. Para dispositivos de rede como roteadores, switches ou firewalls isso é muito importante, afinal, queremos ter certeza que as informações de registro e os carimbos de data/hora estejam com as horas e a datas exatas. Se você tiver problemas de rede ou for hackeado, certifique-se de saber exatamente o que e **quando** aconteceu.

Normalmente, um roteador ou switch será executado no modo cliente NTP, o que significa que ele ajustará seu relógio com base na hora de um servidor NTP. Basicamente, o protocolo NTP descreve o algoritmo que os clientes NTP usam para sincronizar seus relógios com o servidor NTP e os pacotes que são usados entre eles.

Um bom exemplo de servidor NTP é o ntp.pool.org. Este é um cluster de servidores NTP que muitos servidores e dispositivos de rede usam para sincronizar seus relógios.

NTP usa um conceito chamado “stratum” que define quantos saltos NTP um dispositivo está de uma fonte de tempo autorizada. Por exemplo, um dispositivo com ‘stratum 1’ é um dispositivo muito preciso, que pode ter um relógio atômico anexado a ele. Outro servidor NTP que esteja usando este servidor ‘stratum 1’ para sincronizar seu próprio relógio será um dispositivo ‘stratum 2’, porque está um salto NTP mais longe da origem. Quando você configura vários servidores NTP no mesmo cliente, o preferido será o servidor NTP com o valor de ‘stratum’ mais baixo.

Os roteadores e switches Cisco podem usar 3 modos NTP diferentes:

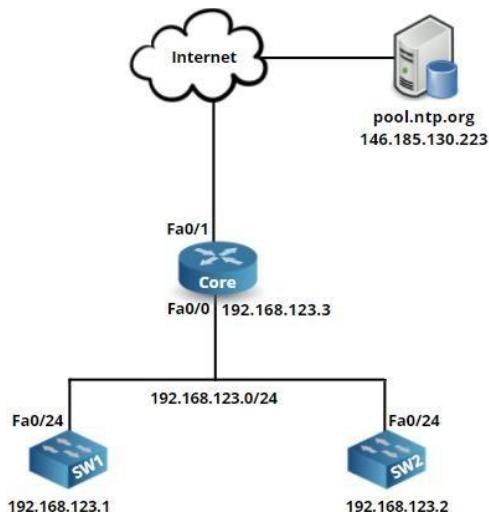
- NTP Client Mode (Modo cliente NTP)
- NTP Server Mode (Modo de servidor NTP).
- NTP Symmetric Mode (Modo simétrico ativo NTP).

O ‘NTP Symmetric Mode’ serve para sincronizar dispositivos NTP uns com os outros, é usado como um mecanismo de backup quando os clientes não conseguem alcançar o servidor NTP (externo).

Agora que entendemos a parte teórica, é hora de aprendermos a configurar o NTP em roteadores e switches Cisco.

Configuração

Usaremos a topologia abaixo:



O roteador na parte superior da topologia é chamado de “CoreRouter”, ele é a conexão de saída da Lan para Internet. Ele usará um dos servidores NTP do pool.ntp.org para sincronizar seu relógio. A rede também possui dois switches que precisam estar com os relógios sincronizados. Ambos os switches se tornarão clientes NTP do CoreRouter, tornando assim o CoreRouter um servidor NTP.

Configuração do Roteador

Primeiro, vamos configurar o CoreRouter. Como dito, usaremos o pool.ntp.org como servidor NTP externo. Mas antes de tudo, precisamos ter certeza que o roteador é capaz de resolver nomes de host, para isso, usaremos o DNS do Google:

```
CoreRouter(config)#ip name-server 8.8.8.8
```

Agora, podemos direcionar o roteador para o servidor: pool.ntp.org

```
CoreRouter(config)#ntp server pool.ntp.org
```

A configuração do NTP é bem simples, apenas um comando e o relógio interno do roteador será sincronizado com o do servidor. Podemos verificar a sincronização com o comando ‘**show ntp associations**’:

```
CoreRouter#show ntp associations
```

```

address          ref clock      st  when   poll  reach  delay  offset  disp
~146.185.130.223 .INIT.        16     -     64     0  0.000  0.000 16000.
* sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured

```

A saída do comando ‘**show ntp associations**’ nos diz se o relógio está sincronizado ou não. No caso em tela, o ~ na frente do endereço IP nos diz que o servidor está configurado, porém, ainda *não* está *sincronizado*. Você pode ver isso porque não há * na frente do endereço IP e o campo “st” (estratum) atualmente é 16.

Existe mais um comando que nos fornece mais informações sobre a configuração do NTP:

```

CoreRouter#show ntp status

Clock is unsynchronized, stratum 16, no reference clock

nominal freq is 250.0000 Hz, actual freq is 250.0000 Hz, precision is 2**24

reference time is 00000000.00000000 (00:00:00.000 UTC Mon Jan 1 1900)

clock offset is 0.0000 msec, root delay is 0.00 msec

root dispersion is 0.16 msec, peer dispersion is 0.00 msec

loopfilter state is 'FSET' (Drift set from file), drift is 0.000000000 s/s

system poll interval is 64, never updated.

```

O roteador nos diz que não estamos sincronizados e que não há relógio de referência, esse processo é demorado mesmo, mas basta esperar alguns minutos, e quando digitarmos os comandos novamente já veremos a diferença:

```

CoreRouter#show ntp associations

address          ref clock      st  when   poll  reach  delay  offset  disp
*~146.185.130.223 193.79.237.14    2     26     64     1 10.857 -5.595 7937.5
* sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured

```

Observe que agora a saída mudou. O * na frente do endereço IP nos diz que o dispositivo está sincronizado com stratum 2! Isso significa que este servidor NTP está muito próximo de uma fonte de tempo confiável. O campo “poll” nos diz que tentaremos sincronizar a cada 64 segundos. Vamos verificar novamente com o comando ‘**show ntp status**’:

```

CoreRouter#show ntp status

Clock is synchronized, stratum 3, reference is 146.185.130.22

nominal freq is 250.0000 Hz, actual freq is 250.0000 Hz, precision is 2**24

reference time is D76513B4.66A4CDA6 (12:40:20.400 UTC Mon Jul 7 2014)

clock offset is -5.5952 msec, root delay is 13.58 msec

root dispersion is 7966.62 msec, peer dispersion is 7937.50 msec

loopfilter state is 'CTRL' (Normal Controlled Loop), drift is -0.000000018 s/s

system poll interval is 64, last update was 43 sec ago.

```

O relógio foi sincronizado com stratum 3, o que faz sentido, já que o servidor tem um stratum 2 e estamos a um “salto” dele.

OBS: A sincronização NTP pode ser muito lenta, portanto, seja paciente quando os dispositivos ainda não estiverem sincronizados. Uma maneira de acelerar um pouco é ajustar o relógio manualmente para que fique mais próximo da hora atual.

Roteadores Cisco têm dois relógios diferentes, um relógio de software e um relógio de hardware, esses relógios operam separadamente um do outro. O comando para visualizar esses dois relógios são esses abaixo:

```
CoreRouter#show clock  
11:30:25.197 UTC Aug Jul 12 2021
```

```
CoreRouter#show calendar  
11:38:25.197 UTC Aug Jul 12 2021
```

O comando ‘**show clock**’ mostra o relógio do software enquanto o comando ‘**show calendar**’ mostra o relógio do hardware. Observe que o horário está diferente nos dois relógios, ou seja, não estão sincronizados! Isso é algo que precisamos consertar:

```
CoreRouter#(config)ntp update-calendar
```

O comando ‘**ntp update-calendar**’ atualizará o relógio do hardware com a hora do relógio do software. Aqui está o resultado:

```
CoreRouter#show clock  
11:30:25.197 UTC Aug Jul 12 2021
```

```
CoreRouter#show calendar  
11:30:25.197 UTC Aug Jul 12 2021
```

Com isso, fechamos a configuração no nosso roteador. Mas, ainda temos que configurar os dois switches para sincronização dos seus relógios.

Configuração do switch

Os dois switches serão configurados para usar o CoreRouter como o servidor NTP, e também serão configurados para sincronizar seus relógios um com o outro. Para começar, vamos configurá-los para usar o CoreRouter como servidor:

```
SW1(config)#ntp server 192.168.123.3
```

Mais uma vez, pode demorar alguns minutos para sincronização, mas após a espera, esse é o resultado que veremos:

```
SW1#show ntp associations  
address          ref clock      st  when   poll  reach  delay  offset    disp  
*~192.168.123.3  146.185.130.223  3    21     64    1      2.5    1.02   15875.  
* master (synced), # master (unsynced), + selected, - candidate, ~ configured  
SW1#show ntp status  
Clock is synchronized, stratum 4, reference is 192.168.123.3  
nominal freq is 119.2092 Hz, actual freq is 119.2089 Hz, precision is 2**18  
reference time is D765271D.D6021302 (14:03:09.835 UTC Mon Jul 7 2014)
```

```
clock offset is 1.0229 msec, root delay is 14.31 msec
```

```
root dispersion is 16036.00 msec, peer dispersion is 15875.02 msec
```

O relógio do SW1 foi sincronizado, e seu stratum foi definido como 4. Isso faz sentido, pois está um “salto” mais longe que seu servidor NTP (CoreRouter). Vamos fazer o mesmo para SW2:

```
SW2(config)#ntp server 192.168.123.3
```

Depois de alguns minutos, eis o que teremos:

```
SW2#show ntp associations
```

address	ref clock	st	when	poll	reach	delay	offset	disp
*~192.168.123.3	146.185.130.223	3	17	64	37	3.4	1.89	875.8

* master (synced), # master (unsynced), + selected, - candidate, ~ configured

```
SW2#show ntp status
```

Clock is synchronized, stratum 4, reference is 192.168.123.3

nominal freq is 119.2092 Hz, actual freq is 119.2084 Hz, precision is 2**18

reference time is D765274D.D51A0546 (14:03:57.832 UTC Mon Jul 7 2014)

clock offset is 1.8875 msec, root delay is 15.18 msec

root dispersion is 1038.39 msec, peer dispersion is 875.84 msec

SW1 e SW2 agora estão usando CoreRouter para sincronizar seus relógios. Também vamos configurá-los para usar um ao outro para sincronização. Este é o symmetric active mode (modo ativo simétrico) mencionado anteriormente, basicamente os dois switches irão “ajudar” um ao outro a sincronizar, essa configuração pode ser útil em caso de alguma falha com CoreRouter:

```
SW1(config)#ntp peer 192.168.123.2
```

```
SW2(config)#ntp peer 192.168.123.1
```

Depois aguardamos alguns minutos, eis o que veremos no SW1 e SW2 quando ambos estiverem sincronizados um com o outro:

```
SW1#show ntp associations
```

address	ref clock	st	when	poll	reach	delay	offset	disp
*~192.168.123.3	146.185.130.223	3	59	64	37	3.0	-0.74	877.4
++192.168.123.2	192.168.123.3	4	50	128	376	2.2	-2.04	1.3

* master (synced), # master (unsynced), + selected, - candidate, ~ configured

```
SW2#show ntp associations
```

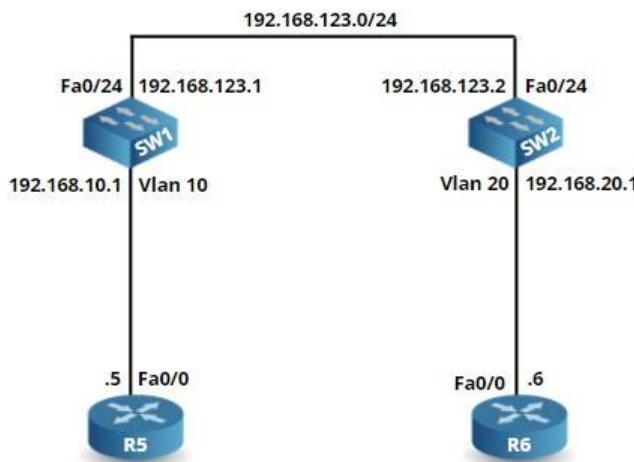
address	ref clock	st	when	poll	reach	delay	offset	disp
*~192.168.123.3	146.185.130.223	3	45	128	377	2.9	1.95	1.0
~192.168.123.1	192.168.123.3	4	67	1024	376	1.8	2.40	1.4

```
* master (synced), # master (unsynced), + selected, - candidate, ~ configured
```

Perfeito, agora tudo está em sincronia. Porém, ainda existem mais algumas coisas que podemos fazer com o NTP. O CoreRouter e os dois switches usam unicast (porta UDP 123) para sincronização, mas também é possível usar multicast ou broadcast. Observe o exemplo abaixo:

NTP - Multicast e Broadcast

Se você tiver mais de 20 dispositivos de rede ou um roteador com recursos de memória e CPU limitados, convém considerar o uso do NTP trabalhando em broadcast ou multicast, pois requer menos recursos. Podemos habilitar multicast ou broadcast no nível da interface. Para demonstrar isso, adicionarei dois roteadores abaixo de SW1 e SW2, que sincronizarão usando multicast ou broadcast, a nossa topologia agora ficará assim:



Vamos configurar o SW1 para usar o endereço multicast 239.1.1.1 e SW2 enviará atualizações de NTP por meio de broadcast:

```
SW1(config)#interface vlan 10
SW1(config-if)#ntp multicast 239.1.1.1
```

```
SW2(config-if)#interface vlan 20
SW2(config-if)#ntp broadcast
```

R5 sincronizará usando multicast:

```
R5(config)#interface fastEthernet 0/0
R5(config-if)#ntp multicast client 239.1.1.1
```

Os comandos são autoexplicativos, vamos ver se funcionou:

```
R5#show ntp associations
address          ref clock      st  when   poll reach  delay  offset  disp
* 192.168.10.1    192.168.123.3  4    14     64      1  1.528  -1.209  0.206
* sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured
```

```
R5#show ntp status
```

```

Clock is synchronized, stratum 5, reference is 192.168.10.1

nominal freq is 250.0000 Hz, actual freq is 250.0174 Hz, precision is 2**24
reference time is D765447B.DA56D83C (16:08:27.852 UTC Mon Jul 7 2014)
clock offset is -0.0012 msec, root delay is 0.01 msec
root dispersion is 0.16 msec, peer dispersion is 0.00 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is -0.000069583 s/s
system poll interval is 64, last update was 35 sec ago.

```

Observe que ele sincronizou com o SW1, conforme indicado no endereço IP na saída do comando.

Vamos configurar a sincronização em broadcast com o R6:

```

R6(config)#interface fastEthernet 0/0
R6(config-if)#ntp broadcast client

```

```

R6#show ntp associations

address          ref clock      st   when   poll reach  delay  offset  disp
* 192.168.20.2    192.168.123.3  4     29     64      1  1.284  -4.035  0.127
* sys.peer, # selected, + candidate, - outlyer, x falseticker, ~ configured

```

```

R6#show ntp status

Clock is synchronized, stratum 5, reference is 192.168.20.2

nominal freq is 250.0000 Hz, actual freq is 250.0132 Hz, precision is 2**24
reference time is D7654496.15979782 (16:08:54.084 UTC Mon Jul 7 2014)
clock offset is -0.0040 msec, root delay is 0.01 msec
root dispersion is 0.59 msec, peer dispersion is 0.00 msec
loopfilter state is 'CTRL' (Normal Controlled Loop), drift is -0.000052939 s/s
system poll interval is 64, last update was 29 sec ago.

```

Há mais um tópico importante para abordar: Segurança! No momento, nossos roteadores aceitarão qualquer fonte como servidor NTP e servirão a qualquer cliente NTP que solicitar atualizações. Para proteger a rede, teremos que configurar autenticação e controle de acesso. Vamos começar com a autenticação.

NTP Autenticação

Quando ativamos a autenticação, todos os pacotes NTP que podem atualizar o relógio tem de ser autenticados. Os pacotes serão autenticados usando HMAC MD5, que contém um número de chave.

Vamos assegurar que SW1 e SW2 se autenticarão no CoreRouter, dessa forma eles aceitarão apenas atualizações de NTP do dispositivo que tenha o endereço IP 192.168.123.3 configurado. Primeiro, vamos configurar o roteador:

```
CoreRouter(config)#ntp authenticate  
CoreRouter(config)#ntp trusted-key 1  
CoreRouter(config)#ntp trusted-key 2  
CoreRouter(config)#ntp authentication-key 1 md5 CISCOBRASIL1  
CoreRouter(config)#ntp authentication-key 2 md5 CISCOBRASIL2
```

Cada switch usará uma chave diferente para autenticação. O comando ‘**ntp authentication-key**’ é necessário para definir o número da chave e a senha. O comando ‘**ntp trusted-key**’ é um pouco estranho, se você não usar, a chave que você configurou não será ativada.

Vamos configurar os switches agora:

```
SW1(config)#ntp authenticate  
SW1(config)#ntp authentication-key 1 md5 CISCOBRASIL1  
SW1(config)#ntp trusted-key 1  
SW1(config)#ntp server 192.168.123.3 key 1
```

```
SW2(config)#ntp authenticate  
SW2(config)#ntp authentication-key 2 md5 CISCOBRASIL2  
SW2(config)#ntp trusted-key 2  
SW2(config)#ntp server 192.168.123.3 key 2
```

A configuração nos switches é semelhante, mas a diferença é que também especificamos a chave para o servidor NTP. SW1 e SW2 usarão apenas 192.168.123.3 para sincronizar seus relógios se a assinatura MD5 estiver correta.

Anteriormente, configuramos SW1 e SW2 para usarem um ao outro como pares e, claro, também podemos usar autenticação entre eles:

```
SW1(config)#ntp authentication-key 12 md5 CISCOBRASIL12  
SW1(config)#ntp trusted-key 12  
SW1(config)#ntp peer 192.168.123.2 key 12
```

```
SW2(config)#ntp authentication-key 12 md5 CISCOBRASIL12  
SW2(config)#ntp trusted-key 12  
SW2(config)#ntp peer 192.168.123.1 key 12
```

A configuração é semelhante, configuramos uma chave e depois afirmamos que ele é confiável.

Com isso fechamos a parte do NTP, agora você é capaz de configurar NTP em qualquer dispositivo de rede da Cisco.

4.3 Explain the role of DHCP and DNS within the network

DHCP e DNS são dois serviços essenciais em redes. Enquanto um servidor DHCP envia informações sobre endereços IPs que os clientes precisam para se conectar à rede, o DNS garante que os servidores, clientes e serviços possam ser encontrados por seus nomes.

Introdução ao DHCP

DHCP (Dynamic Host Configuration Protocol ou ‘protocolo de configuração dinâmica de hosts’) atribui dinamicamente endereço IP, máscara de rede, DNS, default gateway e demais opções de configurações que um dispositivo necessita para ingressar e navegar em uma rede de computadores.

Ele torna mais fácil a adição de novos dispositivos na rede, seja computadores, tablets, celulares, pois o administrador da rede, não precisa configurar endereço IP em todos os dispositivos manualmente, já que o servidor DHCP faz o trabalho

O servidor DHCP distribui endereços IP a partir de um pool de endereço especificado pelo administrador da rede ou pode atribuir endereços estáticos aos clientes, ele identifica o cliente que deverão ter endereços estáticos através do endereço MAC (Media Access Control).

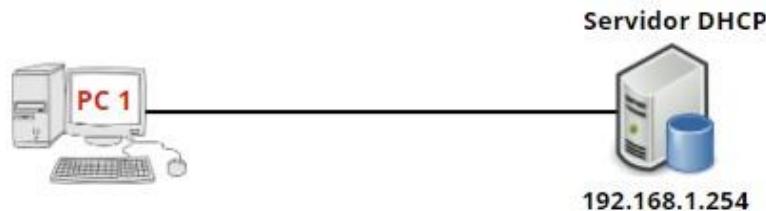
No primeiro cenário, os clientes podem obter IPs diferentes, o que pode ser conveniente se o servidor distribuir endereços de um pequeno pool para um grande número de dispositivos (que não costumam ficar ativos ao mesmo tempo). Se o servidor DHCP distribuir endereços estáticos, todos os clientes sempre receberão o mesmo IP - ideal para serviços de rede como impressora.

O servidor DHCP também determina por quanto tempo um endereço IP é válido. Se o chamado lease (aluguel) expirar enquanto um cliente ainda estiver ativo, ele tentará renovar automaticamente o tempo de lease. Os usuários normalmente não percebem essa troca entre o servidor e o cliente.

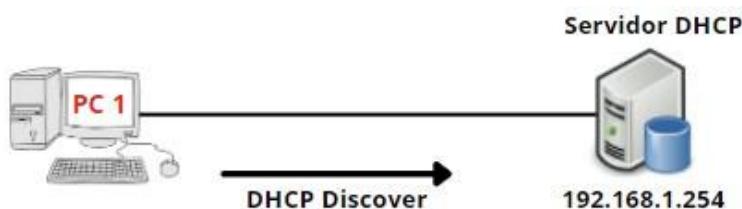
Como mencionado anteriormente, o servidor DHCP também pode transferir outras informações para os clientes, como máscara de sub-rede, servidor de nomes, nome de domínio e gateway – e até mesmo detalhes para inicialização através da rede (inicialização PXE, Preboot eXecution Environment), NTP (Network Time Protocol) ou configuração de proxy via WPAD (Web Proxy Auto-Discovery Protocol).

Funcionamento do DHCP

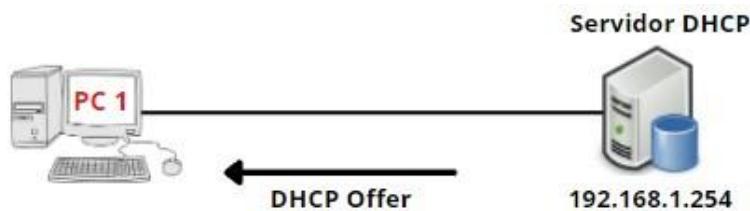
Vamos aprender como funciona o protocolo DHCP. Observe a pequena topologia abaixo:



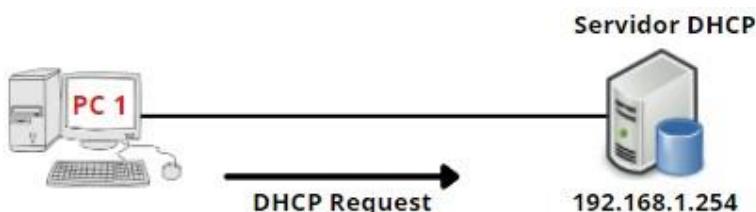
No lado esquerdo temos um computador sem endereço IP. No lado direito, há um servidor DHCP configurado com endereço IP estático 192.168.1.254. Este servidor DHCP fornecerá endereço IP ao computador. Vamos a um passo a passo de como esse processo funciona:



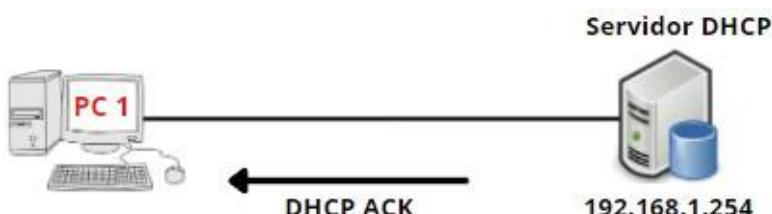
O computador enviará uma mensagem de DHCP Discover (descoberta DHCP). Este pacote é enviado em broadcast porque ele não tem um endereço IP e não sabe se há um servidor DHCP na rede. É claro que em nosso cenário temos um servidor DHCP, então ele responderá a mensagem em broadcast da seguinte forma:



O servidor DHCP responderá com uma mensagem ‘DHCP Offer’ (oferta DHCP), que contém um endereço IP para o computador (o servidor DHCP a essa altura já foi configurado com os endereços IP que ele deve fornecer). Além do endereço IP, pode ser encaminhado também todos os outros recursos que mencionei na introdução. O computador responderá a esta informação:



O computador enviará um pacote chamado de DHCP Request (Solicitação DHCP) em resposta à mensagem de oferta do servidor DHCP, perguntando se não há problema em usar as informações que recebeu. O servidor DHCP responderá da seguinte forma:



O servidor DHCP responderá com uma mensagem DHCP ACK, confirmando ao computador que está tudo bem usar o endereço IP fornecido. Uma abreviação bem comum a todo esse processo é **DORA**: Discover, Offer, Request e Ack. Ainda teremos um tópico ensinando a configuração do DHCP em dispositivos CISCO.

DNS

Graças ao DNS, ninguém precisa se lembrar de endereços IP - o Domain Name System (Sistema de Nomes de Domínio) é um sistema de nomes hierárquico e descentralizado para computadores, serviços e dispositivos que estejam conectados à Internet ou a uma rede privada.

O DNS funciona como uma lista telefônica: ele atribui nomes de domínio como “www.cisco.com” a endereços IP numéricos (178.45.12.152) e vice-versa. O DNS consiste em milhares de servidores trabalhando juntos. Se um servidor não puder resolver um nome ou IP, ele entra em contato com outro servidor, se este não souber ele perguntar ao próximo e assim por diante.

Um servidor DNS em uma rede privada também é responsável pela resolução de nomes. Ele conhece todos os endereços IP e nomes dos dispositivos. Para consultas externas, ou seja, à Internet, o servidor de nomes local pode contatar um ou mais servidores DNS externos.

Tabela de comparação DNS e DHCP

	DHCP	DNS
Definição	Protocolo de configuração Dinâmica de hosts	Sistema de nomes de domínio.
Portas	Executado nas portas 67 e 68	Executado na porta 53
Protocolos suportados	UDP	UDP e TCP
Objetivo	Distribuir endereços IPs automaticamente	Traduzir nomes em endereços IPs
Modelo	Sistema centralizado	Sistema descentralizado
Vantagens	Método confiável de distribuição de IPs, pois dentre outros problemas, evita endereços duplicados na rede.	Facilita a vida do usuário que não precisa lembrar de endereços IPs.

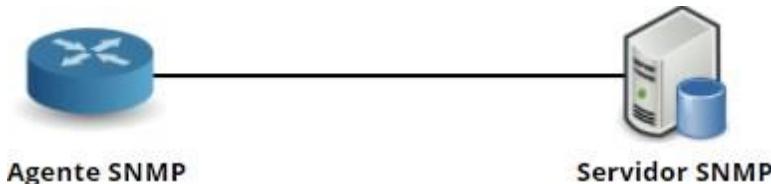
4.4 Explain the function of SNMP in network operations

Imagine que você administre uma rede gigantesca com dezenas de switches, roteadores, uma dúzia de servidores e centenas de estações de trabalho. Em uma rede dessas proporções, com certeza você necessitaria de uma maneira eficiente de monitorar todos esses dispositivos! Isso é possível usando um NMS (Network Management System). Com um NMS, sempre que algo de inesperado acontecer (como uma interface ficar down), você pode receber um e-mail, uma mensagem de alerta, etc, para que possa reagir imediatamente a esse incidente.

Com a popularização das redes na década de 80, convencionou-se que deveria haver algum sistema para monitorar todos os dispositivos de rede baseados em IP. A lógica empregada nesse sistema é que a maioria dos dispositivos como computadores, impressoras, switches e roteadores compartilham algumas características em comum, por exemplo, todos eles têm interfaces, endereço IP, nome de host, buffers e assim por diante.

Com isso em mente, foi criado um banco de dados com variáveis que poderiam ser usadas para monitorar diferentes componentes dos dispositivos de rede, e isso resultou no SNMP (Simple Network Management Protocol).

O SNMP é executado na camada de aplicação e consiste de um “SNMP Manager” (gerenciador SNMP) e um “SNMP Agente” (Agente SNMP). O gerenciador SNMP é o software que está rodando em um computador ou servidor que monitorará os dispositivos de rede, o agente SNMP é executado no próprio dispositivo de rede, ou seja, no cliente.



O banco de dados é chamado de MIB (Management Information Base) e objeto monitorado pode ser o status da interface de um roteador (up ou down), a utilização da CPU em um determinado momento, resumindo, uma infinidade de objetos diferentes. Um objeto no MIB é chamado de OID (Object Identifier).

No exemplo acima, o gerenciador SNMP fará consultas periódicas ao roteador e armazenará essas informações. Desta forma, ele pode criar gráficos para mostrar a utilização da CPU ou utilização da interface em determinado período, ajudando assim a monitorar o desempenho da rede.

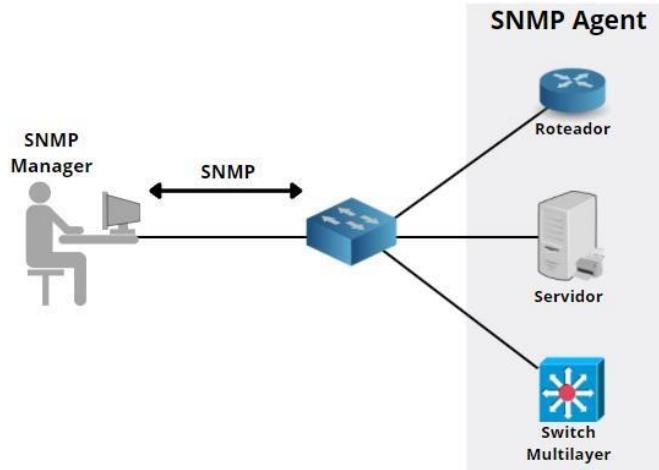
É possível configurar dispositivos de rede através do SNMP. Essa ferramenta é útil especialmente quando precisamos configurar vários switches ou roteadores, através da utilização dessa ferramenta, não precisamos dar telnet/ssh em cada dispositivo separadamente para fazer alterações.

Os pacotes que usados para pesquisar informações são chamados de “SNMP GET Message” e os usados para escrever configurações são chamados de “SNMP SET Message”.

Resumindo, o SNMP é composto por três itens:

- **SNMP Manager** – Um software executado em um servidor para monitorar a rede.
- **SNMP Agent** – Um software que é executado no dispositivo de rede que se deseja monitorar, como roteador, switch.

- **Management Information Base (MIB)** – É a coleção de objetos gerenciados. Esses componentes garantem que a troca de dados entre o gerente e o agente permaneça estruturada. Em outras palavras, o MIB contém um conjunto de perguntas que o SNMP Manager pode fazer ao Agente (e o Agente pode entendê-las). O MIB é compartilhado entre o Agente e o Gerente.



Por exemplo, na topologia acima desejamos monitorar um roteador, um servidor e um switch. Podemos executar o SNMP Agent em todos eles. Em seguida, em um computador (servidor) instalamos o software SNMP Manager para receber informações de monitoramento.

SNMP é o protocolo executado entre o gerente e o agente. A comunicação SNMP entre o gerente e o agente ocorre na forma de mensagens. O processo de monitoramento deve ser feito através de um MIB que é um banco de dados padronizado e contém parâmetros/objetos para descrever esses dispositivos de rede (como endereços IP, interfaces, utilização de CPU, etc). Portanto, o processo de monitoramento agora se torna o processo de GET (obter) e SET (Configurar) as informações do MIB.

Versões SNMP

O SNMP possui várias versões, sendo as três principais:

- SNMP versão 1
- SNMP versão 2c
- SNMP versão 3

O SNMPv1, é a versão original que não é mais utilizada.

SNMPv2c atualizou o protocolo original e ofereceu alguns aprimoramentos. Um dos aprimoramentos mais perceptíveis é a introdução das mensagens INFORM e GETBULK, que serão explicadas posteriormente.

SNMPv1 e v2 não ‘ligam’ muito para questão de segurança, e fornecem segurança com base apenas na ‘community string’. A ‘community string’ é, na verdade, apenas uma senha em texto simples (sem criptografia). Todos os dados enviados em texto não criptografado são vulneráveis à detecção e interceptação de pacotes. Existem dois tipos de ‘community string’ no SNMPv2c:

- **Read-Only (RO):** Permite acesso apenas a leitura dos objetos MIB, esse é mais o seguro dos métodos.
- **Read-Write (RW):** Permite acesso de leitura e gravação nos objetos MIB. Esse método permite que o SNMP Manager altere a configuração de roteadores/switches, etc, portanto, é necessário cautela ao trabalhar com ele.

A ‘community string’ definida no SNMP Manager deve corresponder a uma das ‘community string’ dos agentes, permitindo dessa forma que o ‘Manager’ reconheça e acesse os ‘Agents’.

O SNMPv3 fornece aprimoramentos significativos para lidar com os pontos fracos de segurança existentes nas versões anteriores do SNMP. O conceito de ‘community string’ não existe nessa versão.

SNMPv3 fornece comunicação muito mais segura usando entidades, usuários e grupos. Isso é obtido através da implementação de três novos recursos:

- **Message Integrity (Integridade da mensagem):** Garante que não houve modificação em um pacote durante a fase de envio e recebimento.
- **Authentication (Autenticação):** Usando hashing de senha (com base nos algoritmos HMAC-MD5 ou HMAC-SHA) garante que a mensagem seja de uma fonte válida na rede.
- **Privacy (criptografia):** Usa criptografia (criptografia DES de 56 bits, por exemplo) para criptografar o conteúdo de um pacote.

Nota: Embora o SNMPv3 forneça mais segurança, o SNMPv2c ainda é o mais comum. A Cisco oferece suporte ao SNMPv3 em seus dispositivos desde a versão 12.0.3T do IOS.

As mensagens SNMP são usadas para a comunicação entre o gerenciador SNMP e os agentes. SNMPv1 oferece suporte a cinco mensagens SNMP básicas:

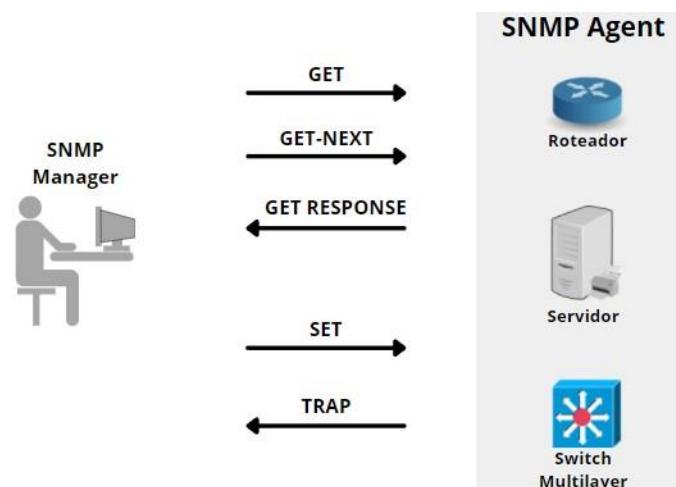
- SNMP GET
- SNMP GET-NEXT
- SNMP GET-RESPONSE
- SNMP SET
- SNMP TRAP

Em geral, as mensagens GET são enviadas pelo Gerenciador SNMP para recuperar informações dos Agentes SNMP, enquanto as mensagens SET são usadas pelo Gerenciador SNMP para modificar ou atribuir o valor aos Agentes SNMP.

É interessante deixar registrado que, GET-NEXT recupera o valor do próximo objeto no MIB, já o GET do objeto em questão.

A mensagem GET-RESPONSE é usada pelos Agentes SNMP para responder às mensagens GET e GET-NEXT.

Ao contrário das mensagens GET ou SET, as mensagens TRAP são iniciadas a partir dos Agentes SNMP para informar o SNMP Manager sobre a ocorrência de um evento. Por exemplo, suponha que você queira ser informado quando o uso da CPU de um servidor ultrapasse 80%. Seria praticamente inviável se o administrador tivesse que usar ativamente a mensagem GET para verificar o uso da CPU periodicamente. Em casos assim, mensagens TRAPs são extremamente adequadas, pois o administrador será informado automaticamente quando o evento ocorrer. A figura abaixo mostra a direção das mensagens SNMP:



No SNMPv2c, duas novas mensagens foram adicionadas: INFORM e GETBULK.

INFORM (informar): Uma desvantagem das mensagens TRAP é que elas não são confiáveis. O SNMP se comunica via UDP, que é um protocolo não confiável. Quando os Agentes SNMP enviam uma mensagem TRAP ao Gerenciador SNMP, ele não consegue “garantir” se a mensagem chegou ao Gerenciador. Para corrigir esse problema, um novo tipo de mensagem denominado INFORM foi introduzido a partir do SNMPv2. O INFORM consegue garantir que a mensagem foi recebida no destino.

GETBULK: A operação GETBULK consegue manipular com eficiência grandes blocos de dados, como várias linhas em uma tabela. O GETBULK preenche uma mensagem de resposta com o máximo de dados solicitados.

OBS: Não há novos tipos de mensagem no SNMPv3 em comparação ao SNMPv2c.

Configuração SNMP

Nessa última parte, veremos a configuração SNMP, assim você terá uma visão mais detalhada de como o protocolo SNMP funciona. O SNMPv2c ainda é mais popular do que o SNMPv3, portanto, configuraremos o SNMPv2c.

1^a passo é configurar a ‘community string’

```
Router(config)#snmp-server community LuizSilverio ro
```

Neste caso, nossa community string foi nomeada de “LuizSilverio”. O **ro** (read-only), significa que utilizaremos o método de leitura.

2^a passo é configurar o endereço IP de um servidor SNMP (SNMP Manager) para receber as TRAPs ou INFORMs

```
Router(config)#snmp-server host 10.10.10.12 version 2c TRAPCisco
```

“TRAPCisco” é a community string para TRAP.

3^a passo é habilitar as SNMP Traps:

```
Router(config)#snmp-server enable traps
```

Se não quisermos habilitar todas as mensagens de trap, podemos especificar de quais traps queremos receber notificações. Por exemplo, se você deseja apenas receber traps sobre interfaces up/down, use este comando:

```
Router(config)#snmp-server enable traps link cisco
```

É claro que temos que configurar um SNMP Manager em um computador\servidor com essas ‘community string’ para que possamos monitorar os dispositivos. Porém, a configuração do SNMP Manager foge do escopo do CCNA e não será abordada nesse livro.

4.5 Describe the use of syslog features including facilities and levels

Mesmo que você nunca tenha ouvido falar em syslog, provavelmente já viu ele atuando quando introduziu alguns comandos em um roteador ou switch. Observe as linhas abaixo:

```
R1#  
*Feb 14 09:38:48.132: %SYS-5-CONFIG_I: Configured from console by console
```

```
R1#  
*Feb 14 09:40:09.325: %LINK-3-UPDOWN: Interface GigabitEthernet0/1, changed state  
to up
```

```
*Feb 14 09:40:10.326: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to up
```

Sempre que algo relevante acontece em um roteador ou switch, o IOS nos informa em tempo real. Quem emite essas informações é o syslog.

Por padrão, as mensagens de syslog são enviadas apenas para o console. Isso ocorre porque o comando do **logging console** vem habilitado por default. Se você efetuar login por meio de telnet ou SSH, não verá nenhuma mensagem syslog. É possível habilitá-lo para telnet ou SSH com o comando **terminal monitor**.

Armazenamento de mensagens syslog

É possível armazenar as mensagens de Syslog tanto localmente quanto remotamente. Primeiro vamos estudar o armazenamento local.

Armazenamento local

As mensagens de syslog no console ou telnet/SSH são úteis se estivermos na frente do monitor naquele momento, mas caso não estejamos, ou se por algum motivo necessitarmos ver mensagens antigas elas não seriam úteis em nada. Porém, felizmente, o IOS mantém um histórico de mensagens syslog. Podemos acessá-las com o comando **show logging**:

```
R1#show logging

Syslog logging: enabled (0 messages dropped, 3 messages rate-limited, 0 flushes,
0 overruns, xml disabled, filtering disabled)

No Active Message Discriminator.

No Inactive Message Discriminator.

Console logging: level debugging, 34 messages logged, xml disabled,
filtering disabled

Monitor logging: level debugging, 0 messages logged, xml disabled,
filtering disabled

Buffer logging: level debugging, 34 messages logged, xml disabled,
filtering disabled

Exception Logging: size (8192 bytes)

Count and timestamp logging messages: disabled

Persistent logging: disabled

No active filter modules.

Trap logging: level informational, 38 message lines logged

Logging Source-Interface:          VRF Name:
```

```

Log Buffer (8192 bytes):

*Feb     8 08:51:58.706: %PA-3-PA_INIT_FAILED: Performance Agent failed to
initialize (Missing Data License)

*Feb 8 08:52:05.064: %LINK-3-UPDOWN: Interface GigabitEthernet0/0, changed state
to up

*Feb 8 08:52:05.068: %LINK-3-UPDOWN: Interface GigabitEthernet0/1, changed state
to up

```

Observe acima algumas mensagens syslog do histórico armazenado localmente no dispositivo. Por padrão, um dispositivo armazena até 8.192 bytes de mensagens syslog em sua RAM, e como sabemos, esse histórico irá se perder quando o dispositivo for reinicializado, afinal está armazenado em uma memória volátil. É possível aumentar o tamanho do buffer de registro:

```
R1(config)#logging buffered 16384
```

O comando acima reserva até 16384 bytes de RAM para mensagens syslog. Essa configuração pode ser vista com o seguinte comando:

```
R1#show logging | include Log Buffer

Log Buffer (16384 bytes):
```

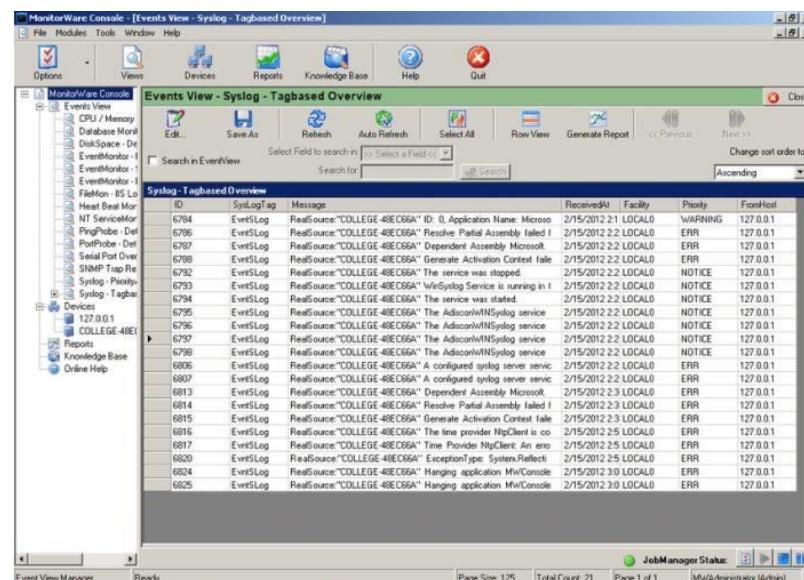
Servidor Syslog

Um histórico local é bom, mas, como fica armazenado na RAM, esse histórico some quando reiniciamos o roteador ou switch. Pense na seguinte situação, o roteador travou e para realizar o troubleshooting precisamos analisar se ele registrou algo antes de cair! Se você tiver dezenas de roteadores e switches, fazer login em cada dispositivo, um por um, para procurar mensagens syslog não constitui uma maneira muito eficiente de procurar algo.

Em redes de produção, usamos um servidor central denominado servidor syslog, e podemos configurar roteadores e switches para encaminhar mensagens syslog para esse servidor da seguinte maneira, sendo 192.168.1.2 o endereço IP do servidor.

```
R1(config)#logging 192.168.1.2
```

Aqui está uma captura de tela de um servidor syslog:



Existem centenas de programas para servidor Syslog, tanto para Linux como Windows, alguns deles gratuitos.

Formato de Mensagem Syslog

Vamos analisar o formato das mensagens do syslog, observe a mensagem abaixo:

```
R1#  
*Aug 14 09:40:10.326: %LINEPROTO-5-UPDOWN: Line protocol on Interface  
GigabitEthernet0/1, changed state to up
```

Acima podemos ver que a interface GigabitEthernet 0/1 está UP, mas há um pouco mais de informação nessa saída. Vamos detalhar como o IOS formata essas mensagens de log:

- **Carimbo de data/hora:** Aug 14 0:40:10.326
- **Facility:**% LINEPROTO
- **Nível de gravidade:** 5
- **Mnemônico:** UPDOWN
- **Descrição:** Line protocol on Interface GigabitEthernet0/1, changed state to up

O carimbo de data/hora (**timestamp**) é bastante autoexplicativo, sem ele seria impossível saber quando um evento ocorreu. É possível desativá-lo ou substituí-lo por uma sequência numérica:

```
R1(config)#no service timestamps
```

```
R1#  
%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to administratively  
down
```

Veja como habilitar a sequência de números:

```
R1(config)#service sequence-numbers
```

```
R1#  
000045: %SYS-5-CONFIG_I: Configured from console by console  
000046: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
```

O syslog é basicamente o processo que gerou a mensagem de syslog. Se você olhar atentamente para algumas das mensagens syslog acima, irá notar que %LINEPROTO informa sobre os protocolos da interface, % SYS informa sobre mensagens gerais do sistema e % LINK informa sobre interfaces que estão UP ou Down.

O nível de gravidade é importante, pois nos diz o quanto importante é a mensagem. Nem tudo o que acontece no roteador ou switch tem importância igual ou exige ação imediata. Voltaremos a essas mensagens em breve.

O mnemônico é um código curto para a mensagem. Por exemplo, “UPDOWN” para interfaces que caem (down) e voltam (UP). “CHANGED” para quando o status da interface muda e permanece naquele estado. Essas informações são úteis quando estamos procurando por tipos de mensagens específicas.

Níveis de gravidade do syslog

Níveis de gravidade ou Severity Levels. Existem diferentes níveis de gravidade. Uma interface que cai é mais importante que uma mensagem informando que saímos da configuração global. No total, existem 8 níveis de gravidade:

Número	Nome	Tradução
0	Emergency	Emergência
1	Alert	Alerta
2	Critical	Critico
3	Error	Erro
4	Warning	Aviso
5	Notice	Notificação
6	Informational	Informativo
7	Debug	Depuração

Quanto menor o número, mais importante é a mensagem do syslog. Emergency e alert são usados quando algo muito grave está acontecendo, por exemplo, quando o roteador fica sem memória e um processo falha. As mensagens critical, error e de warning são usadas para eventos importantes, como interfaces que caem. Eis um exemplo:

```
R1#
*Feb 14 12:02:38: %LINK-5-CHANGED: Interface FastEthernet0/1, changed state to
administratively down
```

Observe o número 5, ele notifica que uma interface foi encerrada administrativamente (administratively down). Abaixo, a mensagem de uma interface que retorna para o status UP:

```
R1#
*Feb 14 12:03:36: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
```

Este é considerado um evento importante com nível de gravidade 3.

Se você estiver realizando um debug no roteador, provavelmente deseja ver as mensagens de depuração no console, mas talvez não queira enviar essas mesmas mensagens para o servidor syslog ou deixá-las registradas no histórico local do roteador. O IOS permite definir quais mensagens syslog desejamos ver, salvar ou enviar para o servidor syslog. Por exemplo:

```
R1(config)#logging console ?
<0-7>          Logging severity level
alerts           Immediate action needed      (severity=1)
critical         Critical conditions        (severity=2)
debugging        Debugging messages        (severity=7)
discriminator   Establish MD-Console association
emergencies     System is unusable       (severity=0)
errors          Error conditions        (severity=3)
filtered         Enable filtered logging
guaranteed       Guarantee console messages
informational    Informational messages    (severity=6)
notifications   Normal but significant conditions (severity=5)
warnings        Warning conditions        (severity=4)
xml             Enable logging in XML
<cr>
```

Com o comando de ‘logging console’, é possível decidir quais níveis de gravidade desejamos ver no console. O padrão é mostrar tudo, até mensagens de depuração (debugging):

```
R1(config)#logging console debugging
```

Podemos fazer o mesmo quando estivermos conectados através do telnet ou SSH:

```
R1(config)#logging monitor debugging
```

Como o armazenamento local do roteador ou switch é limitado, é interessante armazenar apenas ‘avisos’ (Warning) e níveis de gravidade mais altos:

```
R1(config)#logging buffered warnings
```

É possível verificar o logging com o seguinte comando:

```
R1#show logging

Syslog logging: enabled (0 messages dropped, 3 messages rate-limited, 0 flushes,
0 overruns, xml disabled, filtering disabled)

No Active Message Discriminator.

No Inactive Message Discriminator.

Console logging: level debugging, 30 messages logged, xml disabled,
                  filtering disabled

Monitor logging: level debugging, 0 messages logged, xml disabled,
                  filtering disabled

Buffer logging: level warnings, 28 messages logged, xml disabled,
                  filtering disabled

Exception Logging: size (8192 bytes)

Count and timestamp logging messages: disabled

Persistent logging: disabled

No active filter modules.

Trap logging: level informational, 32 message lines logged

Logging Source-Interface:          VRF Name:

Log Buffer (8192 bytes):
```

Já para o servidor syslog, é recomendável enviar o máximo de informações, exceto, mensagens de depuração:

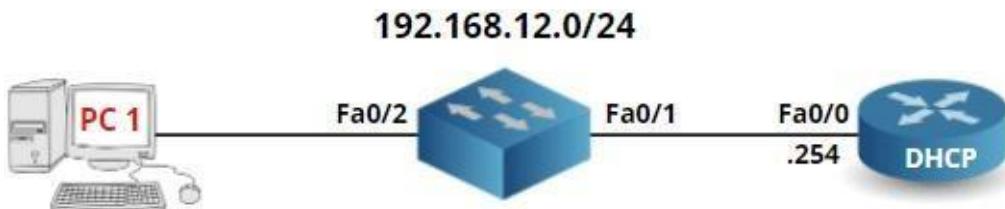
```
R1(config)#logging trap informational
```

4.6 Configure and verify DHCP client and relay

Anteriormente fizemos uma breve introdução ao DHCP explicando como é o seu funcionamento, a troca de pacotes e o popular DORA. Agora é hora de entrarmos na configuração.

Configurando Servidor DHCP em roteadores Cisco

Os roteadores Cisco e switches que operam na camada 3 podem ser configurados como servidor DHCP. É uma configuração bem simples, além disso, vamos aprender a verificar se os dispositivos realmente receberam endereços IPs fornecidos pelo nosso servidor. Usaremos a seguinte topologia:



Acima temos um roteador chamado 'DHCP'. O roteador e o computador estão conectados um ao outro por meio de um switch, e estão na mesma VLAN com endereço de sub-rede 192.168.12.0/24. O primeiro passo, é preparar a interface:

```
DHCP(config)#interface fastEthernet 0/0
DHCP(config-if)#no shutdown
DHCP(config-if)#ip address 192.168.12.254 255.255.255.0
```

Agora vamos configurar o servidor DHCP:

```
DHCP(config)#ip dhcp pool POOL_1
DHCP(dhcp-config)#network 192.168.12.0 255.255.255.0
```

O comando **ip dhcp pool** cria um pool de endereços DHCP, junto ao comando, precisamos especificar o nome do pool, no caso em tela, escolhi o nome ‘pool_1’. Este pool DHCP usará a rede 192.168.12.0/24. Basicamente, isso é tudo que precisamos fazer para colocar o servidor DHCP em funcionamento, não há necessidade de iniciar um serviço ou algo assim.

Podemos verificar se temos clientes DHCP usando o seguinte comando:

```
DHCP#show ip dhcp binding
Bindings from all pools not associated with VRF:
IP address          Client-ID/           Lease expiration      Type
                           Hardware address/
                           User name
192.168.12.2        0063.6973.636f.2d63..    Mar 02 2002 12:24 AM   Automatic
                           6330.372e.3132.3265.
                           2e30.3030.302d.4661.
                           302f.30
```

Observe acima que temos um cliente DHCP, e este cliente recebeu o endereço IP 192.168.12.2. Em redes de produção, também usaremos DHCP para distribuir algumas outras configurações, como gateway padrão, servidor DNS e mais alguns parâmetros. Vamos ver como incluirmos esses parâmetros na configuração do servidor:

```
DHCP(config)#ip dhcp pool POOL_1
DHCP(dhcp-config)#default-router 192.168.12.254
DHCP(dhcp-config)#dns-server 8.8.8.8
```

Acima, configuramos o endereço IP 192.168.12.54 como o gateway padrão para os clientes DHCP com o comando “default-router”. O comando “dns-server” permite especificar um servidor DNS.

Outra opção que temos é exclusão de endereços IP. Com a configuração até agora, nosso servidor DHCP fornecerá o endereço IP .2,3,4,5,6 etc. Veja como fazer isso:

```
DHCP(config)#ip dhcp excluded-address 192.168.12.100
```

O comando para exclusão é **ip dhcp excluded-address**, com ele o endereço IP 192.168.12.100 não será entregue a nenhum cliente DHCP. Este recurso é interessante, pois este pode ser o endereço IP de uma impressora, por exemplo, que precisa ter endereço IP fixo.

Por último, mas não menos importante, se você tiver telefones VoIP ou access point, provavelmente terá que usar alguns campos de “opção” ao fornecer endereços IP aos clientes. Isso também é possível:

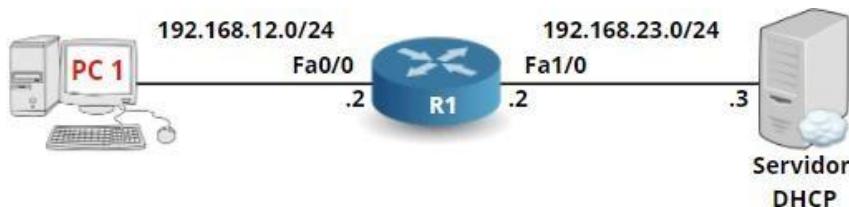
```
DHCP(config)#ip dhcp pool POOL_1  
DHCP(dhcp-config)#option 150 ip 192.168.12.200
```

A opção 150 é usada por telefones IP para localizar o servidor TFTP, possibilitando que eles possam obter seus arquivos de configuração. No comando acima, os telefones irão buscar o servidor TFTP no endereço IP 192.168.12.200.

DHCP Relay Agent

O DHCP é usado para atribuir endereços IP automaticamente a hosts em uma rede. Ele usa 4 pacotes diferentes para fazer isso (lembre-se DORA). Como um host não tem um endereço IP para começar, ele utiliza mensagens de broadcast na esperança de encontrar um servidor DHCP.

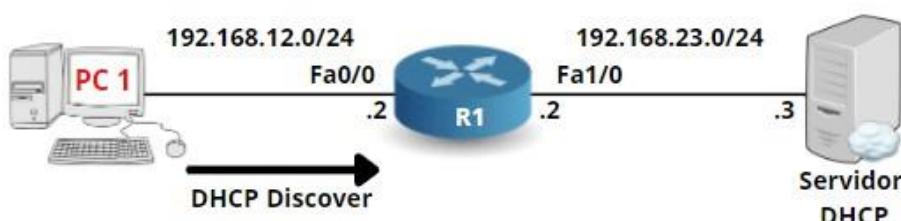
O problema de utilizar pacotes broadcast é que isso significa que o servidor DHCP precisa estar no mesmo domínio de broadcast, pois os roteadores, como explicado nos primeiros tópicos desse livro, não encaminham pacotes broadcast. Observe a topologia abaixo:



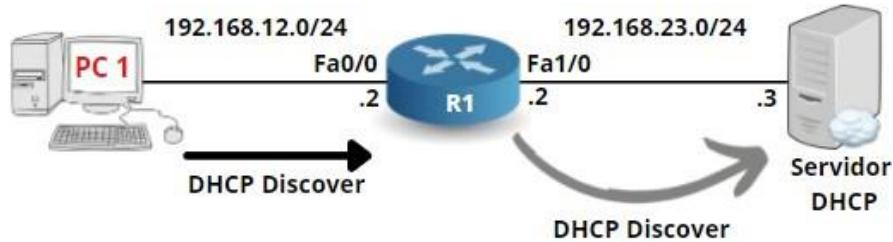
No lado esquerdo temos um cliente (PC1), no meio um roteador (R1) e no lado direito está o nosso servidor DHCP. O cliente deseja obter um endereço IP por meio de DHCP, e enviará uma mensagem DHCP Discovery por broadcast. O roteador, fazendo seu trabalho, não encaminhará o tráfego de broadcast, logo, o pacote DHCP Discovery nunca alcançará o servidor DHCP!

Como você pode imaginar, há uma maneira de resolvemos isso. Existe um recurso chamado “DHCP Relay Agent”. Através desse recurso, o roteador é capaz de encaminhar “DHCP Request” do cliente para o servidor DHCP; quando o servidor DHCP responder, ele encaminhará as mensagens de volta ao cliente.

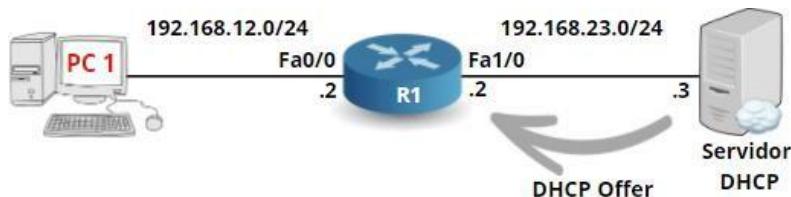
Vamos ilustrar esse processo em detalhes:



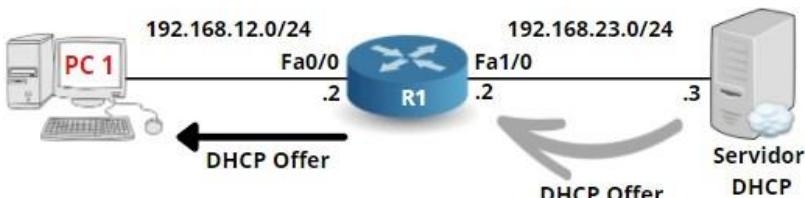
A primeira coisa que acontecerá é que o cliente irá transmitir uma mensagem de descoberta de DHCP (DHCP Discover), o roteador receberá essa mensagem, pois está no mesmo domínio de broadcast do cliente. Eis o que acontece a seguir:



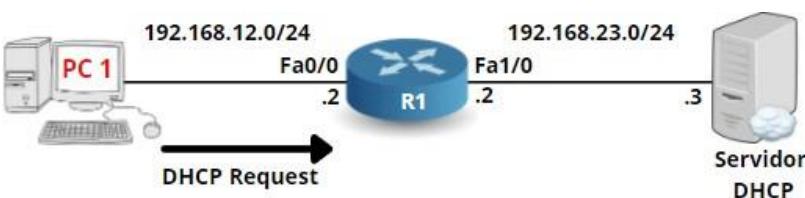
O roteador recebe a mensagem “DHCP Discover” em sua interface FastEthernet 0/0, normalmente ele apenas iria descartar este pacote. Porém, com o recurso “DHCP Relay Agent” habilitado, ele tomará outra ação. Encaminhará o pacote “DHCP Discover” como um pacote unicast, e também inserirá um campo chamado **giaddr** (Gateway IP Address - endereço IP do gateway) no pacote DHCP. Ele irá inserir o endereço IP 192.168.12.2 neste campo, pois recebemos o “DHCP discover” na interface FastEthernet 0/0. Este campo **giaddr** é exigido pelo servidor DHCP para que ele possa identificar de qual pool ele deve selecionar o endereço IP. Além disso, o endereço IP de origem deste pacote **unicast** será 192.168.12.2. Vamos continuar:



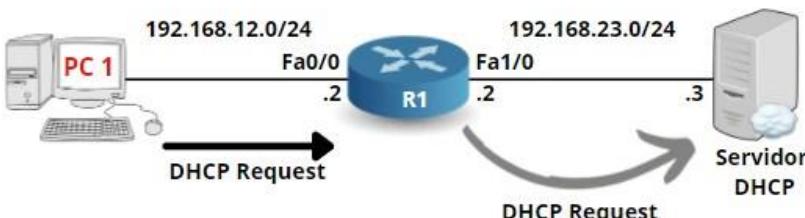
O servidor DHCP recebeu a mensagem “DHCP discover” e, em troca, enviará uma mensagem “DHCP offer”. Isso será enviado como um pacote **unicast** para o roteador:



O roteador, atuando como ‘relay’, encaminhará o pacote com a mensagem “DHCP Offer” na interface FastEthernet0/0 como uma mensagem **broadcast**.



O cliente concorda com o conteúdo do “DHCP Offer” e cria um “DHCP Request” que será enviado em broadcast. O roteador receberá esse pacote em broadcast e fará o seguinte:



Assim como a mensagem inicial “DHCP Discover”, o “DHCP Request” será encaminhado como um pacote unicast. Mais uma vez, o campo **giaddr** será inserido com o endereço IP 192.168.12.2. O servidor DHCP receberá o “DHCP Request” e dará continuidade ao processo:

Por último, o servidor DHCP enviará um “DHCP ACK” em resposta ao “DHCP Request”. Esse pacote é enviado ao roteador usando unicast e o roteador irá enviar essa mensagem por broadcast através da interface FastEthernet 0/0 até o cliente. O cliente agora possui um endereço IP e a missão foi concluída.

Agora que conhecemos toda a parte teórica de funcionamento do “DHCP relay agente”, vamos aprender a configura-lo.

Configuração do DHCP relay agent

Vamos usar a mesma topologia do tópico anterior, com duas mudanças, dessa vez utilizaremos um roteador como servidor DHCP e outro roteador como cliente.



Vamos começar configurando as interfaces:

```
PC1(config)#interface FastEthernet 0/0
PC1 (config-if)#no shutdown
```

```
R1(config)#interface FastEthernet 0/0
R1(config-if)#no shutdown
R1(config-if)#ip address 192.168.12.2 255.255.255.0
R1(config)#interface FastEthernet 0/1
R1(config-if)#no shutdown
R1(config-if)#ip address 192.168.23.2 255.255.255.0
```

```
DHCP(config)#interface FastEthernet 0/0
DHCP(config-if)#no shutdown
DHCP(config-if)#ip address 192.168.23.3 255.255.255.0
```

Nenhuma novidade até aqui. Vamos configurar um pool DHCP para a rede 192.168.12.0/24. Essa rede é onde está o cliente:

```
DHCP(config)#ip dhcp pool REDE_12
DHCP(dhcp-config)#network 192.168.12.0
```

Não configuraremos nenhuma outra opção como gateway ou servidor DNS, pois a única coisa que nos interessa nesse momento é o recurso DHCP agent relay.

O servidor DHCP receberá pacotes DHCP do roteador com endereço IP de origem 192.168.12.2, portanto, precisamos ter certeza de que o servidor DHCP sabe como acessar esta rede. Uma rota estática fará o trabalho:

```
DHCP(config)#ip route 192.168.12.0 255.255.255.0 192.168.23.2
```

Agora, precisamos habilitar a opção DHCP agent relay, e para executar isso, precisamos de pouquíssimos comandos:

```
R1(config)#interface FastEthernet 0/0
R1(config-if)#ip helper-address 192.168.23.3
```

Basta usar o comando **ip helper-address**. Para verificar se ele foi ativado, é só digitar os comandos abaixo:

```
R1#show ip interface FastEthernet 0/0
FastEthernet0/0 is up, line protocol is up
  Internet address is 192.168.12.2/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 1500 bytes
  Helper address is 192.168.23.3
```

Isso é tudo que temos que configurar, vamos descobrir se esta configuração funciona habilitando a depuração no roteador, dessa forma, conseguiremos ver se ele está retransmitindo os pacotes DHCP:

```
R1#debug ip dhcp server packet
```

Agora diremos ao cliente (PC1) para obter endereço IP por meio de DHCP:

```
PC1(config)#interface FastEthernet 0/0
PC1(config-if)#ip address dhcp
```

Eis o que você verá no roteador:

```
R1#
DHCPD: Finding a relay for cliente
0063.6973.636f.2d63.3230.332e.3266.3161.2e30.3030.302d.4661.302f.30 on interface
FastEthernet0/0.

DHCPD: setting giaddr to 192.168.12.2.

DHCPD: BOOTREQUEST from
0063.6973.636f.2d63.3230.332e.3266.3161.2e30.3030.302d.4661.302f.30 forwarded to
192.168.23.3.

DHCPD: forwarding BOOTREPLY to client c203.2f1a.0000.

DHCPD: broadcasting BOOTREPLY to client c203.2f1a.0000.
```

Observe que ele está retransmitindo algo que recebeu do cliente na interface FastEthernet0/0, e o campo giaddr está definido como 192.168.12.2.

O pacote está sendo encaminhado ao servidor DHCP, e o cliente recebeu um endereço IP:

```
PC1#show ip int brief
```

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	192.168.12.1	YES	DHCP	up	up

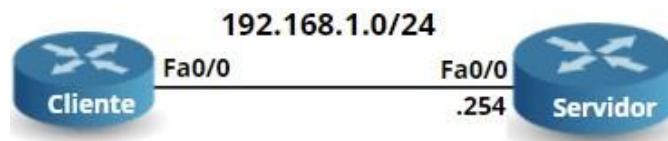
É isso! Nosso roteador recebeu com sucesso um endereço IP através de um servidor DHCP em outra rede.

Cientes DHCP

Roteadores Cisco ou switches layer 3 são frequentemente usados como servidor DHCP, nesses casos, o roteador fornece aos hosts um endereço IP. Porém, também podemos usar os roteadores Cisco como clientes DHCP, o que é útil se o seu ISP fizer uso de endereços IP dinâmicos para clientes.

Configuração

Esta é a topologia que usaremos:



Servidor

Vamos relembrar como configuramos um servidor DHCP em um roteador Cisco, primeiro e único passo é criar um pool para nossa sub-rede local, e nesse cenário também precisaremos incluir uma rota padrão:

```
Servidor(config)#ip dhcp pool Rede_Cliente
Servidor(dhcp-config)#network 192.168.1.0/24
Servidor(dhcp-config)#default-router 192.168.1.254
```

Isso é tudo de que precisamos, vamos dar uma olhada no cliente DHCP agora.

Cliente

Só precisamos de um comando na interface selecionada para habilitar o DHCP:

```
Cliente(config)#interface FastEthernet 0/0
Cliente(config-if)#ip address dhcp
Cliente(config-if)#no shutdown
```

Depois de alguns segundos:

```
Cliente#
%DHCP-6-ADDRESS_ASSIGN: Interface FastEthernet0/0 assigned DHCP address
192.168.1.1, mask 255.255.255.0, hostname Client
```

Observe, já temos um endereço IP:

```
Cliente#show ip interface brief
Interface          IP-Address      OK? Method      Status          Protocol
FastEthernet0/0    192.168.1.1    YES  DHCP        up             up
```

O roteador também instala uma rota padrão:

```

Cliente#show ip route static

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
      + - replicated route, % - next hop override

Gateway of last resort is 192.168.1.254 to network 0.0.0.0

S*    0.0.0.0/0 [254/0] via 192.168.1.254

```

Observe que a distância administrativa é muito grande (254). Isso garante que qualquer outra rota tenha preferência sobre esta.

4.7 Explain the forwarding per-hop behavior (PHB) for QoS such as classification, marking, queuing, congestion, policing, shaping

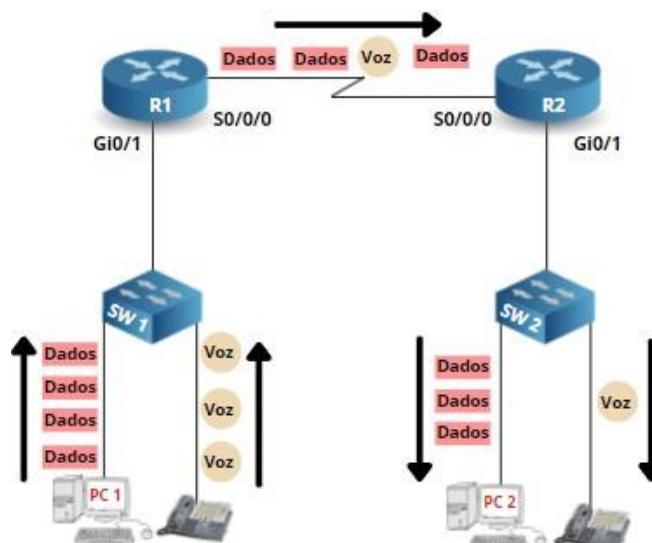
Para o CCNA é cobrado apenas uma pequena introdução sobre o assunto QOS (Quality of service ou qualidade de serviço), porém, a introdução por si só é bem extensa. Mas não se preocupe, vamos abordar todos os temas cobrados na prova.

Introdução

Os dispositivos de rede não se importam com o tipo de tráfego que estão encaminhando. Quando um switch recebe um quadro Ethernet, ele procura o endereço MAC de destino e encaminha o quadro em direção a esse destino. O mesmo se aplica a um roteador, ele recebe um pacote IP, procura o destino na tabela de roteamento e encaminha o pacote.

Não faz diferença para o roteador ou switch que seja uma ligação telefônica ou download de algum arquivo, ele tratará todos os quadros e pacotes da mesma forma.

Essa lógica de encaminhamento é chamada de **best effort** (melhor esforço) ou **FIFO** (First In First Out – Primeiro a entrar, primeiro a sair). Às vezes, isso pode ser um problema. Eis um exemplo rápido:



Na topologia acima vemos uma pequena rede com dois roteadores, dois switches, dois computadores e dois telefones IP. Usamos Gigabit Ethernet em todos os lugares, exceto entre os dois roteadores que estão conectados através de um link serial; este é um link serial lento de 1,54 Mbps.

Quando o computador PC1 e o telefone IP transmitem pacotes de dados e voz, destinados ao computador e ao telefone IP conectados ao R2, é provável que haja congestionamento no link serial. O roteador enfileira (queue) os pacotes que estão esperando para serem transmitidos, mas a fila não é ilimitada, há um tamanho máximo que ela pode atingir. Pense comigo, o que o roteador faz quando a fila atinge o tamanho máximo? Ele descarta os pacotes de dados? Os pacotes de voz? Quando descartamos pacotes de voz, o usuário do outro lado reclama sobre a baixa qualidade da ligação. Quando descartamos pacotes de dados, o usuário pode reclamar que as velocidades de transferência estão baixas.

QoS faz uso de ferramentas para mudar a forma como o roteador ou switch lidam com esses diferentes pacotes. Por exemplo, podemos configurar o roteador para que o tráfego de voz seja priorizado em detrimento do tráfego de dados.

Características do tráfego de rede

Existem quatro características de tráfego de rede com as quais devemos lidar, a saber:

- **Bandwidth** – Largura de banda
- **Delay** - Atraso
- **Jitter** - Variação de atraso dos pacotes
- **Loss** – Perda

Vamos ver cada um desses tópicos:

Bandwidth: A largura de banda é a velocidade do link em bits por segundo (bps). Com a utilização das ferramentas do QoS, podemos dizer ao roteador como usar essa largura de banda.

Com o FIFO (Quando falamos em FIFO pense em uma fila), os pacotes são enviados por ordem de chegada. Uma das coisas que podemos fazer com QoS é criar diferentes filas e colocar certos tipos de tráfego nessas filas. Podemos configurar o roteador para que a fila ‘um’ obtenha 50% da largura de banda, a fila ‘dois’ obtenha 20% da largura de banda e a fila ‘três’, obtenha os 30% restantes.

Delay: Atraso é o tempo que leva para um pacote ir da origem até o destino; isso é chamado de atraso unidirecional (one-way delay). O tempo que leva para ir da origem até destino e voltar é chamado de atraso de ida e volta (round-trip delay). Existem diferentes tipos de atraso; sem entrar em muitos detalhes, vamos para uma visão geral:

- **Processing delay:** Atraso de processamento é o tempo que um dispositivo leva para realizar todas as tarefas necessárias para encaminhar o pacote. Por exemplo, um roteador deve fazer uma pesquisa na tabela de roteamento, verificar tabela ARP e listas de acesso de saída antes de encaminhar um pacote. Dependendo do modelo do roteador, CPU e método de comutação, esses passos podem afetar e causar atraso no processamento.
- **Queuing delay:** Atraso na fila é a quantidade de tempo que um pacote está esperando em uma fila (queue). Quando uma interface está congestionada, o pacote terá que esperar na fila antes de ser transmitido.
- **Serialization delay:** Atraso de serialização é o tempo que leva para enviar todos os bits de um quadro para a interface física realizar a transmissão.
- **Propagation delay:** Atraso de propagação é o tempo que leva para os bits cruzarem um meio físico. Por exemplo, o tempo que os bits levam para viajar através de um link de fibra óptica de 10km é muito menor do que o tempo que leva para os bits viajarem usando links de satélite.

Alguns desses atrasos, como o ‘propagation delay’, são algo que não podemos mudar pois não temos gerência. O que podemos fazer com QoS, no entanto, é influenciar o atraso de enfileiramento (queuing delay). Por exemplo, você pode criar uma fila prioritária que sempre é atendida antes das outras filas. Você pode adicionar pacotes de voz à fila de prioridade para que eles não tenham que esperar muito, reduzindo assim o atraso no envio desses pacotes.

Jitter: É a variação do atraso unilateral em um fluxo de pacotes. Por exemplo, digamos que um telefone IP envie um fluxo contínuo de pacotes de voz. Devido ao congestionamento na rede, alguns pacotes estão atrasados. O atraso entre os pacotes 1 e 2 é de 20 ms, o atraso entre os pacotes 2 e 3 é de 40 ms, o atraso entre os pacotes 3 e 4 é de 5 ms, etc. O receptor desses pacotes de voz deve lidar com o jitter, certificando-se que os pacotes tenham um atraso constante ou teremos uma péssima qualidade na ligação.

Loss: É a quantidade de dados perdidos, geralmente mostrada como uma porcentagem de pacotes enviados que foram perdidos. Se você enviar 100 pacotes e apenas 95 chegarem ao destino, terá 5% de perda de pacotes. A perda de pacotes é sempre possível. Por exemplo, quando há congestionamento, os pacotes serão enfileirados, mas quando a fila estiver cheia os pacotes serão descartados. Com QoS, podemos pelo menos decidir quais pacotes serão descartados quando a fila estiver cheia.

Tipos de tráfego

Com QoS, podemos mudar o comportamento dos dispositivos e consequentemente da rede, fazendo com que determinado tráfego tenha preferência sobre outro tráfego. Podemos fazer isso quando se trata de largura de banda, atraso, jitter e perda. O que você precisa configurar, entretanto, realmente depende dos aplicativos que você usa. Vamos examinar mais de perto os diferentes aplicativos e tipos de tráfego.

Aplicação em lote

Vamos começar com um exemplo simples, um usuário que deseja baixar um arquivo da Internet como um pequeno clipe de 100MB.

Vamos pensar sobre a importância da largura de banda, atraso, jitter e perda quando se trata de baixar um arquivo como este.

O arquivo possui 100MB ou 104857600 bytes. Um pacote IP tem 1500 bytes por padrão, sem o cabeçalho IP e TCP, há 1460 bytes restantes para o segmento TCP. Precisamos de aproximadamente $104857600/1460 = \sim 71.820$ pacotes IP para transferir este arquivo para o computador.

Ter uma boa largura de banda, faz a diferença entre esperar alguns segundos, minutos ou dias para baixar um arquivo como este.

E o delay (atraso)? Há um atraso unilateral para levar os dados do servidor para o computador. Quando você clica no link de download, pode demorar um pouco antes do download começar. Uma vez que os pacotes cheguem, não importa muito qual é o atraso ou a variação do atraso (jitter) entre os pacotes. Você não está interagindo com o download, apenas esperando que ele seja concluído.

E quanto à perda de pacotes? Transferências de arquivos como essas utilizam TCP e, quando alguns pacotes são perdidos, o TCP retransmite seus dados, garantindo que o download chegue completo ao seu computador.

Uma aplicação como navegador Web (Google Chrome, Firefox), que apenas baixa um arquivo, é uma aplicação não interativa, geralmente chamado de aplicação em lote (Batch application) ou transferência em lote. Em casos como esse, é bom ter largura de banda, pois reduz o tempo de espera pela conclusão do download. Atraso, jitter e perda nesse caso, não importam. Com o QoS, podemos atribuir largura de banda suficiente para aplicativos como esse, para garantir que os downloads sejam concluídos no menor tempo possível reduzindo assim a perda de pacotes, evitando retransmissões.

Aplicação Interativa

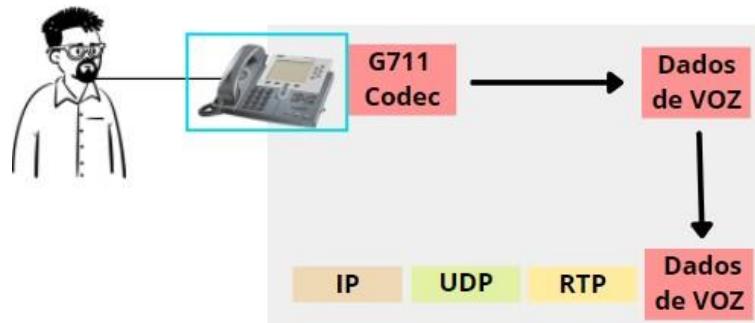
Outro tipo de aplicação é a interativa. Um bom exemplo é quando você usa telnet ou SSH para acessar um roteador ou switch.

Esses aplicativos não exigem muita largura de banda, mas são sensíveis a atrasos e perda de pacotes. Como você está digitando comandos e esperando uma resposta, pode ser chato trabalhar com delay muito grande. Se você já teve que acessar um roteador por meio de um link de satélite, sabe do que estou falando. Os links de satélite podem ter um atraso unilateral de 500-700ms, o que significa que quando você digitar alguns caracteres, haverá uma pequena pausa antes dos caracteres aparecerem no console.

Com QoS, podemos garantir que, em caso de congestionamento, as aplicações interativas sejam atendidas antes das aplicações em lote que consomem muita largura de banda.

Aplicações de voz e vídeo

As aplicações de voz (e vídeo) são as aplicações mais “difíceis” de lidar, pois são muito sensíveis a atrasos, jitter e perda de pacotes. Primeiro, vamos a uma visão geral e rápida de como o VoIP funciona:



Acima temos um usuário se comunicando pelo VoIP. O VoIP, utiliza um codec para processar o som analógico transformando em um sinal digital. O som analógico é digitalizado por um determinado período de tempo, que geralmente é de 20ms. Com o codec G711, cada 20ms de áudio corresponde a 160 bytes de dados.

O telefone criará então um novo pacote IP com cabeçalhos UDP e RTP (Realtime Transport Protocol), adicionando os dados de voz a ele e em seguida encaminhará o pacote IP para o destino. Os cabeçalhos IP, UDP e RTP adicionam 40 bytes de sobrecarga, de modo que o pacote IP terá 200 bytes no total.

Por um segundo de áudio, o telefone criará 50 pacotes IP. $50 \text{ pacotes IP} * 200 \text{ bytes} = 10.000 \text{ bytes por segundo}$. Isso é 80 Kbps. O codec G.729 utiliza menos largura de banda (porém, com qualidade de áudio reduzida), ele utiliza cerca de 24 Kbps.

A largura de banda não é um grande problema para VoIP, o grande desafio é o atraso. Se você está falando com alguém ao telefone, espera que seja em tempo real. Se o atraso for muito grande, a conversa se tornará semelhante a uma conversa de walkie-talkie, em que temos que esperar alguns segundos antes de obter uma resposta. Já o problema com o jitter é devido ao codec precisar de um fluxo constante de pacotes IP, pois ele necessita de um fluxo contínuo e estável para decodificar os dados de voz de volta para o sinal analógico. Os codecs podem contornar um pouco o jitter, porém, existem limitações.

A perda de pacotes é outro grande problema, muitos pacotes perdidos, significa que as conversas terão lacunas. O tráfego de voz em uma rede de dados é possível, mas você precisará de QoS para garantir que haja largura de banda suficiente para manter o atraso, o jitter e a perda de pacotes sob controle. Aqui estão algumas diretrizes para controlar o tráfego de voz:

- Atraso unilateral: <150 ms.
- Jitter: <30 ms.
- Perda: <1%

O tráfego de vídeo (interativo) tem requisitos semelhantes ao tráfego de voz. Porém, o tráfego de vídeo requer mais largura de banda, mas isso realmente depende do codec e do tipo de vídeo que você está transmitindo. Por exemplo, se eu gravar um vídeo do console do meu roteador, 90% da tela permanecerá a mesma. A imagem de fundo permanece a mesma, apenas o texto muda de vez em quando. Um vídeo com muita ação, como um vídeo de esportes, requer mais largura de banda. Assim como o tráfego de voz, o tráfego de vídeo interativo é sensível a atrasos, jitter e perda de pacotes. Aqui estão algumas diretrizes:

- Atraso unilateral: 200 - 400 ms.
- Jitter: 30 - 50 ms.
- Perda: 0,1% - 1%

Ferramentas QoS

Falamos um pouco sobre por que precisamos de QoS e dos diferentes tipos de aplicações com requisitos distintos. Agora vamos falar sobre as ferramentas reais que podemos usar para implementar QoS:

- **Classification and marking:** A Classificação e marcação é utilizada para que possamos dar um tratamento diferente a determinados pacotes, para isso, temos que identificá-los e marcá-los.
- **Queuing – Congestion Management:** Enfileiramento - Gerenciamento de congestionamento é utilizado porque em vez de ter uma grande fila onde os pacotes são tratados com FIFO (Fila), podemos criar várias filas com prioridades diferentes.
- **Shaping and Policing:** Modelagem e Policiamento, essas duas ferramentas são usadas para limitar a taxa de tráfego.
- **Congestion Avoidance:** Prevenção de congestionamento, existem algumas ferramentas que podemos usar para gerenciar a perda de pacotes e reduzir o congestionamento.

Vamos examinar todas essas ferramentas.

Classification and marking

Antes de dar tratamento diferenciado aos pacotes, temos que identificá-los. Isso é chamado de classificação.

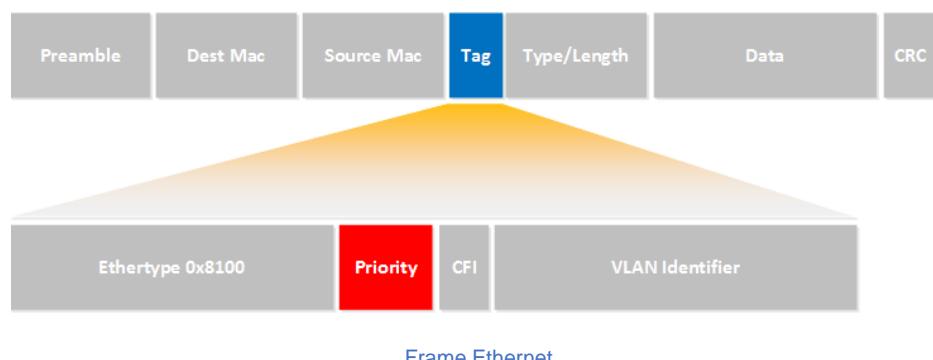
A classificação pode ser realizada de várias maneiras. Uma maneira comum é usar uma lista de acesso para dar ‘match’ com certas características do pacote IP, como endereço de origem e destino, número de porta. Por exemplo, uma lista de acesso que corresponda à porta de destino TCP 80 é uma maneira rápida de classificar todo o tráfego HTTP.

Depois da classificação do tráfego, é uma prática recomendada marcar o pacote (mark the packet).

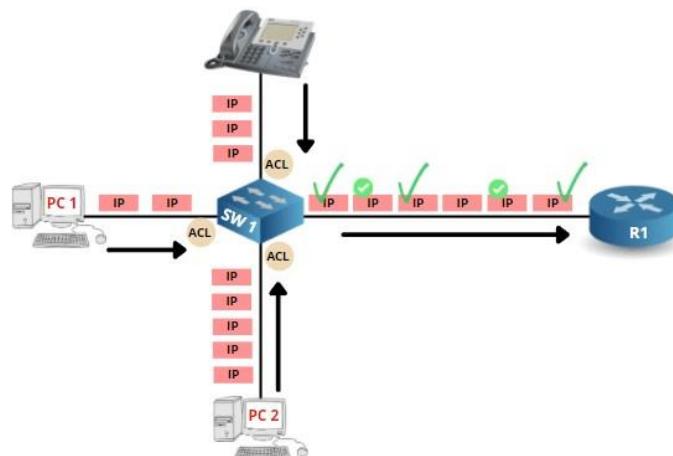
Marcar um pacote significa alterar um ou mais campos do cabeçalho em um pacote IP ou em um quadro Ethernet. Por exemplo, um pacote IP tem o campo ToS (Tipo de serviço) que podemos usar para marcar o pacote:



Os frames Ethernet não têm esse campo, mas possuem um campo semelhante utilizado em interfaces trunk. A tag adicionada pelo protocolo 802.1Q tem um campo de prioridade:



Abaixo, uma ilustração para ajudar a visualizarmos a classificação e marcação:



Na imagem temos um switch com dois hosts e um telefone. O switch recebe vários pacotes IP dos hosts e do telefone. O switch está configurado para classificar esses pacotes usando uma lista de acesso, então, ele marcará os pacotes IP usando o campo ToS do cabeçalho IP.

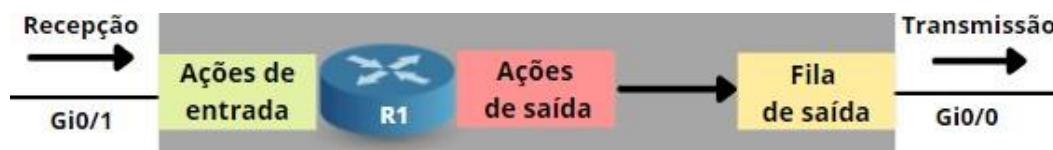
A maioria dos telefones IP marca os pacotes IP que eles criam.

O motivo pelo qual usamos a marcação é que às vezes a classificação requer algumas listas/regras de acesso complexas que podem degradar o desempenho do roteador ou switch que está fazendo a classificação. No exemplo acima, o roteador recebe os pacotes marcados para que não precise fazer classificações complexas usando listas de acesso como o switch. Ele ainda fará a classificação, mas só precisa procurar os pacotes marcados.

Gestão de congestionamento

Todo dispositivo de rede utiliza enfileiramento (queuing). Por exemplo, quando um roteador recebe um pacote IP ele verifica a sua tabela de roteamento, decide por qual interface enviará o pacote, e em seguida encaminha esse pacote. Caso a interface esteja ocupada, o pacote será colocado em uma fila até que a interface esteja liberada.

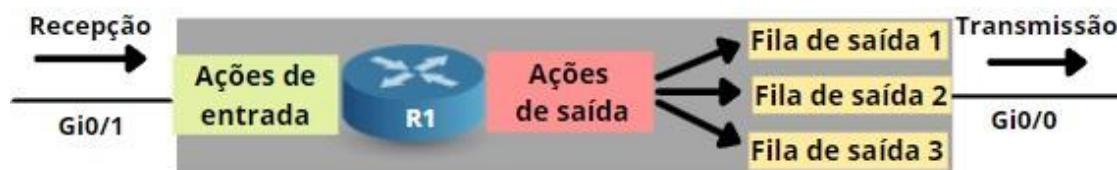
Apesar de estarmos falando de roteadores e pacotes, o mesmo processo se aplica a switches e outros dispositivos de rede. Eis uma ilustração desse processo:



Acima, a representação de um roteador recebendo um pacote, ele pode executar uma ou mais ações de entrada, como por exemplo, uma lista de acesso para filtragem desses pacotes. Depois que o roteador decide para onde encaminhar o pacote, ele pode realizar uma ou mais ações de saída, como por exemplo, NAT. Esse pacote é colocado em uma fila de saída, e fica aguardando nessa fila até que interface esteja pronta (livre), para só então ser transmitido.

Na imagem acima, temos apenas uma fila de saída, e por isso, todos os pacotes serão tratados de acordo com a ordem de chegada. Aqui estamos lidando com um escalonador FIFO (First In First Out – Primeiro a Entrar, Primeiro a Sair), onde há apenas uma fila e todos devem esperar nessa fila.

A maioria dos dispositivos de rede oferece várias filas de saída, nesses casos, a imagem ficaria mais ou menos assim:

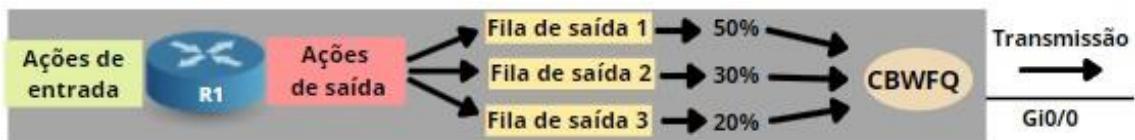


Round Robin

As filas são atendidas de acordo com o Agendador (Scheduler) utilizado. O agendamento ‘round robin’ é um algoritmo de agendamento que percorre as filas seguindo uma ordem, revezando cada uma dessas filas. Por exemplo, pode ser encaminhado um pacote de cada fila, começando com a fila 1, depois fila 2, depois fila 3, retornando para fila 1, etc.

Weighted round Robin (round robin ponderado) dá mais preferência a certas filas. Por exemplo, ele pode encaminhar quatro pacotes da fila 1, depois dois pacotes da fila 2, um pacote da fila 3, retornando para fila 1, e assim sucessivamente.

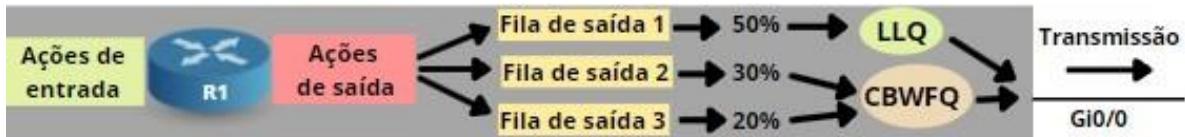
Os roteadores Cisco utilizam o escalonador conhecido como CBWFQ (Class Based Weighted Fair Queuing), que garante uma largura de banda mínima para cada classe quando há congestionamento. O CBWFQ usa o planejamento “round Robin” ponderado e permite configurar a ‘ponderação’ como uma porcentagem da largura de banda da interface. Aqui está uma ilustração para ajudá-lo a visualizar esse processo:



Fila de baixa latência - Low Latency Queuing

O agendamento round robin funciona muito bem para aplicações de dados, pois garante uma certa largura de banda para cada fila. No entanto, não é a forma ideal para lidar com tráfego sensível a atrasos, como VoIP. Por exemplo, quando o Agendador estiver esvaziando a fila 2 e 3, os pacotes na fila 1 ainda estarão esperando para serem encaminhados, provocando atraso.

Em casos em que o tráfego é sensível a atrasos e jitter, esse tráfego deve ser adicionado a uma fila prioritária (priority queue):



Observe acima que a primeira fila agora está conectada ao LLQ. Sempre que um pacote é adicionado à fila 1, ele será encaminhado imediatamente, e todas as outras filas terão que esperar.

É importante definir um limite para a fila de prioridade, caso contrário, é possível que o Planejador (scheduler) esteja tão ocupado encaminhando pacotes da fila de prioridade que as outras filas nunca serão atendidas. Quando essas filas ficam cheias, os pacotes são descartados. Isso é chamado de “inanição de fila” (queue starvation).

Definir um limite para a fila de prioridade introduz outro problema. E se tivermos tanto tráfego de voz a ponto de começar a acontecer descartes na fila de prioridade? Isso afetaria todas as chamadas de voz. Normalmente, isso é resolvido com o CAC (Controle de admissão de chamadas). Resumindo, o CAC é um sistema que pode ser configurado em um PBX, o que garante que só haverá até uma quantidade X de chamadas de voz simultâneas. Se houver capacidade para 15 chamadas de voz simultâneas e ocorrer a tentativa de realização da 16ª ligação, essa ligação receberá o sinal de ocupado ou a chamada de voz será redirecionada por meio da PSTN (Linha telefônica convencional).

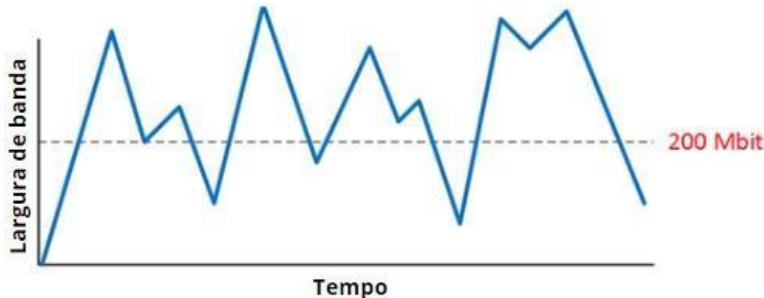
Policing and Shaping

Policing e Shaping são duas ferramentas de QoS usadas para limitar a taxa de bits. As polices descartam o tráfego enquanto os shapers retêm os pacotes em uma fila. Observe o exemplo abaixo:

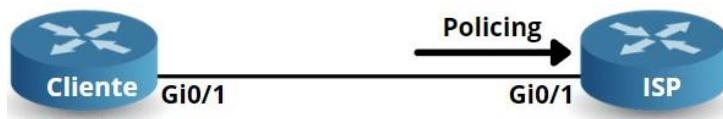


Acima, temos um roteador cliente e um roteador ISP conectados através de interfaces Gigabit Ethernet. Essas interfaces funcionam a 1000Mbit. Vamos montar um cenário em que o cliente solicite uma conexão de 200Mbit. Como a taxa de transmissão das interfaces é de 1000Mbit, o ISP terá que de alguma forma limitar o tráfego a 200Mbit, descartando o tráfego que ultrapassar o limite contratado. Uma maneira de limitar o tráfego é através do uso do Policing,

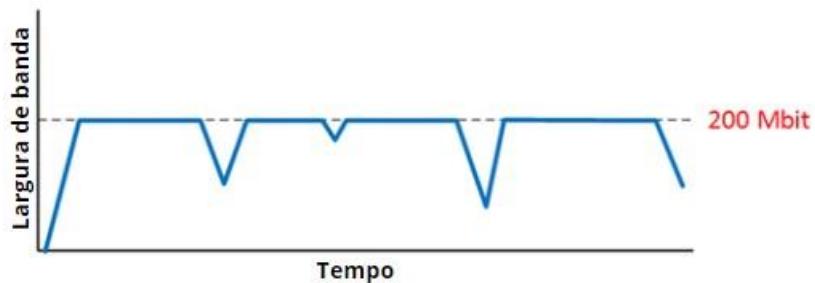
Sem policing, a taxa de bits ficaria conforme o gráfico abaixo:



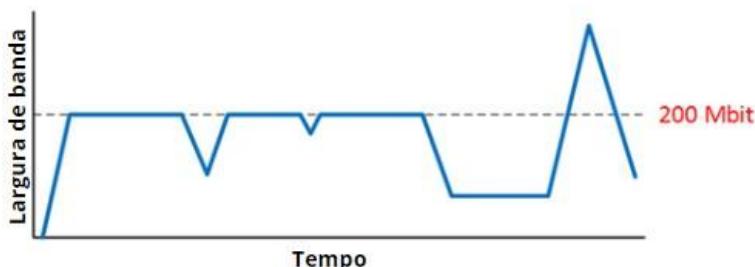
A linha tracejada é a taxa de bits pela qual o cliente pagou. Isso normalmente é chamado de CIR (Taxa de Informações Comprometidas). Sem policing, o cliente obteria uma taxa de bits maior do que aquela pela qual pagou. Em casos assim, o ISP configura o policing na interface de entrada:



Com o policing ativado, a taxa de bits fica assim:



200Mbits agora é um limite rígido que pode não ser totalmente justo para o cliente. Durante um longo período de tempo, é impossível obter uma taxa média de 200 Mbits. Por causa disso, o policing é frequentemente implementado para que possa "estourar" o limite de tráfego por um curto período após momentos de inatividade:

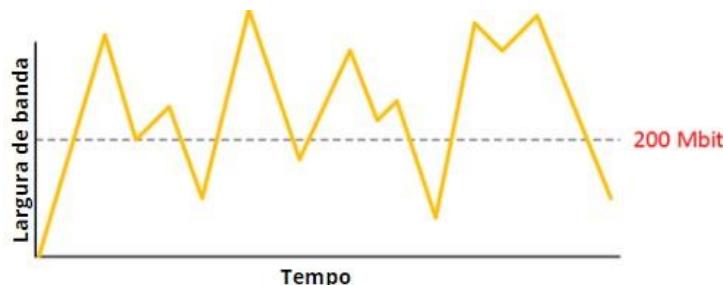


Observe que após um período de tempo com a taxa de transmissão abaixo do contratado, é possível o cliente exceder a taxa CIR de 200 Mbit por um breve período antes que a policing seja acionada.

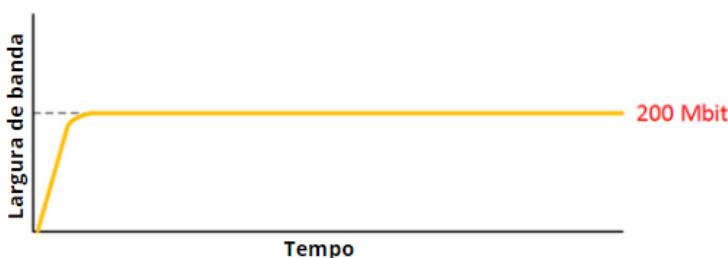
Shaping

No exemplo anterior, você viu como um ISP pode usar policing para interromper o tráfego. Se o cliente tiver uma taxa CIR de 200 Mbit e ele exceder essa taxa, o tráfego excedente será bloqueado.

Para evitar que isso aconteça, podemos implementar shaping no lado do cliente. O shaper irá enfileirar as mensagens, atrasando-as para uma determinada taxa CIR. Sem shaping, a taxa de bits que o cliente envia pode ter a seguinte aparência:



Tudo que estiver acima da linha tracejada será descartado pela policing do ISP. Depois de configurar o shaper, a taxa de bits ficará assim:



O Shaper enfileirará o tráfego excedente, os pacotes ficarão nessa fila até que possam ser transmitidos dentro da taxa de transmissão contratada. Isso evita que o tráfego seja interrompido no ISP.

O shaper resolve um problema, mas introduz uma série de outros problemas, pois a maneira como ele atua pode provocar delay e jitter. Há formas de contornar esses problemas, mas fogem do escopo do CCNA.

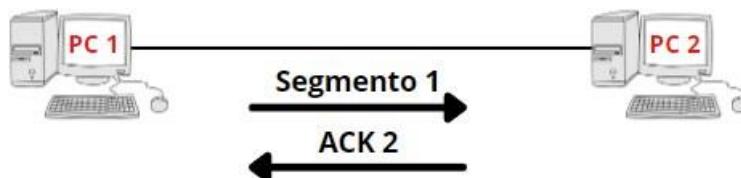
Congestion Avoidance

Para entender a prevenção de congestionamento, temos que falar sobre TCP e tamanho da janela.

O TCP utiliza controle de fluxo baseado em tamanho de janela. Nesse sistema, o receptor informa ao remetente quantos bytes ele pode enviar antes de esperar uma confirmação (acknowledgment) que esses dados foram recebidos. Quanto maior o tamanho da janela, menor será o cabeçalho e maior será a taxa de transferência.

Veja como funciona o mecanismo de inicialização lenta do TCP (TCP slow-start):

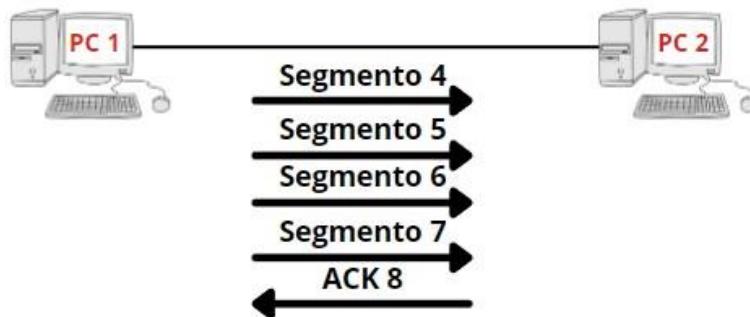
O TCP pode usar “janela de congestionamento” (Congestion Window - CWND) e “janela do receptor” (receiver window - RWN) para controlar a taxa de transferência e evitar o congestionamento da rede. Quando não há perda de pacotes, o tamanho da janela aumentará, dobrando de tamanho. Abaixo você pode ver que o PC2 recebe um único segmento TCP, e logo em seguida vem a confirmação (ACK).



PC1 aumentará o tamanho da janela e agora enviará dois segmentos TCP antes do PC2 enviar o ACK:



O tamanho da janela dobrará novamente, dessa vez PC1 enviará quatro segmentos TCP antes que o ACK seja enviado:



Continuamos dobrando o tamanho da janela até que os segmentos TCP se percam, ou quando atingirem o tamanho da janela do receptor (RWND). Para cada segmento TCP perdido, o tamanho da janela é reduzido pela metade.

Agora, vamos examinar mais detalhadamente o enfileiramento e como o tamanho da janela TCP se aplica ao enfileiramento. Eis o exemplo de uma fila de saída:



A fila de saída acima possui cinco pacotes, ainda há espaço para mais pacotes. Caso a interface de saída esteja ocupada, mais pacotes serão enfileirados:

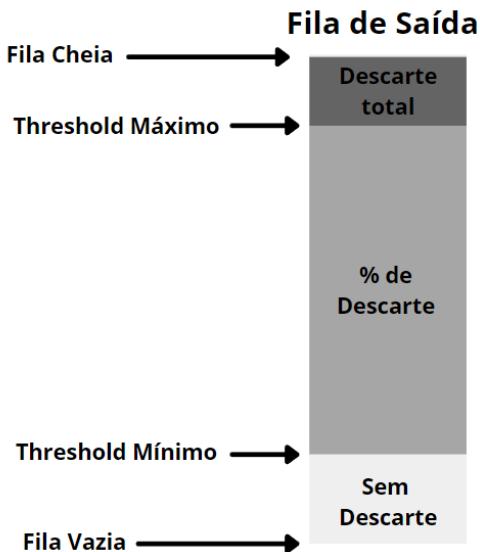


Observe que nesse momento, a fila está cheia, portanto, se outro pacote chegar, ele será descartado:



Esse processo é chamado de “Queda da cauda” (tail drop). Para lidar com esse processo, podemos usar ferramentas de ‘prevenção de congestionamento’ (congestion avoidance) como o WRED. Essas ferramentas irão monitorar a fila de saída e, uma vez que ela atinja determinado nível, irá descartar segmentos TCP, esperando que, ao reduzir o tamanho da janela, as conexões TCP diminuam, reduzindo assim o congestionamento e evitando a “queda da cauda”.

Abaixo, uma ilustração desse processo:



Quando a fila está vazia, não descartamos nenhum pacote. Uma vez que a fila está entre os limites mínimo e máximo, descartamos uma pequena porcentagem de pacotes. Quando o limite máximo é ultrapassado, todos os pacotes são descartados.

A ferramenta de prevenção de congestionamento pode descartar pacotes aleatoriamente ou pode ser configurada para dar a determinados pacotes um tratamento diferente com base em sua marcação.

4.8 Configure network devices for remote access using SSH

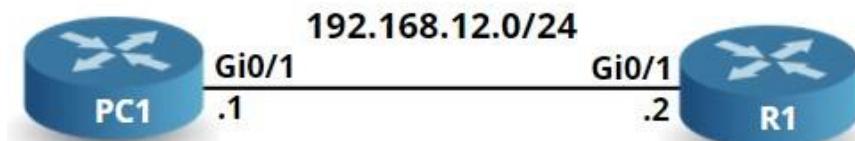
SSH (Secure Shell) é um método seguro para acesso remoto, pois inclui autenticação e criptografia. Para conseguir esse nível de segurança, ele utiliza um par de chaves RSA: Sendo uma chave pública e outra privada.

Existem duas versões, a versão 1 e a 2. A versão 2 é mais segura, e por isso a mais usada.

Por último, mas não menos importante, para configurar o SSH, é necessário uma imagem IOS que ofereça suporte a recursos de criptografia. Caso contrário, somente será possível acesso remoto ao dispositivo por meio do Telnet.

Configuração

Para demonstrar a configuração do SSH, usaremos a seguinte topologia:



Vamos configurar SSH no R1 para que possamos acessá-lo de qualquer outro dispositivo. R2 será usado como um cliente SSH.

Servidor SSH

O nome do par de chaves RSA será o nome do host e o nome de domínio do roteador. Primeiro, vamos configurar um nome de host:

```
Router(config)#hostname R1
```

E agora, um nome de domínio:

```
R1(config)#ip domain-name CISCOBRASIL.LOCAL
```

Agora podemos gerar o par de chaves RSA:

```
R1(config)#crypto key generate rsa
The name for the keys will be: R1.CISCOBRASIL.LOCAL
Choose the size of the key modulus in the range of 360 to 4096 for your
General Purpose Keys. Choosing a key modulus greater than 512 may take
a few minutes.

How many bits in the modulus [512]: 2048
% Generating 2048 bit RSA keys, keys will be non-exportable...
[OK] (elapsed time was 3 seconds)
```

Quando usamos o comando **crypto key generate rsa**, ele pergunta quantos bits desejamos usar para o tamanho da chave. Por questões de segurança, no momento, o tamanho de chave de 2048 bits é aceitável. Tamanhos de chave de 1024 ou menores devem ser evitados, e tamanhos maiores demoram muito para serem calculados.

Agora que o par de chaves foi gerado, a seguinte mensagem aparecerá:

```
R1#
%SSH-5-ENABLED: SSH 1.99 has been enabled
```

Como você pode ver acima, o SSH está correndo na versão 1, que é a versão padrão. Vamos alterá-la para a versão 2:

```
R1(config)#ip ssh version 2
```

O SSH está habilitado, mas também temos que configurar as linhas VTY:

```
R1(config)#line vty 0 4
R1(config-line)#transport input ssh
R1(config-line)#login local
```

Os comandos acima garantem que para acesso remoto ao roteador, apenas o SSH será usado, garantem também que iremos utilizar usuários do banco de dados local. Mas, para isso, precisamos criar um usuário:

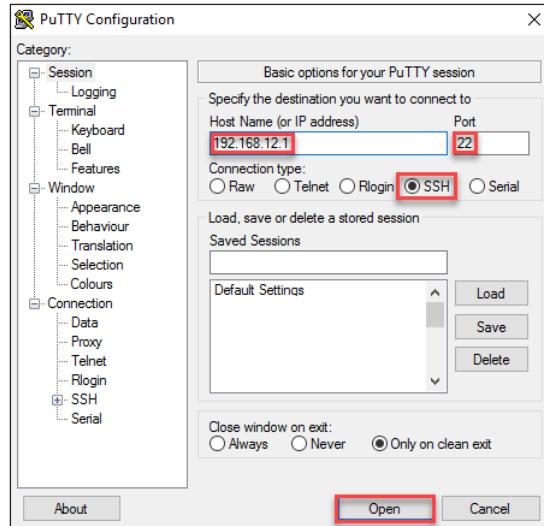
```
R1(config)#username Luiz password Cisco123
```

Tudo resolvido. Vamos verificar se somos capazes de nos conectar ao roteador R1 por meio do SSH.

Cliente SSH

Vamos conectar no R1 utilizando um computador com Windows 10 instalado.

Para essa conexão, precisamos de um cliente SSH instalado no Windows, o mais utilizado é provavelmente o Putty. A única coisa que precisamos fazer após abrir o Putty, é selecionar o protocolo SSH, inserir o endereço IP do dispositivo e deixar a porta padrão em 22:



Essa será a mensagem que aparecerá no console do putty:

```
login as: Luiz
Using keyboard-interactive authentication.

Password:

R1>
```

É possível logar diretamente como cliente SSH a partir de outro dispositivo Cisco, vamos utilizar o R2 como cliente SSH:

```
R2#ssh ?
  -c      Select encryption algorithm
  -l      Log in using this user name
  -m      Select HMAC algorithm
  -o      Specify options
  -p      Connect to this port
  -v      Specify SSH Protocol Version
  -vrf   Specify vrf name
WORD  IP address or hostname of a remote system
```

Existem algumas opções, mas, no mínimo, devemos especificar o nome de usuário e endereço IP:

```
R2#ssh -l Luiz 192.168.12.1
Password:

R1>
```

Assunto encerrado, estamos conectados ao R1 por meio do SSH.

4.9 Describe the capabilities and function of TFTP/FTP in the network

Neste último tópico desse bloco, veremos as funcionalidades e diferenças entre os protocolos FTP e TFTP.

FTP

FTP significa “File Transfer Protocol” (Protocolo de Transferência de Arquivos). Ele é usado para enviar/receber arquivos para um dispositivo remoto. O FTP é definido na RFC 959.

O FTP estabelece duas conexões entre o cliente e servidor, uma para informações de controle e outra para transferência de dados. As informações de controle carregam comandos/respostas, e a de dados, obviamente, carrega dados. A autenticação precisa ser feita inicialmente por meio da validação de nome de usuário e senha. Uma vez feito isso, os arquivos podem ser transferidos entre os dois dispositivos. O FTP lida tanto com arquivos binários como com arquivos formato de texto.

Quando um cliente solicita conexão com o servidor FTP, uma conexão TCP é estabelecida através da porta 21 do servidor. Após a autenticação, outra conexão TCP é estabelecida para a transferência dos dados na porta 20.

TFTP

TFTP significa “Trivial File Transfer Protocol” (Protocolo trivial de transferência de arquivos) é definido na RFC 783. Ele é mais simples que o FTP, pois faz a transferência de arquivos entre cliente e servidor sem solicitar autenticação do usuário e outros recursos úteis suportados pelo FTP. Uma clara diferença é que, o TFTP usa UDP enquanto FTP utiliza TCP.

Devido a utilização do UDP, o TFTP se torna um protocolo não confiável, por isso, ele usa a camada de aplicação para recuperação de dados. Isso é feito incorporando um pequeno cabeçalho entre o cabeçalho UDP e os dados. Este cabeçalho incorpora códigos de leitura (read), escrita (write) e confirmação (acknowledgement) junto com um esquema de numeração. Esse esquema de numeração é utilizado para acusar o recebimento e reenviar os dados em caso de falhas na soma de verificação (checksum). Em suma, o TFTP envia um bloco e espera a confirmação antes de enviar outro bloco.

As principais diferenças entre FTP e TFTP são:

- O FTP usa as portas de números 20 e 21 do TCP, o TFTP usa a porta 69 do UDP.
- O FTP pode ser usado interativamente. O TFTP permite apenas a transferência unidirecional de arquivos.
- O FTP trabalha com TCP, é orientado à conexão e fornece controle confiável. O TFTP trabalha com UDP, por isso, possui cabeçalho menor e praticamente não fornece nenhum controle.
- O FTP fornece autenticação do usuário. TFTP não.

Exercícios

1. Qual comando permite que um roteador se torne um cliente DHCP?
 - a) ip address dhcp
 - b) ip helper-address
 - c) ip dhcp pool
 - d) ip dhcp cliente
2. Um engenheiro de rede precisa realizar o backup da configuração de 20 roteador. Qual protocolo permite que o engenheiro execute esta função usando o Cisco IOS MIB?
 - a) CDP
 - b) SNMP
 - c) SMTP
 - d) ARP
3. Quais são as duas tarefas que devem ser executadas para configurar o NTP no modo cliente de forma confiável em um servidor na rede? (Escolha duas opções)
 - a) Habilitar autenticação NTP
 - b) Verificar o fuso horário
 - c) Desativar as transmissões NTP
 - d) Especificar o endereço IP do servidor NTP
 - e) Definir a chave privada do servidor NTP
4. Se uma mensagem de nível de aviso for enviada a um servidor syslog, qual evento ocorreu?
 - a) Um dispositivo de rede foi reiniciado
 - b) Ocorreu uma falha na verificação do ARP
 - c) Uma rota oscilou
 - d) Uma operação de 'debug' está em operação
5. Em um ambiente CDP, o que acontece quando a interface CDP e um dispositivo adjacente é configurada sem um endereço IP?
 - a) O CDP torna-se inoperante naquele vizinho
 - b) O CDP utiliza endereço IP de outra interface para aquele vizinho
 - c) O CDP opera normalmente, mas não pode fornecer informações de endereço IP daquele vizinho
 - d) O CDP opera normalmente, mas não pode fornecer nenhuma informação para aquele vizinho
6. Quais são as duas afirmações sobre NTP são verdadeiras?
 - a) NTP usa UDP sobre IP.
 - b) Os roteadores Cisco podem atuar tanto como clientes e servidores NTP.
 - c) Os roteadores Cisco podem atuar apenas como servidores NTP.
 - d) Os roteadores Cisco podem atuar apenas como clientes NTP.
 - e) NTP usa TCP sobre IP.
7. Qual comando deve ser utilizado para configurar um delay de 5 segundos no LLDP?
 - a) lldp timer 5000
 - b) lldp holdtime 5
 - c) lldp reinit 5000
 - d) lldp reinit 5
8. Qual afirmação sobre o Cisco Discovery Protocol é verdadeira?
 - a) É um protocolo proprietário da Cisco
 - b) É executado na camada de rede
 - c) Somente pode descobrir informações de roteadores mas não de switches
 - d) É executado na camada física e na camada de enlace de dados
9. Um engenheiro de rede deve criar um diagrama de uma rede composta por dispositivos de diversos fabricantes. Qual comando deve ser configurado nos dispositivos Cisco para que a topologia de rede possa ser mapeada?
 - a) Device(Config)#lldp run
 - b) Device(Config)#cdp run
 - c) Device(Config-if)#cdp enable
 - d) Device(Config)#flow-sampler-map topology
10. Qual função o 'SNMP agent' executa?
 - a) Envia informações sobre variáveis MIB em resposta às solicitações do NMS
 - b) Coordena a autenticação entre um dispositivo de rede e um servidor TACACS + ou RADIUS
 - c) Solicita informações dos dispositivos remotos sobre eventos catastróficos do sistema.

- d) Gerencia o roteamento entre os dispositivos de Camada 3
11. Quais são as duas funções do protocolo DHCP?
- a) O servidor DHCP oferece a capacidade de excluir endereços IP específicos de um pool de endereços IP
 - b) O cliente DHCP pode solicitar até quatro endereços de servidor DNS
 - c) O servidor DHCP atribui endereços IP sem exigir que o cliente os renove
 - d) O servidor DHCP entrega\aluga (lease) endereços IP de clientes dinamicamente
 - e) O cliente DHCP mantém um pool de endereços IP
12. Quais são as duas principais razões pelas quais um administrador de rede usaria o CDP?
- a) Para verificar o tipo de cabo que interconecta dois dispositivos
 - b) Para determinar o status dos serviços de rede em um dispositivo remoto
 - c) Para obter informações de VLAN dos switches diretamente conectados
 - d) Para verificar a conectividade de Camada 2 entre dois dispositivos quando a Camada 3 falhar
 - e) Para obter o endereço IP de um dispositivo conectado, a fim de telnet para o dispositivo
 - f) Para determinar o status dos protocolos de roteamento entre roteadores conectados diretamente
13. Qual comando determinar os endereços que foram atribuídos por um servidor DHCP?
- a) Show ip DHCP database
 - b) Show ip DHCP pool
 - c) Show ip DHCP binding
 - d) Show ip DHCP server statistic
14. Qual informação podemos encontrar em um servidor DHCP?
- a) Uma lista de endereços IP disponíveis em um pool
 - b) Uma lista de endereços IP públicos e seus nomes resolvidos pelo DNS
 - c) Nomes de usuários e senhas dos usuários do domínio
 - d) Lista de endereços MAC atribuídos estaticamente
15. Qual tecnologia deve ser implementada para configurar o monitoramento dos dispositivos de rede com a mais alta segurança?
- a) Syslog
 - b) NetFlow
 - c) IP SLA
 - d) SNMPv3
16. Qual comando deve ser inserido para configurar o DHCP relay?
- a) ip helper-address
 - b) ip address dhcp
 - c) ip dhcp pool
 - d) ip dhcp relay
17. Qual protocolo um dispositivo IPv4 usa para obter um endereço IP atribuído de forma dinâmica?
- a) ARP
 - b) DHCP
 - c) CDP
 - d) DNS
18. Um analista de rede deve configurar a data e a hora em um roteador usando o modo EXEC. A data deve ser definida para 12h. Qual comando deve ser usado?
- a) Clock timezone
 - b) Clock summer-time-recurring
 - c) Clock summer-time date
 - d) Clock set
19. No QoS, qual método de priorização é apropriado para chamadas interativas de voz e vídeo?
- a) expedited forwarding
 - b) traffic policing
 - c) round-robin scheduling
 - d) low-latency queuing

Respostas: 1) a 2) b 3) a, d 4) c, 5) c, 6) a, b 7) d 8) a, 9) a 10) a 11)a, d 12) d, e 13) c 14) a 15) d 16) a 17) b 18) d 19) d

5.0 Security Fundamentals

Chegamos ao bloco totalmente dedicado aos fundamentos de segurança. É importante ressaltar que este é um bloco onde a teoria é tão ou mais importante do que a prática, por isso, é um bloco mais teórico.

5.1 Define key security concepts (threats, vulnerabilities, exploits, and mitigation techniques)

A popularização das redes de computadores e da Internet trouxe uma série de ameaças à segurança. Felizmente, as pessoas estão tomando maior consciência da importância desse tema e da responsabilidade que todos possuem.

A segurança é uma das principais preocupações dos engenheiros e analistas de rede. A cada dia que passa os ataques estão mais sofisticados, novas brechas e ataques surgem dia após dia. Além disso, os atacantes aproveitam descuidos dos usuários e administradores para explorar novas formas de ataque. Manter uma rede segura é uma tarefa ingrata. Neste bloco, vamos discutir as ameaças à segurança, vulnerabilidades e técnicas de **mitigação**.

Threats

Uma ameaça à segurança cibernética refere-se a qualquer possível ataque malicioso que visa acessar dados ilegalmente, interromper as operações digitais ou danificar informações. As threats (ameaças) cibernéticas podem se originar de vários atores, incluindo hacktivistas, grupos terroristas, Estados-nação hostis, organizações criminosas, hackers solitários e funcionários descontentes.

Nos últimos anos, vários ataques cibernéticos de alto perfil resultaram na exposição de inúmeros dados confidenciais. Por exemplo, a violação da Equifax. Em 2017, uma invasão nos sistemas da Equifax comprometeu os dados pessoais de cerca de 143 milhões de consumidores, incluindo datas de nascimento, endereços e números da previdência social. Em 2018, a Marriott International divulgou informações de que hackers acessaram seus servidores e roubaram dados de cerca de 500 milhões de clientes. Em ambos os casos, a ameaça à segurança cibernética foi possibilitada pela falha da organização em implementar, testar e *retestar* as proteções técnicas, como criptografia, autenticação e firewalls.

Entre outras ações potencialmente prejudiciais, cyber attackers podem usar os dados confidenciais de um indivíduo ou empresa para roubar informações ou obter acesso às suas contas bancárias, razão pela qual é importantíssimo para o administrador de redes conhecer o mínimo de segurança.

Abaixo, as cinco principais ameaças à segurança de uma rede:

1. Malware

Malware é um termo genérico para qualquer tipo de “**malicious software**” (“software malicioso”) projetado para se infiltrar em um dispositivo sem o conhecimento do usuário. Existem muitos tipos de malware. Cada um funciona de maneira diferente na busca de seus objetivos. No entanto, todas as variantes de malware compartilham duas características fundamentais: são sorrateiras e trabalhamativamente contra os interesses dos usuários.

Adware, spyware, vírus, botnets, cavalos de Troia, worms, rootkits e ransomware, todos se enquadram na definição de malware. É também importante observar que o malware não é apenas uma ameaça para computadores com Windows. Macs e dispositivos móveis também são alvos frequentes.

Malware é um vírus? Sim e não. Embora todos os vírus de computador sejam malware, nem todo malware é um vírus. Os vírus são apenas um tipo de malware. Muitas pessoas usam os dois termos de forma intercambiável, mas do ponto de vista técnico, vírus e malware não são a mesma coisa.

Pense assim: malware é um código malicioso. Vírus de computador são códigos maliciosos que se espalham por computadores e redes.

Independentemente do tipo, todo malware segue o mesmo padrão básico: O usuário, sem querer, baixa ou instala o malware, infectando assim o dispositivo.

A maioria das infecções por malware ocorre quando ele é baixado sem querer. Isso pode acontecer ao clicar em um link, em um e-mail ou ao visitar um site mal-intencionado. Em outros casos, os cibercriminosos espalham malware por meio de serviços de compartilhamento de arquivos peer-to-peer ou em pacotes de download de software gratuitos. Incorporar um malware em um torrent ou em um arquivo popular é uma maneira eficaz de espalhá-lo por uma ampla base de usuários. Os dispositivos móveis também podem ser infectados por SMS.

Outra técnica é carregar o malware no firmware de um pen drive ou unidade flash USB. Como o malware é carregado no hardware interno do dispositivo (e não no armazenamento de arquivos), é improvável que o ele seja detectado. Por isso, nunca se deve inserir um pen drive desconhecido em um computador.

Depois que o malware é instalado, ele infecta o dispositivo e começa a trabalhar para os cibercriminosos. O que separa os vários tipos de malware é a maneira como eles fazem isso.

Tipos comuns de malware:

A grande maioria de malware se enquadra nas seguintes categorias básicas, dependendo de seu funcionamento:



Ransomware: É a versão malware da carta de resgate de um sequestrador. Normalmente, ele bloqueia ou nega o acesso ao dispositivo e/ou encripta pedindo um resgate em troca, um tipo de sequestro de dados. Pessoas ou grupos que armazenam informações vitais em seus dispositivos correm risco com a ameaça de ransomware.

Spyware: Coleta informações sobre um dispositivo ou rede e transmite esses dados para o invasor. Os cibercriminosos normalmente usam spyware para monitorar a atividade de uma pessoa na Internet e coletar dados pessoais, incluindo credenciais de login, números de cartão de crédito ou informações financeiras, para fins de fraude ou roubo de identidade.

Worms: São projetados com um objetivo em mente: proliferação. Um worm infecta um computador e se replica em seguida, espalhando-se para dispositivos adicionais enquanto permanece ativo em todas as máquinas infectadas. Alguns worms atuam como agentes de entrega para instalação de malware adicional. Outros tipos são projetados apenas para se espalhar, sem causar danos intencionais às máquinas host, mas eles sobrecarregam as redes, por vezes, ocupando toda a largura de banda.

Adware: O trabalho do adware é gerar receita para o desenvolvedor exibindo para a vítima anúncios indesejados. Tipos comuns de adware incluem jogos gratuitos ou barras de ferramentas do navegador. Eles coletam dados pessoais sobre a vítima para usá-los para personalizar os anúncios que exibem. Embora a maioria dos adwares seja instalada legalmente, eles são tão irritantes quanto outros tipos de malware.

Cavalos de Troia: Os poetas gregos antigos contavam que os guerreiros atenienses se esconderam dentro de um cavalo de madeira gigante e saíram depois que os troianos o puxaram para dentro das muralhas da cidade. Um cavalo de Tróia é, portanto, um veículo para atacantes ocultos. Cavalo do Troia é malware que se infiltra no dispositivo da vítima apresentando-se como software legítimo. Uma vez instalado, o cavalo do Troia é ativado, podendo até mesmo baixar malwares adicionais.

Botnets: Um botnet não é um tipo de malware, mas uma rede de computadores ou código de computador que pode transmitir ou executar malware. Os invasores infectam um grupo de computadores com software malicioso, conhecidos como "bots". Esses Bots podem receber comandos de seu controlador. Esses computadores formam uma rede, fornecendo ao controlador acesso a um poder de processamento coletivo substancial, que pode ser usado para coordenar ataques, enviar spam, roubar dados e criar anúncios falsos nos navegadores.

2. Denial of Service – DoS (Negação de serviço)

O ataque do tipo DoS (Denial Of Service, em inglês), também conhecido como ataque de negação de serviço, é uma tentativa de fazer com que aconteça uma sobrecarga em um servidor ou computador fazendo com que recursos do sistema fiquem indisponíveis para seus utilizadores.

Para isso, o invasor utiliza meios para enviar diversos pedidos de pacotes para o alvo, com a finalidade de que este fique tão sobrecarregado que não consiga mais responder a nenhuma requisição. Assim, os utilizadores não conseguem mais acessar dados do computador, por ele estar indisponível e não conseguir responder a novos pedidos.

Os alvos mais comuns dos ataques de negação de serviço são servidores web. Com o ataque, o hacker ou cracker tenta tornar as páginas hospedadas indisponíveis. Esse ataque não se caracteriza como uma invasão do sistema, visto que ele realiza apenas a invalidação por meio de sobrecarga, sem ter acesso ou danificar nenhum dos dados internos do sistema.

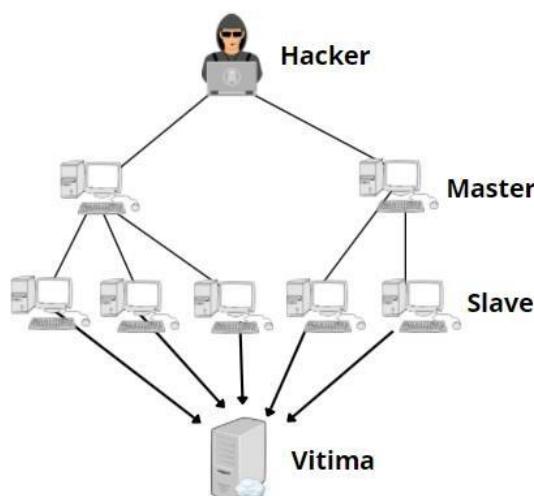
Os atacantes geralmente utilizam a obstrução do meio de comunicação entre os utilizadores e o sistema de modo a não se comunicarem corretamente. Outra maneira de realizar o ataque é forçar a vítima a reinicializar ou consumir todos os recursos de memória, processamento ou de outro hardware de modo a deixá-lo impossibilitado de fornecer o serviço.

Os ataques DoS envolvem apenas um atacante, sendo um único computador a fazer vários pedidos de pacotes para o alvo. Nesse tipo de ataque, o hacker pode apenas derrubar servidores com pouco poder de processamento e computadores comuns com pouca banda e com baixas especificações técnicas.

Ataques DDoS

Já no ataque distribuído de negação de serviço, conhecido como DDoS (Distributed Denial of Service, em inglês) .O ataque acontece de forma similar ao DoS, porém, ele ganha algumas camadas extras. Nele, um computador mestre pode gerenciar uma série de outros computadores, que são chamados de zumbis.

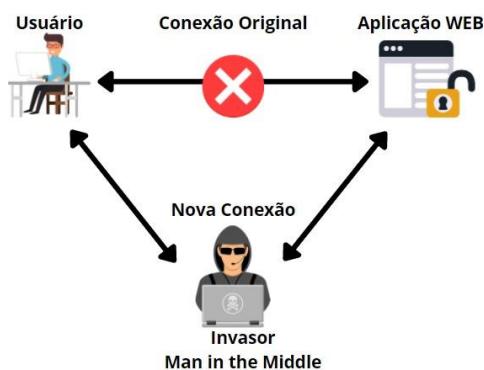
Por meio do DDoS, o hacker ou cracker invade um computador mestre e este, por sua vez, escraviza várias máquinas, fazendo com que elas passem a acessar um determinado recurso em um servidor ao mesmo tempo. Assim, todos os zumbis acessam juntamente e de maneira ininterrupta o mesmo recurso de um servidor, tentando sobrecarregá-lo.



Levando em consideração que a maioria dos servidores web possuem uma capacidade limitada de usuários que podem atender ao mesmo tempo, esse grande número de tráfego impossibilita que o servidor seja capaz de atender novos pedidos. Então, o servidor pode reiniciar ou mesmo ficar travado dependendo do recurso que foi vitimado.

3. Man-in-the-middle (Homem no Meio)

O conceito por trás do ataque MITM é bastante simples e não se restringe ao universo online. O invasor se posiciona entre duas partes que tentam comunicar-se, intercepta mensagens enviadas e depois se passa por uma das partes envolvidas. O envio de contas e faturas falsas poderia ser um exemplo desta prática no mundo offline, o criminoso as envia ao correio das vítimas e rouba os cheques enviados como forma de pagamento. No universo online, os ataques são mais complexos. Apesar de basear-se na mesma ideia, o invasor deve permanecer invisível entre a vítima e uma instituição verdadeira para que o golpe tenha sucesso.



Variantes do ataque MITM

Na forma mais comum de MITM o golpista usa um roteador WiFi como mecanismo para interceptar conversas de suas vítimas, o que pode se dar tanto através de um roteador corrompido quanto através de falhas na instalação do equipamento. Numa situação comum o agressor configura seu laptop, ou outro dispositivo wireless, para atuar como ponto de WiFi e o nomeia com um SSID comum a redes públicas. Então quando um usuário se conecta ao “roteador” e tenta navegar em sites delicados como de bancos ou comércio eletrônico o invasor rouba suas credenciais.

Uma versão mais recente do ataque MITM é chamada man-in-the-browser. Nesta variável o agressor usa um dos inúmeros métodos para implementar um código malicioso no browser do computador da vítima. O malware silenciosamente grava informações enviadas a vários sites. Esta modalidade tem se popularizado ao longo dos anos porque possibilita atacar a um grande volume de usuários por vez e mantém o criminoso a uma distância segura (para ele) de suas vítimas.

4. Phishing

Phishing é um termo originado do inglês (fishing) que em computação se trata de um tipo de roubo de identidade online. Essa ação fraudulenta é caracterizada por tentativas de adquirir ilicitamente dados pessoais de outra pessoa, sejam senhas, dados financeiros, dados bancários, números de cartões de crédito ou simplesmente dados pessoais.

O Termo Phishing foi criado em meados de 1996 por cibercriminosos que praticavam roubo de contas da AOL (America Online). Um ano depois, em 1997, o termo passou a ser citado na mídia e a partir de então se tornou popular. Naquela época, as contas hackeadas já podiam ser utilizadas como moeda de troca no mundo hacker. Trocas como 10 phishs (contas hackeadas) por uma parte de um programa malicioso aconteciam com frequência no universo dos cibercriminosos. Hoje, o Phishing desenvolveu-se e tornou-se muito mais poderoso e obscuro do que antigamente.

O fraudador utiliza e-mail, aplicativos e sites projetados especificamente para roubar dados pessoais. O criminoso se faz passar por uma pessoa ou empresa confiável enviando uma mensagem para conseguir atrair suas vítimas. Dessa maneira, ao enviar uma mensagem para um e-mail, aplicativo ou outras ferramentas, o fraudador apenas aguarda até que o destinatário receba e abra a mensagem. Em muitos casos, isso já basta para que a vítima caia no golpe. Em outros é preciso que a vítima clique em um determinado link para que assim o criminoso tenha acesso às informações que deseja.

Os golpistas enviam milhões de mensagens por dia, na esperança de encontrar usuários inexperientes que possam ser vítimas do ataque. Eles adotam o envio massivo de spams e acabam obtendo um razoável sucesso, ultrapassando 5% em alguns casos, segundo dados do Anti-Phishing Working Group.

Os ataques de Phishing basicamente funcionam em seis etapas: planejamento, preparação, ataque, coleta, fraude e pós-ataque.

Na primeira etapa, os golpistas escolhem os seus alvos e definem qual será o objetivo do ataque. É nesse ponto que definem se a intenção é conseguir dados pessoais, dados bancários, criar contas em nome da vítima, transferir dinheiro para uma outra conta bancária ou diversos outros tipos de fraudes.

Na fase da preparação, os fraudadores criam o material que servirá como "isca" para vitimar as pessoas. É nesse momento que as mensagens são elaboradas, bem como os e-mails, os sites e os links que serão utilizados durante o crime.

A etapa do ataque consiste no envio das mensagens elaboradas. Elas podem ser feitas via e-mail, por meio de sites, malwares, VoIP, ou aplicativos de mensagens instantâneas.

Na fase seguinte, o *cibercriminoso* coleta os dados obtidos após o ataque e os prepara para serem usados para finalizar o crime. Assim, a etapa da fraude começa a acontecer quando o golpista usa os dados que possui para acessar uma conta, criar novas identidades, roubar dinheiro ou realizar algum outro tipo de crime utilizando-se de dados da vítima. Ainda, os golpistas podem utilizar os dados que coletaram para vender a outras pessoas ou usá-los em um próximo ataque.

Na última etapa, o cracker destrói os mecanismos para a criação e execução do ataque com o objetivo de eliminar evidências. Em casos onde o roubo inclui dinheiro, essa etapa envolve formas de lavagem para dificultar qualquer tipo de investigação policial.

Alguns cuidados simples podem ser tomados para evitar que ataques assim possam acontecer. Como não abrir mensagens que sejam de um remetente desconhecido, não executar arquivos baixados automaticamente ou realizar downloads de fontes não confiáveis.

5. Ataque de força bruta (Brute Force attack)

Um ataque de força bruta ocorre quando hackers utilizam o processo de tentativa e erro auxiliado por computadores para tentar quebrar uma senha. O escopo e a definição de força bruta foram ampliados à medida que a tecnologia evoluiu.

Nos anos 1970, um hacker poderia, na teoria, tentar apenas milhares de variações de senha diferentes a cada segundo. Hoje, a computação moderna possibilita centenas de bilhões de tentativas de login por segundo.

Embora o significado de força bruta tenha sido ampliado, o método permanece igual: tentar o máximo de combinações de senha possíveis até encontrar senha certa. Para o hacker descobrir a senha correta geralmente é resultado do tempo e dos recursos que ele está disposto a gastar. Mas quais métodos os hackers modernos usam?

Os cinco tipos mais comuns de ataques de força bruta são: ataques simples, ataques de dicionário, ataques híbridos, ataques reversos e “stuffing” de credenciais. Todas as pessoas interessadas e com algum know-how podem adquirir uma ferramenta de “*descriptografia* por força bruta”, que é um tipo de software que conduz automaticamente ataques de força bruta.

Na maior parte do tempo, as pessoas usam ferramentas de força bruta para quebrar senhas ou *descriptografar* bancos de dados de senhas roubadas. A eficácia de uma ferramenta de força bruta depende dos recursos e poder de computação dos seus criadores.

O típico hacker solitário em um quarto pode não ter dinheiro para comprar o computador mais avançado para quebra de senhas. Mas a definição de hacker mudou ao longo do tempo. Hoje, muitos *cibercriminosos* fazem parte de grupos bem financiados e bem organizados, com acesso às principais técnicas de quebra de senha disponíveis.

Ataques simples de força bruta

Ataques simples de força bruta exigem pouco poder de computação e engenhosidade. Eles experimentam sistematicamente combinações de palavras, letras e caracteres até terem sucesso. Senhas longas e complexas estão além do escopo desses ataques, que normalmente são limitados a variações das senhas mais comuns ou prováveis.

Realizar um ataque simples de força bruta é tão fácil que pode ser feito manualmente, embora, claro, isso seja mais demorado.

Um *bot* pode quebrar facilmente uma senha previsível com força bruta. Por isso, algumas das piores senhas são números sequenciais (123456), nome de pessoas, data de aniversário, ou a notória (e ainda surpreendentemente popular) “senha”.

Ataques simples de força bruta ainda são eficazes porque muitos usuários não percebem o perigo de usarem senhas simples.

Ataques de dicionário

Ataques de dicionário visam senhas mais obscuras e usam um dicionário digital ou lista de palavras como auxílio. Escolher uma palavra mais obscura para uma senha pode ajudar a proteger contra ataques de hacking simples de força bruta, pois muitos hackers desistirão se demorarem muito. Mas usar palavras mais obscuras ou complexas não protegerá contra ataques de dicionário.

Ataques de dicionário tentam adivinhar a senha passando por cada palavra, combinações comuns dessa palavra com outras, variações de grafia e palavras em vários idiomas. Se você estiver usando uma única palavra como senha, um ataque de dicionário de força bruta terá sucesso em segundos.

Ataques híbridos de força bruta

Ataques híbridos de força bruta combinam simples híbridos de força bruta e ataques de dicionário. Senhas comuns são misturadas com palavras de dicionário e caracteres aleatórios para criar um banco de dados maior de combinações de senha. Uma senha como “5enh4” pode enganar ataques de dicionário, mas oferece pouca proteção contra um ataque híbrido.

Hackers que usam ataques híbridos personalizarão a estratégia de ataque, em vez de simplesmente tentar uma palavra por vez. O invasor conhece as combinações de palavras prováveis com base em listas de palavras (talvez compradas na dark web), dados demográficos do alvo e conhecimento geral do comportamento humano. Ele então configura o ataque para priorizar essas combinações.

Ataques reversos de força bruta

Ataques reversos de força bruta invertem a ordem das operações: Eles começam com uma senha comum ou conhecida e tentam adivinhar o nome de usuário por força bruta. As senhas de violações de dados às vezes vazam online. Quando isso acontece, elas costumam ser usadas para lançar ataques reversos.

Muitas pessoas nunca consideram a segurança do ID de login, o que torna o hack de força bruta de nomes de usuário mais lucrativo do que pode parecer.

“Stuffing” de credenciais

“Stuffing” de credenciais ocorre quando um hacker obtém seu nome de usuário e sua senha de um site e depois tenta fazer login com as mesmas credenciais em outros sites. Em vez de um ataque de força bruta que visa uma senha ou nome de usuário, ele tenta descobrir por força bruta os locais em que a senha ou o nome do usuário é usado. Por isso, esse é um dos motivos para nunca salvar senhas no navegador.

Se você usar a mesma senha ou nome de usuário em vários sites, como muitos fazem, basta que uma seja comprometida e nenhuma estará segura. Além de usar senhas exclusivas em todas as suas contas, considere aumentar sua segurança com uma ferramenta antivírus.

Vulnerabilities

Uma vulnerabilidade é uma falha ou fraqueza em um sistema ou rede que pode ser explorada para causar danos ou permitir que um invasor manipule o sistema de alguma forma.

Isso é diferente de uma “ameaça cibernética” porque, embora uma ameaça cibernética possa envolver um elemento externo, existem vulnerabilidades intrínsecas aos dispositivos da rede (por exemplo, um computador, banco de dados ou até mesmo um dispositivo de rede como roteador). Além disso, eles geralmente não são o resultado de esforço intencional de um invasor - embora os cibercriminosos aproveitem essas falhas em seus ataques.

A maneira como uma vulnerabilidade é explorada depende da natureza da vulnerabilidade e dos motivos do invasor. Essas vulnerabilidades podem existir devido a interações imprevistas de diferentes programas, componentes de hardware. É importante saber que as vulnerabilidades estão presentes em praticamente todas as redes - não há como identificá-las e resolvê-las por causa da natureza incrivelmente complexa da arquitetura de uma rede moderna.

No entanto, é possível reduzir significativamente o risco de violação de dados ou eventos semelhantes, conhecendo algumas das vulnerabilidades mais comuns e encontrando maneiras de resolvê-las.

As vulnerabilidades de segurança podem ser divididas em vários tipos com base em diferentes critérios, por exemplo: Local ou dispositivo onde está a vulnerabilidade, o que a causou ou como ela pode ser usada. Algumas categorias de vulnerabilidade incluem:

- **Vulnerabilidades de rede:** São problemas com hardware ou software em uma rede que a expõe a uma possível invasão de terceiros. Os exemplos incluem pontos de acesso Wi-Fi inseguros e firewalls com patches de atualização desatualizados.
- **Vulnerabilidades do sistema operacional:** São as vulnerabilidades que ocorrem em um sistema operacional específico. Os hackers podem explorar essa vulnerabilidade para obter acesso a um ativo no qual o sistema operacional está instalado - ou para causar danos. Os exemplos incluem contas de superusuário padrão que podem existir em algumas instalações de SO e programas de backdoor ocultos.
- **Vulnerabilidades humanas:** O elo mais fraco em muitas arquiteturas de segurança cibernética é o elemento humano. Os erros cometidos por usuários podem facilmente expor dados confidenciais, criar pontos de acesso exploráveis para invasores ou interromper sistemas.
- **Vulnerabilidades de processo:** Algumas vulnerabilidades podem ser criadas por controles de processos específicos (ou a falta deles). Um exemplo seria o uso de senhas fracas (que também pode ser considerada uma vulnerabilidade humana).

Exploits

Um exploit é um programa, ou um pedaço de código, projetado para encontrar e tirar proveito de uma falha de segurança ou vulnerabilidade em um aplicativo ou sistema de computador, normalmente, para fins maliciosos, como instalação de malwares. Uma exploração não é um malware em si, mas sim um método usado pelos cibercriminosos para distribuir o malware.

Algumas vulnerabilidades exigem que o invasor inicie uma série de operações suspeitas para configurar uma exploit. Normalmente, a maioria das vulnerabilidades são resultado de um bug de software ou da própria arquitetura do sistema. Os invasores escrevem um código para aproveitar essas vulnerabilidades e injetar vários tipos de malwares no sistema.

Muitos fornecedores de software corrigem bugs conhecidos para remover essas vulnerabilidades. Os softwares de segurança também ajudam a detectar, relatar e bloquear operações suspeitas. Abaixo, os dois tipos de exploits:

1. Exploits conhecidos

Depois que um exploit é informado aos autores do software afetado, a vulnerabilidade geralmente é corrigida por meio de um patch de correção, tornando assim o exploit inutilizável. Essas informações também ficam disponíveis para fornecedores de softwares e equipamentos de segurança para que eles tomem suas próprias medidas.

2. Exploits desconhecidos

Exploits desconhecidos por todos, exceto pelas pessoas que desenvolveram, são chamadas de “zero-day exploits”. Esses são certamente os exploits mais perigosos, pois ocorrem quando um software ou arquitetura de sistema contém uma vulnerabilidade de segurança crítica e o fornecedor\desenvolvedor ainda não tem conhecimento dessa falha.

A vulnerabilidade torna-se conhecida apenas quando um hacker é detectado explorando-a, daí o termo zero-day exploits (exploração do dia zero). Uma vez que tal exploit ocorra, os sistemas que executam o software ficam vulneráveis a um ataque cibernético. A única chance nesse caso, é que o dispositivo ou software de segurança detecte e bloqueeie o malware resultante do exploit por comportamento estranho.

Mitigation techniques

As técnicas e métodos de mitigação dependem principalmente do tipo de ameaça. Abaixo as principais técnicas de mitigação:

1. Treinamento e conscientização:

O treinamento do usuário é considerado a técnica de mitigação mais barata e eficaz. É a melhor maneira de evitar que os usuários cometam erros que levem a possibilidade de ataques de engenharia social. Esses treinamentos devem mostrar ao usuário que é importante conhecer e seguir os procedimentos, protocolos e políticas de segurança. Portanto, o treinamento de usuários oferece uma vantagem real a um custo relativamente baixo.

2. Gerenciamento adequado dos Patches de segurança:

Quando um aplicativo ou sistema operacional é lançado, ele não é perfeito do ponto de vista da segurança. Depois do lançamento, atualizações e patches de segurança são lançados continuamente. Os sistemas de atualização do Windows são um bom exemplo disso que estamos falando. É vital manter os dispositivos e aplicações com patchs de segurança atualizados.

3. Políticas e procedimentos:

Os procedimentos e políticas de segurança devem ser descritos de forma clara e direta, permitindo assim uma fácil compreensão por todos os usuários. Os procedimentos e políticas de segurança devem definir os comportamentos aceitáveis em redes de computadores da organização, como os sites e serviços que podem ser acessados. Os usuários devem ler e aceitar as políticas e procedimentos, de preferência assinando um formulário de concordância.

5.2 Describe security program elements (user awareness, training, and physical access control)

Os elementos do programa de segurança são essenciais para manter uma rede de computadores bem protegida. Eles incluem a explicação e conscientização da importância de treinamentos constantes, políticas, procedimentos e ameaças. Um programa de conscientização sobre segurança pode ajudar muito nos esforços de uma empresa para melhorar e manter a segurança. Esses esforços precisam ser contínuos e devem fazer parte da prática cotidiana de comunicação da organização.

Conscientização do usuário:

A educação e a conscientização ajudam a garantir que as informações de segurança sejam transmitidas às pessoas apropriadas em tempo hábil. A maioria dos usuários não está ciente das ameaças modernas. Por isso, é necessário estabelecer processos para explicar de forma concisa e clara o que está acontecendo e o que está sendo feito para corrigir essas ameaças. Esse tipo de abordagem aumenta a aceitação dos usuários, pois o conscientiza de que ele é um dos atores principais nessa engrenagem. Publicar informações por meio de sites na intranet da empresa e através do e-mail institucional são métodos educacionais eficazes para conscientização dos usuários. Pode-se adotar um processo de notificação regular para transmitir informações sobre problemas e mudanças de segurança. Em geral, quanto mais informação publicada de maneira regular, mais as pessoas perceberão o fato de que a segurança é responsabilidade de todos.

Em suma, a conscientização do usuário é um esforço para criar segurança, um pensamento que deve ser comum e regular para todos os membros da equipe. Infelizmente, a conscientização sobre a segurança do usuário é geralmente o elemento mais negligenciado no gerenciamento de segurança. Na verdade, a falta de consciência é o motivo básico do sucesso dos ataques de engenharia social.

Training

Os esforços em educação e treinamento devem ajudar os usuários a compreender claramente a prevenção, a fiscalização e as ameaças. Integrando os esforços da equipe de TI, o departamento de segurança provavelmente também será responsável por um programa de conscientização sobre segurança.

Programas de treinamento educacionais precisam ser personalizados para, pelo menos, três públicos diferentes:

- A Organização
- Sua Gestão
- A Equipe Técnica

Essas três partes organizacionais têm diferentes deliberações e preocupações. Por exemplo, com o treinamento para toda a organização, todos entendem as políticas, procedimentos e recursos disponíveis para lidar com questões de segurança, portanto, ajuda a garantir que todos os funcionários estejam na mesma página. A lista a seguir classifica os tipos de questões que os membros de uma organização devem conhecer e compreender.

1. Organização

De preferência, um programa de treinamento de conscientização de segurança para toda a organização deve abranger as seguintes áreas:

- Importância da segurança;
- Responsabilidades das pessoas na organização;
- Políticas e procedimentos;
- Políticas de uso;
- Critérios de seleção de conta e senha;
- Prevenção de engenharia social;

Este treinamento pode ser realizado usando uma equipe interna ou contratando instrutores externos. Recomenda-se fazer esse tipo de treinamento durante a orientação a novos funcionários e reuniões de equipe. O treinamento precisa ser repetido periodicamente (duas vezes por ano geralmente funciona bem). Além disso, é necessário coletar a assinatura dos funcionários como prova de que receberam o treinamento e estão cientes das políticas.

2. Gestão

Os gerentes estão preocupados com questões mais universais da organização, incluindo a implementação de políticas e procedimentos de segurança.

Os gerentes necessitam saber o propósito e as razões de um programa de segurança; como funciona e por que é necessário. Eles devem receber treinamento adicional ou exposição que descreva os problemas, ameaças e as técnicas para lidar com essas questões.

A gerência também deve se preocupar com os efeitos da produtividade, aplicação e como os vários departamentos são afetados pelas políticas de segurança.

3. Equipe técnica

A equipe técnica necessita de conhecimento especial sobre os métodos, implementações e recursos dos sistemas usados para gerenciar a segurança.

Os administradores de rede desejarão avaliar como gerenciar a rede, as práticas recomendadas e os problemas de configuração relacionados às tecnologias que eles suportam.

Os desenvolvedores e implementadores desejarão avaliar o efeito dessas medidas nos sistemas existentes e nos novos projetos de desenvolvimento.

O treinamento que os administradores e desenvolvedores precisam deve ser específico do fornecedor, pois os fornecedores possuem seus próprios métodos de implementação de segurança.

Lembre-se de que todos os seus esforços serão em vão se não alcançarem o público apropriado. Gastar uma hora pregando sobre segurança de banco de dados provavelmente será uma hora perdida se os únicos membros da audiência forem operadores de entrada de dados que são pagos para simplesmente digitar e fazer alterações básicas.

Physical access control

Os controles de acesso físico são mecanismos projetados para minimizar o risco de danos causados por pessoas não autorizadas em ambientes críticos. Um exemplo simples é uma fechadura de porta inteligente, que não permitirá que pessoas não autorizadas tenham acesso a locais sensíveis; a instalação de sensores biométricos, como varredura da íris ou reconhecimento de impressão digital são ótimas opções para aumentar a segurança física do ambiente.

Várias empresas estão retirando as interfaces USB, COM, LPT dos computadores para evitar que funcionários mal intencionados roubem informações ou insiram algum software prejudicial. Além disso, a utilização de senha na BIOS do computador ajuda na proteção em caso de roubo ou perda do dispositivo.

5.3 Configure device access control using local passwords

O uso de proteção por senha para controlar ou restringir o acesso à interface de linha de comando (CLI) de dispositivos (roteadores, switches, firewall, etc), é um dos elementos fundamentais de um plano de segurança.

Existem alguns tipos de conexões físicas possíveis quando estamos falando de roteadores e switches. Por exemplo, a entrada CTY, que é a porta console. Em qualquer roteador e switch, ele aparecerá na configuração como ‘line con 0’ e na saída do comando ‘show line’, como CTY.

A porta console é usada principalmente para acesso ao sistema local por meio de um terminal de console. A linha AUX é a porta auxiliar, vista na configuração como linha aux 0. As linhas VTY são as linhas do Terminal Virtual do roteador, usadas exclusivamente para controlar as conexões Telnet e SSH de entrada. Elas são virtuais, no sentido de que são uma função do software - não há hardware associado a elas. Elas aparecem na configuração como linha vty 0 4.

Cada linha dessa pode ser configurada com proteção por senha. As linhas podem ser configuradas para usar uma senha para todos os usuários ou com senhas específicas para cada um dos usuários. As senhas específicas do usuário podem ser configuradas localmente no roteador ou através de um servidor de autenticação, como o TACACS. É comum ver roteadores com uma única senha para o console e senhas específicas de usuários para outras conexões de entrada.

Configurando senhas para usuário local

Para estabelecer um sistema de autenticação baseado em nome de usuário, basta utilizar o comando **username** no modo de configuração global, e para habilitar a verificação local de senha no momento do login, use o comando **login local**, no modo de configuração de linha **vty**.

```
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#username luiz password Cisco
Router(config)#username Joao password Brasil
Router(config)#line vty 0 4
```

```
Router(config-line)#login local  
Router(config-line)#end
```

Configurar senha da linha AUX

Para configurar uma senha na linha AUX, digite o comando **password** no modo de configuração da linha **aux**. Para habilitar a verificação de senha no login, basta aplicar o comando **login**.

```
Router(config)#line aux 0  
Router(config-line)#password C!$c0!@#  
Router(config-line)#login  
Router(config-line)#end
```

É necessário controle físico para limitar o acesso das pessoas até o hardware dos dispositivos de redes, e proteções como login e senha para manter a segurança logica dos equipamentos.

5.4 Describe security password policies elements, such as management, complexity, and password alternatives (multifactor authentication, certificates, and biometrics)

Sem uma política de segurança adequada, a disponibilidade da rede pode ser comprometida. A política começa com avaliação de risco e a formação de uma equipe para responder a qualquer eventualidade. Uma implementação bem sucedida exige maior segurança ao acesso à rede, essa maior segurança pode ser implementada com o uso de vários mecanismos de autenticação que veremos a seguir.

Gerenciamento de senhas:

As senhas são um conjunto de ‘strings’ fornecidas pelos usuários nos prompts de autenticação. Embora as senhas ainda sejam consideradas um dos métodos mais seguros de autenticação, elas estão expostas a uma série de ameaças à segurança quando utilizadas de forma incorreta. O papel do gerenciamento de senhas surge nesse cenário.

O gerenciamento de senhas é um conjunto de princípios e boas práticas a serem seguidos pelos usuários durante o armazenamento e utilização, visando protegê-las e impedir o acesso de pessoas não autorizadas. Essas boas práticas incluem:

- Usar senhas fortes e exclusivas para cada site e aplicação;
- Redefinição das senhas após um determinado período de tempo;
- Configuração e autenticação de dois fatores para todas as contas;
- Não compartilhar senhas com amigos, familiares e colegas;
- Periodicamente analisar as possíveis violações e a partir dessa análise, tomar as medidas necessárias.

Complexidade de senha:

Uma política de senha é um conjunto de regras escritas como parte da política de segurança organizacional que dita os requisitos mínimos de senhas dos usuários e dos dispositivos, bem como uma ferramenta técnica de aplicação, que impõe as regras de senha a serem adotadas.

A política de senha geralmente inclui os requisitos de comprimento mínimo, duração máxima em meses, histórico de senhas e alguns requisitos de complexidade, que podem ser, por exemplo, um mínimo de três tipos de caracteres diferentes (letras maiúsculas e minúsculas, números e símbolos), ou que a senha não possa ter o nome do usuário, nome real e endereço de e-mail. As senhas com mais de 12 caracteres são consideradas seguras e aquelas com mais de 15 caracteres são consideradas muito seguras.

Normalmente, quanto mais caracteres em uma senha, junto com algum tipo de complexidade de caractere, mais resistente ela é às técnicas de quebra de senha, especificamente ataques de força bruta. Exigir mudanças regulares de senha,

como a cada 90 dias, e proibir a reutilização de senhas anteriores (histórico de senha) irá melhorar a segurança de um sistema que usa senhas como o principal meio de autenticação.

Alternativas de senha:

À medida que os ataques focados em senhas aumentam, as empresas e usuários precisam mudar para meios mais avançados de autenticação. Há várias opções que podem substituir as senhas tradicionais, abaixo algumas alternativas:

- **Autenticação multifator:** A autenticação multifator significa autenticar o usuário por dois ou mais métodos de acesso. Um sistema que autentica os usuários por um cartão inteligente, ou que utilize biometria, detecção de íris, e outros, pertencem ao tipo de autenticação multifator.
- **Autenticação de dois fatores:** Forma de autenticar o usuário por meio de algo que ele possui ou conhece. Por exemplo, a autenticação por um cartão inteligente pertence ao tipo de autenticação de dois fatores:
 - **Algo que você sabe:** Um nome de usuário, uma senha, uma frase secreta ou um número de identificação pessoal (PIN).
 - **Algo que você tem:** Um dispositivo de segurança física que autentica usuários, como um cartão inteligente, crachá ou chaveiro.
 - **Algo que você é:** Alguma característica distinta e específica, como uma biometria.
 - **Algum lugar onde você está:** Algum fator de localização, é baseado em geolocalização.
 - **Algo que você faz:** Algumas ações que os usuários devem realizar para concluir a autenticação, como digitar algo no teclado.

Para o exame, tenha em mente que a autenticação de dois fatores é um subconjunto da autenticação de vários fatores.

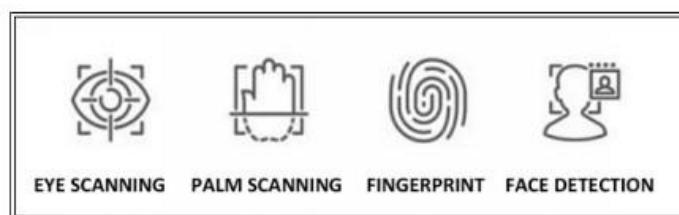
Certificados:

Um certificado é uma forma de credencial digital que valida usuários, computadores ou dispositivos na rede. É uma declaração assinada digitalmente que relaciona as credenciais de uma chave pública à identidade da pessoa, dispositivo ou serviço que contém a chave privada correspondente.

Biometria:

O acesso biométrico é a melhor maneira de construir segurança física, pois usa uma característica física única de cada pessoa para permitir o acesso a um ambiente controlado.

Essas características físicas incluem impressões digitais, impressões de mãos, reconhecimento de voz, varreduras de retina, e assim por diante.



5.5 Describe remote access and site-to-site VPNs

Nesse tópico falaremos sobre a definição de acesso remoto e o que é VPN, em especial site-to-site.

Remote Access

Acesso remoto é uma tecnologia que permite que um computador consiga acessar um servidor privado – normalmente de uma empresa – por meio de um outro computador que não está fisicamente conectado àquela rede. A conexão à distância é feita com segurança de dados em ambos os lados e pode trazer diversos benefícios para manutenção, por exemplo.

Na prática, essa tecnologia é o que permite acessar e-mails e arquivos corporativos fora do local de trabalho, assim como compartilhar a tela do seu computador em aulas ou palestras à distância, de modo a fazer com que o receptor visualize exatamente o que é reproduzido no computador principal e, por vezes, faça edições e alterações mediante permissão no computador remoto.

O acesso remoto também pode ocorrer via Internet, e controlar computadores de terceiros. Seu uso mais frequente é para suporte técnico de softwares, já que o técnico pode ver e até pedir permissões para manipular a máquina completamente sem estar diante do computador.

Utilizando as ferramentas adequadas, é possível acessar computadores com qualquer sistema operacional, em qualquer rede, a partir de desktop, smartphone ou tablet.

A maneira mais comum de usar o acesso remoto é por meio de uma VPN (Rede Privada Virtual), que consegue estabelecer uma ligação direta entre o computador e o servidor de destino – criando uma espécie de "túnel protegido" na Internet. Isto significa que o usuário pode acessar tranquilamente seus documentos, e-mails corporativos e sistemas na nuvem através da VPN, sem preocupação de ser interceptado por administradores de outras redes.

Site-to-site VPNs

Ao usar uma conexão WAN privada de um provedor de serviços, você confia que o ISP manterá a confidencialidade dos seus dados. O provedor de serviços deverá separar seu tráfego dos demais clientes, e certificar que ninguém mais possa ver seus dados.

Porém, quando estamos usando a Internet e desejamos enviar tráfego do ponto A para o ponto B, não temos controle sobre quais redes serão usadas para ir da origem ao destino. Sempre há o risco que alguém no caminho possa estar capturando os pacotes e acessando os dados.

VPNs (Virtual Private Network) ajudam a estabelecer uma conexão segura em uma rede insegura, como a Internet. Esta é uma ótima alternativa para conexões WAN privadas, já que o acesso à Internet geralmente é mais barato e está disponível em quase todos os lugares. VPNs fornecem alguns recursos, como:

- **Confidencialidade:** Impede que alguém não autorizado leia seus dados. Isso é implementado com criptografia.
- **Autenticação:** Verifica se o roteador/firewall ou usuário remoto que está enviando tráfego VPN é o dispositivo ou roteador legítimo (é quem diz ser).
- **Integridade:** Verifica se o pacote VPN não foi alterado de alguma forma durante o trânsito.
- **Anti-replay:** Evita que alguém capture o tráfego e o reenvie tentando aparecer como um dispositivo/usuário legítimo.

Tipos VPN

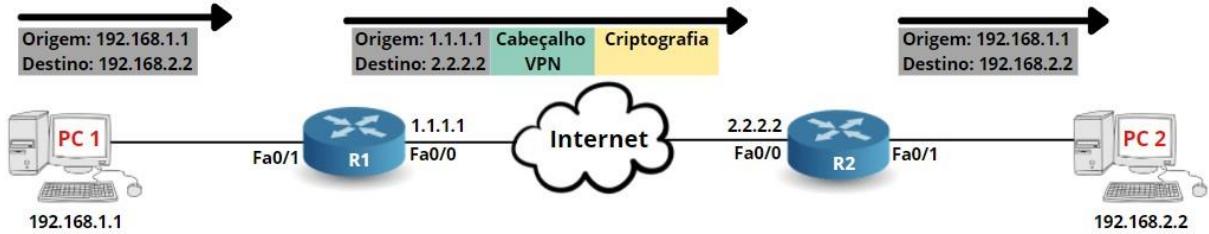
OS dois tipos mais comuns de VPN:

- Site-to-site VPN - VPN site para site
- client-to-site VPN (remote user) - VPN cliente para site (usuário remoto)

Site-to-site VPN

Com a VPN site-to-site, temos um dispositivo de rede em cada site, entre esses dois dispositivos de rede construímos um túnel VPN. Cada extremidade do túnel VPN criptografará o pacote IP original, adicionando um cabeçalho VPN, um novo cabeçalho IP e, em seguida, encaminhará o pacote criptografado para a outra extremidade do túnel.

Eis um exemplo de túnel VPN:



Abaixo, um passo a passo do processo acima:

1. PC1 envia um pacote IP com origem 192.168.1.1 e destino 192.168.2.2.
2. R1 criptografa o pacote IP, adiciona um cabeçalho VPN e cria um novo cabeçalho IP com seu próprio endereço de IP público, tendo como destino o endereço 2.2.2.2.
3. R1 envia o novo pacote para R2.
4. R2 recebe o pacote, verifica se o pacote realmente veio de R1, descriptografa e encaminha para PC2.
5. PC2 recebe o pacote IP original.

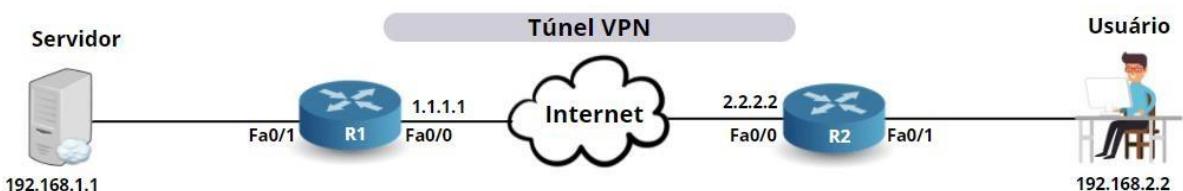
No exemplo acima, utilizamos dois roteadores diferentes, mas poderíamos usar firewalls como o ASA, Firepower, Fortigate Palo Alto etc., inclusive, firewalls costumam ser mais usados do que roteadores para formação de VPNs.

Outra vantagem dos túneis (VPN) é que eles permitem que as LANs com endereços IPs privados comuniquem entre si.

Client-to-site VPN

Embora não esteja no escopo da prova, vale a pena darmos uma pequena introdução sobre o que é VPN cliente para site.

Client-to-site VPN também é conhecida como VPN de usuário remoto. O usuário instala um cliente VPN em seu computador, laptop, smartphone ou tablet. O túnel VPN é estabelecido entre o dispositivo do usuário e o dispositivo de rede remoto. Eis um exemplo:



Na imagem acima, o usuário estabeleceu um túnel VPN entre seu computador e roteador R1. Isso permite que o usuário accesse o servidor atrás de R1 de forma remota.

Protocolos VPN

Não entraremos na configuração de VPNs, porém é interessante saber os protocolos utilizados. Os mais comuns são:

- **IPSec**
- **PPTP**
- **L2TP**
- **VPN SSL**

Vamos a um apanhado geral de cada um desses protocolos.

IPSec

O protocolo IPv4 em si não possui nenhum recurso de segurança, razão pela qual o IPSec foi criado. O IPSec não é um protocolo, mas é uma estrutura que oferece recursos de confidencialidade, integridade, autenticação e anti-replay. Tal como o protocolo IP, ele atua na camada três do modelo OSI.

Ele usa uma variedade de protocolos, e a vantagem dessa estrutura é que os protocolos que ela utiliza podem mudar no futuro. Por exemplo, atualmente, podemos usar algoritmos de criptografia como DES, 3DES ou AES, mas se um novo algoritmo for criado, o IPSec poderá usá-lo.

É possível usar IPSec para:

- Criação de túnel VPN site a site.
- Criação de túnel VPN cliente-para-site (usuário remoto).
- Entre dois servidores para autenticar e/ou criptografar o tráfego.

PPTP

PPTP (Point to Point Tunneling Protocol) é um dos protocolos VPN mais antigos, foi lançado por volta de 1995. Ele usa um túnel GRE para encapsulamento e PPP para autenticação (usando MS-Chap ou MS-Chap v2). A criptografia é feita com o protocolo MPPE.

Como já existe há algum tempo, o PPTP é compatível com muitos clientes e sistemas operacionais. O PPTP, entretanto, provou ser inseguro, então é totalmente desaconselhável o uso desse protocolo.

L2TP

L2TP (Layer Two Tunneling Protocol - Protocolo de encapsulamento de camada dois) é uma extensão do PPTP e, como o nome indica, nos permite encapsular o tráfego da camada dois sobre as conexões da camada três. O L2TP pode ser usado se para “conectar” duas LANs remotas em uma única sub-rede em ambos os sites. O L2TP em si não oferece criptografia, e é por isso que ele costuma ser usado junto com o IPSec. Quando utilizamos L2TP e IPSec juntos, geralmente chamamos de L2TP/IPSec

VPN SSL

SSL (Secure Sockets Layer) é o protocolo normalmente utilizado para criptografar o tráfego entre um navegador e um servidor web. Quando você navega na Internet usando HTTP, tudo é transmitido em texto claro. Para conexões seguras, tem-se que usar HTTPS.

Apesar de se chamar SSL VPN, hoje usamos TLS (Transport Layer Security) para HTTPS, que é o sucessor do SSL.

Uma das vantagens do SSL VPN é que ele utiliza HTTPS, e por isso é possível usá-lo em praticamente qualquer lugar. A maioria dos pontos de acesso Wi-Fi públicos permite o tráfego HTTPS, enquanto alguns podem bloquear outros tipos de tráfego, como o IPSec. Outra razão para a popularidade do SSL VPN é que, com ele, não há necessidade de usarmos um cliente de software.

A maioria das soluções SSL VPN oferece um “site” e através dele podemos acessar as aplicações. Para alguns recursos avançados, pode ser necessário a instalação de um software.

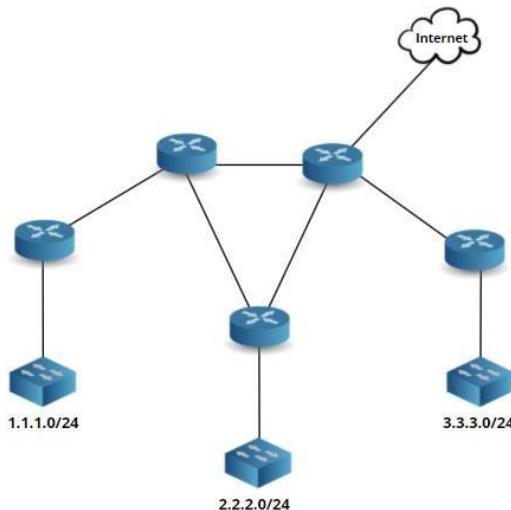
5.6 Configure and verify access control lists

Veremos agora como configuramos e verificamos as famosas listas de acesso (Access-list), também veremos a diferença entre listas de acesso standard (padrão) e extended (estendidas).

As listas de acesso funcionam tanto na camada de rede (layer 3) como na camada de transporte (layer 4), e são usadas para dois objetivos específicos:

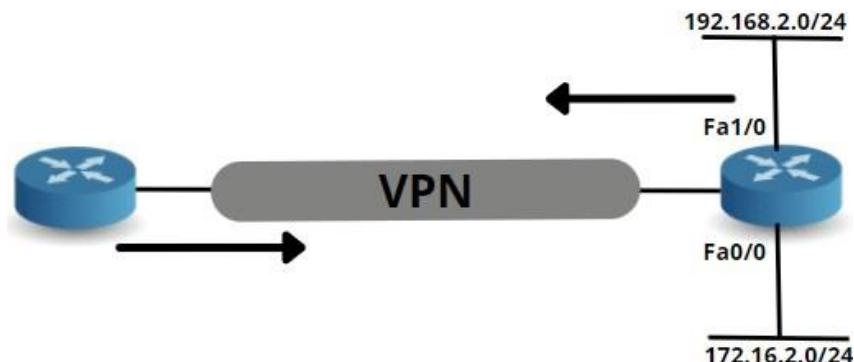
- Filtering - Filtros
- Classification – Classificação

Observe a topologia abaixo:



Filtragem (filtering): A filtragem é usada para permitir ou impedir que determinado tráfego chegue a certas partes da rede. Caso não haja o processo de filtragem, o tráfego pode ir a qualquer lugar. Observe a topologia acima, não é recomendável, por questões de segurança, que os pacotes IP vindos da Internet entrem livremente na Lan. Além de bloquear o tráfego advindo da Internet, também é possível utilizar uma lista de acesso para bloquear pacotes IP que tem origem dentro da nossa própria rede, por exemplo, podemos bloquear que pacotes vindos da rede de 3.3.3.0/24 cheguem até a rede 1.1.1.0/24.

Classification (Classificação): A classificação não descarta pacotes IP como acontece na filtragem, ela é usada para “selecionar” o tráfego. Observe a topologia abaixo:

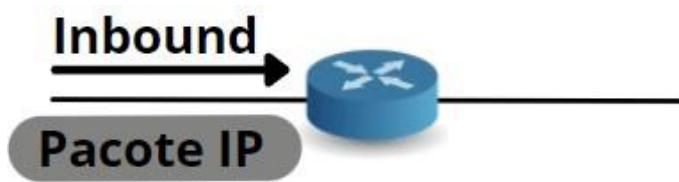


Na imagem acima, temos uma VPN criptografando o tráfego entre os dois roteadores. Sempre que criamos uma VPN, podemos usar uma lista de acesso para “selecionar” o tráfego que deve ser criptografado. Por exemplo, o tráfego vindo da rede 192.168.2.0/24 deve ser criptografado, mas o tráfego da rede 172.16.2.0/24 não. Podemos usar uma lista de acesso para fazer essa ‘seleção’, isso é chamado de classificação.

Depois de criar uma lista de acesso, existem 3 locais onde pode aplicá-la:

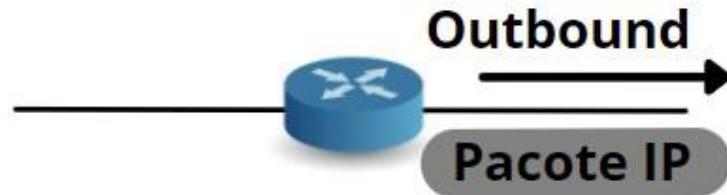
1. Inbound – Entrada:

Podemos coloca-la no sentido de **entrada (inbound)** da interface, o que significa que todos os pacotes que chegarem ao roteador atingirão a lista de acesso, antes de ‘entrar’ no roteador.

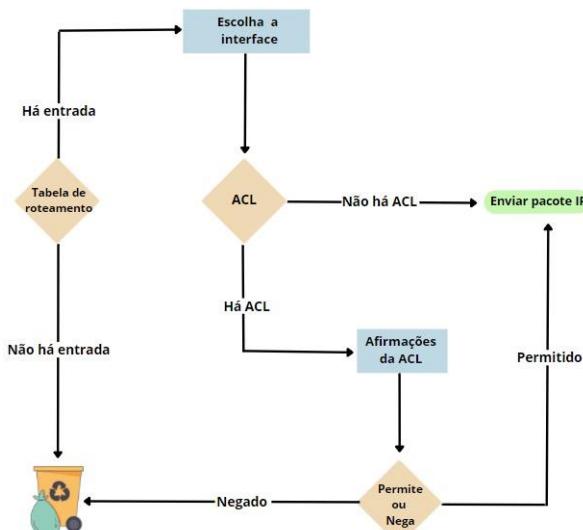


2. Outbound – Saída:

Outra opção é colocar a lista de acesso no sentido de **saída (outbound)**. Nesse caso, os pacotes IP passarão pelo roteador e, quando saírem pela interface é que serão verificados na lista de acesso.



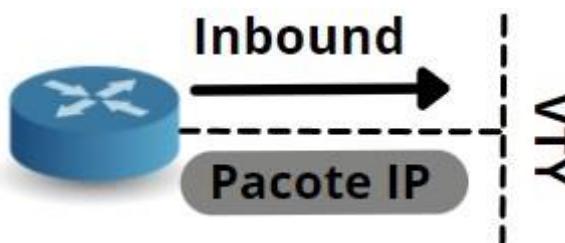
Quando uma lista de acesso de saída (access-list outbound) é aplicada no roteador, eis o que acontece:



- 1- Os pacotes IP entram no roteador.
- 2- O roteador verificará se conhece o destino consultando a tabela de roteamento.
- 3- Se não houver nenhuma entrada correspondente na tabela de roteamento, o pacote IP será descartado.
- 4- Se houver uma entrada correspondente na tabela de roteamento, ela selecionará a interface de saída.
- 5- Se não houver lista de acesso, o pacote IP será enviado pela interface escolhida.
- 6- Se houver uma lista de acesso, o roteador verifica o pacote IP e o compara com a lista de acesso.
- 7- Se a lista de acesso permitir o pacote IP, ele será encaminhado, caso contrário, será descartado.

3. VTY

A terceira opção é aplicar a access-list à linha VTY. Com isso, protegemos o tráfego telnet ou SSH de tentativas de conexão a partir de endereços IP não permitidos.



Chegou a hora de vermos como é configurada uma access list:

```
Router#show access-lists
```

```

Standard IP access-list 1

 10 permit 192.168.1.0, wildcard bits 0.0.0.255
 20 permit 192.168.2.0, wildcard bits 0.0.0.255
 30 permit 172.16.0.0, wildcard bits 0.0.255.255

```

Listas de acesso funcionam usando **instruções (statements)**. Na saída acima, você pode ver que a lista de acesso número 1 tem 3 declarações (statements), número 10, 20 e 30. Sempre que um pacote IP chegar até essa lista de acesso, os seguintes passos serão tomados:

1. As listas de acesso são processadas de cima para baixo, portanto, primeiro ela verificará se o pacote corresponde à instrução 10.
2. Se não corresponder à instrução 10, verificaremos se corresponde à instrução 20.
3. Se não corresponder à instrução 20, verificaremos se corresponde à instrução 30.
4. Se não corresponder à instrução 30, o pacote será descartado.

Se um pacote **corresponder (match)** com uma determinada declaração, haverá uma ação imediata. O pacote será **permitido** (será encaminhando) ou **negado** (será descartado). Por exemplo, se tivermos um pacote que corresponda à instrução 10, o roteador **não verificará** se "também" corresponde à instrução 20.

No final de cada lista de acesso há um **deny any (negar tudo)**, o que significa que se você não permitir algo **explicitamente**, ele será descartado. Você **não vê** esse negar, mas está lá!

Existem dois tipos de listas de acesso:

- **Standard** access-lists (Listas de acesso padrão)
- **Extended** access-lists (Listas de acesso estendidas)

Vamos começar com a standard access-list:



A standard access-list é muito básica, pois com ela só é possível verificar os endereços IP de origem, não é possível fazer nada mais específico que isso.

A extended access-list oferece muito mais opções. Com ela é possível verificar os endereços IP de origem e destino, e também combinar informações da camada de transporte (camada 4), como números de porta TCP ou UDP.



Isso não significa que as standard access-list são ruins, já que às vezes os endereços IP de origem são tudo o que precisamos nos preocupar. Por exemplo, se quisermos uma lista de acesso para selecionar quais redes devem ser traduzidas pelo NAT, uma lista de acesso padrão servirá perfeitamente.

A tabela abaixo, mostra como podemos reconhecer e diferenciar uma lista de acesso padrão e estendida:

Tipo de ACL	Número - Identificação
Padrão	1 -99 ou 1300 – 1999
Estendido	100 – 199 ou 2000 – 2699
Nomeado	Pode-se escolher o nome que desejar

Para criarmos uma lista de acesso padrão, é preciso escolher um número entre 1-99 ou 1300-1999. Para a lista de acesso estendida, escolhemos um número entre 100-199 ou 2000-2699. Também é possível usar listas de acesso nomeadas escolhendo um nome, esse recurso pode ser usado tanto para listas de acesso padrão como estendidas.

Antes de partirmos para a parte prática, onde tudo fará mais sentido, faz-se necessário passar mais algumas diretrizes para configuração de access-list:

- Access-lists são criadas globalmente e só depois são atribuídas a uma interface.
- Só é possível ter uma única ACL por direção, portanto, é impossível ter 2 listas de acesso de entrada.
- As boas práticas recomendam colocar as instruções mais específicas no topo lista de acesso, pois quando um pacote corresponde a uma instrução, o roteador não verifica se ela corresponde a outras instruções.
- Não se esqueça que a última afirmação é negar tudo. Como disse anteriormente, você não vê essa afirmação no roteador, mas está lá.

Wildcard Mask

Quando trabalhamos com listas de acesso, trabalhamos também com um conceito de máscara diferente do que vimos até aqui, trabalhamos com a máscara coringa. Observe o exemplo da lista de acesso abaixo:

```
Router#show access-lists
Standard IP access-list 1
    10 permit 192.168.1.0, wildcard bits 0.0.0.255
    20 permit 192.168.2.0, wildcard bits 0.0.0.255
    30 permit 172.16.0.0, wildcard bits 0.0.255.255
```

As listas de acesso não usam máscaras de sub-rede, mas bits coringa. Isso significa que no conceito binário, o número “0” será substituído pelo número “1” e vice-versa.

Observe alguns exemplos:

A máscara de sub-rede 255.255.255.0 seria 0.0.0.255 com a máscara coringa. Para essa explicação ficar mais compreensível, precisamos adentrar nos números binários:

Bits	128	64	32	16	8	4	2	1
255	1	1	1	1	1	1	1	1

Este é o primeiro octeto da máscara de sub-rede (255.255.255.0) em binário, como você pode ver todos os valores possuem o número 1, o que significa que o número decimal é 255.

Bits	128	64	32	16	8	4	2	1
0	0	0	0	0	0	0	0	0

Este também é o primeiro octeto, mas agora com bits coringa. Para chegarmos ao equivalente coringa, precisamos inverter os bits: Quando houver o número ‘1’, colocamos o número ‘0’ e vice versa. É por isso que agora temos o número decimal ‘0’.

Vamos fazer o exemplo com outra máscara de sub-rede, vamos pegar a máscara 255.255.255.128. Conhecemos a parte 255.255.255.X, então, nos resta calcular apenas a parte .128.

Bits	128	64	32	16	8	4	2	1
128	1	0	0	0	0	0	0	0

Vamos apenas inverter os bits:

Bits	128	64	32	16	8	4	2	1
255	0	1	1	1	1	1	1	1

E agora vamos transformar em números decimais: $64 + 32 + 16 + 8 + 4 + 2 + 1$, que no caso em tela, nos dá o número decimal 127.

A máscara de sub-rede 255.255.255.128 transformada em máscara coringa será **0.0.0.127**.

Vamos para mais um exemplo. Pegaremos a máscara de sub-rede 255.255.255.224:

Bits	128	64	32	16	8	4	2	1
224	1	1	0	0	0	0	0	0

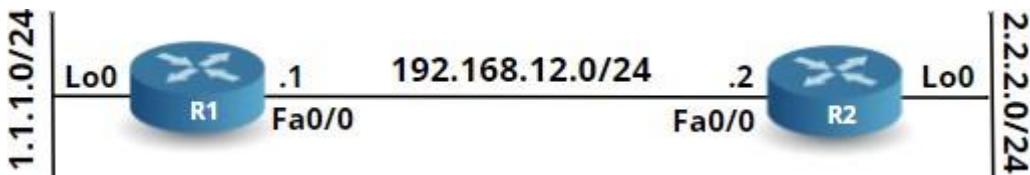
Vamos invertos os bits:

Bits	128	64	32	16	8	4	2	1
31	0	0	0	1	1	1	1	1

Logo, a máscara coringa é $16 + 8 + 4 + 2 + 1 = 31$ ou 0.0.0.31

Standard access-list

Depois de muita teoria, é hora de configurarmos uma access-list em um roteador Cisco. Usaremos a topologia abaixo:



Temos dois roteadores, cada roteador possui uma interface loopback. Usaremos duas rotas estáticas para que os roteadores possam alcançar a interface loopback atrás do outro roteador:

```
R1(config)#ip route 2.2.2.0 255.255.255.0 192.168.12.2
```

```
R2(config)#ip route 1.1.1.0 255.255.255.0 192.168.12.1
```

Agora vamos começar a configuração da standard access-list! Criaremos uma ACL no R2 que permita apenas o tráfego da rede 192.168.12.0/24:

```
R2(config)#access-list 1 permit 192.168.12.0 0.0.0.255
```

Esta única permissão de entrada é suficiente. Tenha em mente que em todo final de uma lista de acesso tem um “negar qualquer coisa (deny any)”. Não o vemos, mas está lá. Vamos aplicar esta lista de acesso no sentido de entrada (inbound) no R2:

```
R2(config)#interface fastEthernet 0/0
R2(config-if)#ip access-group 1 in
```

Nós aplicamos a access-list através do comando **ip access-group** na interface escolhida. Como aplicamos no sentido inbound usamos a palavra-chave **in**.

Podemos verificar se a lista de acesso foi aplicada com sucesso usando o comando **show ip interface**. Observe que a access-list 1 foi aplicada no sentido inboud.

```
R2#show ip interface fastEthernet 0/0
FastEthernet0/0 is up, line protocol is up
```

```
Internet address is 192.168.12.2/24
Broadcast address is 255.255.255.255
Address determined by setup command
MTU is 1500 bytes
Helper address is not set
Directed broadcast forwarding is disabled
Outgoing access-list is not set
Inbound access-list is 1
```

Vamos realizar alguns testes com ping para validar o funcionamento da nossa access-list:

```
R1#ping 192.168.12.2
Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.12.2, timeout is 2 seconds:
!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms
```

Conforme esperado, o ping foi bem-sucedido; vamos verificar a lista de acesso:

```
R2#show access-lists
Standard IP access-list 1
    10 permit 192.168.12.0, wildcard bits 0.0.0.255 (15 matches)
```

Esse comando é muito interessante. Observe que podemos ver o número de matchs (correspondência). No caso acima tivemos 15.

Vamos tentar um ping com origem diferente da que temos permitido na access-list:

```
R1#ping 192.168.12.2 source loopback 0
Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.12.2, timeout is 2 seconds:
Packet sent with a source address of 1.1.1.1
U.U.U

Success rate is 0 percent (0/5)
```

Ao enviar um ping, você pode usar a palavra-chave **source** para selecionar a interface de origem. O endereço IP de origem deste pacote IP agora é 1.1.1.1, e conforme esperado, o ping não obteve resposta, afinal, não há correspondência dessa rede na nossa access-list, o que significa que o tráfego será descartado.

```
R2#show access-lists
Standard IP access-list 1
    10 permit 192.168.12.0, wildcard bits 0.0.0.255 (15 matches)
```

Não veremos com o comando **show access-list**, porque o “deny any”, que é implícito em todas as access-list, está impedindo (repetirei ‘ad nauseam’ porque isso será cobrado).

Vamos para um novo exemplo, agora queremos negar o tráfego da rede 192.168.12.0/24, mas permitir todas as outras redes. A configuração será dessa forma:

```
R2(config)#access-list 2 deny 192.168.12.0 0.0.0.255  
R2(config)#access-list 2 permit any
```

Vamos criar uma nova access-list cuja primeira instrução negará a rede 192.168.12.0/24. A segunda instrução será uma permissão qualquer. Por causa dessa permissão, nada atingirá o invisível "deny any", com exceção de 192.168.12.0/24 que dará ‘match’ na primeira linha. Vamos aplicar a nova lista de acesso, mas antes vamos apagar a access-list que fizemos anteriormente:

```
R2(config-if)#no ip access-group 1 in
```

E agora, aplicar a nova access-list:

```
R2(config-if)#ip access-group 2 in
```

Hora de testarmos:

```
R1#ping 2.2.2.2  
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:  
U.U.U  
Success rate is 0 percent (0/5)
```

```
R2#show access-lists 2  
Standard IP access-list 2  
10 deny    192.168.12.0, wildcard bits 0.0.0.255 (11 matches)  
20 permit any
```

Observe que os pings deram ‘macth’ logo na primeira instrução da lista e foram descartados....

```
R1#ping 2.2.2.2 source loopback 0  
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 2.2.2.2, timeout is 2 seconds:  
Packet sent with a source address of 1.1.1.1  
!!!!  
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/4/4 ms
```

```
R2#show access-lists 2  
Standard IP access-list 2
```

```

10 deny 192.168.12.0, wildcard bits 0.0.0.255 (11 matches)

20 permit any (15 matches)

```

Os pings vindo da interface loopback0 de R1 passam pela primeira instrução e encontram correspondência na segunda instrução, só então são permitidos.

Preste bastante atenção nessa parte: Para remover uma declaração de uma access-list, não faça da maneira abaixo:

```
R2(config)#no access-list 2 deny 192.168.12.0 0.0.0.255
```

Apesar de parecer que removemos apenas essa linha, nós acabamos de remover toda a access-list:

```
R2#show access-lists 2
```

Observe, toda a lista se foi! Não usamos o comando **no access-list** para remover uma linha da access-list. O roteador interpreta o comando “**no access-list 2**” como instrução para remover toda a access-list. Divertido de descobrir em laboratório, mas não em um ambiente de produção. Mais à frente aprenderemos como manipular essas linhas.

Além de aplicar uma lista de acesso em interfaces, também podemos aplicá-la às linhas VTY. Isso é útil para proteger o acesso telnet ou SSH do roteador. Vamos configurar R1 para que o acesso telnet seja permitido apenas da rede 192.168.12.0/24:

```

R1(config)#access-list 3 permit 192.168.12.0 0.0.0.255
R1(config)#line vty 0 4
R1(config-line)#access-class 3 in

```

Observe que criamos a access-list 3, mas usamos o comando ‘**access-class**’ nas linhas VTY. Nas interfaces usamos o comando “**access-group**”, mas nas linhas VTY usamos “**access-class**” para aplicá-los.

Vamos tentar conectar no roteador através do telnet:

```

R2#telnet 192.168.12.1
Trying 192.168.12.1 ... Open
Password required, but none set
[Connection to 192.168.12.1 closed by foreign host]

```

O “open” significa que há conexão naquela determinada porta. A conexão foi encerrada porque não configuramos uma senha para o telnet, mas a access-list está funcionando:

```

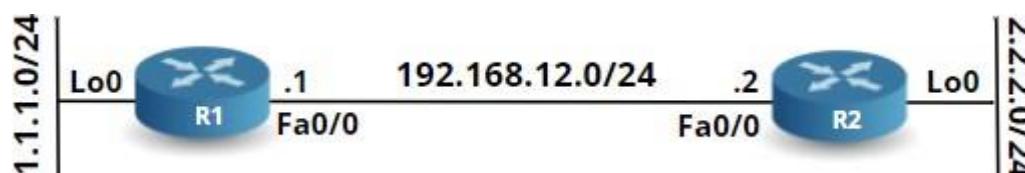
R1#show access-lists
Standard IP access-list 3
10 permit 192.168.12.0, wildcard bits 0.0.0.255 (2 matches)

```

Observe que dois pacotes deram ‘match’ na instrução contida na lista de acesso 3.

Extended Access-List

Agora daremos uma olhada na Extended Access-List. Usaremos a mesma topologia da configuração anterior:



Usando a Extended Access-List, é possível criar instruções muito mais complexas. Faremos um exemplo onde precisaremos atender os seguintes requisitos:

- O tráfego da rede 1.1.1.0/24 tem permissão para se conectar ao servidor HTTP no R2, mas pode conectar apenas ao endereço IP 2.2.2.2.
- Todo o outro tráfego deve ser negado.

Agora precisamos traduzir isso para uma instrução de Extended Access-List. Basicamente, a sintaxe seria parecida com essa:

[source] + [source port] to [destination] + [destination port] em português claro: **[Endereço de origem]** + **[porta de origem]** para **[endereço de destino]** + **[porta de destino]**

Vamos para configuração:

```
R2(config)#access-list 100 ?  
    deny      Specify packets to reject  
    dynamic   Specify a DYNAMIC list of PERMITs or DENYs  
    permit     Specify packets to forward  
    remark    Access-list entry comment
```

Em primeiro lugar, precisamos selecionar se vamos permitir ou negar. A propósito, também podemos usar um ‘remark’. Com ele é possível adicionar um comentário às instruções da lista de acesso. Vamos selecionar ‘permit’ (permitir):

```
R2(config)#access-list 100 permit ?  
    <0-255>  An IP protocol number  
    ahp       Authentication Header Protocol  
    eigrp    Cisco's EIGRP routing protocol  
    esp      Encapsulation Security Payload  
    gre      Cisco's GRE tunneling  
    icmp    Internet Control Message Protocol  
    igmp    Internet Gateway Message Protocol  
    ip      Any Internet Protocol  
    ipinip  IP in IP tunneling  
    nos     KA9Q NOS compatible IP over IP tunneling  
    ospf    OSPF routing protocol  
    pcp     Payload Compression Protocol  
    pim     Protocol Independent Multicast  
    tcp     Transmission Control Protocol  
    udp     User Datagram Protocol
```

Observe a quantidade de opções que temos. Como queremos permitir o tráfego HTTP, vamos selecionar o protocolo TCP:

```
R2(config)#access-list 100 permit tcp ?  
A.B.C.D  Source address  
any      Any source host  
host     A single source host
```

Agora temos que selecionar uma origem (source). Podemos digitar um endereço de rede com um wildcard mask ou utilizar a palavra-chave **any** ou **host**. Essas duas palavras-chave são “atalhos”. Deixe-me explicar:

Se você digitar “0.0.0.0 255.255.255.255”, terá todas as redes. Em vez de digitar isso, podemos usar a palavra-chave **any**.

Se você digitar um endereço como “2.2.2.2 0.0.0.0”, estará selecionado um único endereço IP. Em vez de digitar a wildcard mask “0.0.0.0”, podemos usar a palavra-chave **host**.

Queremos selecionar a rede 1.1.1.0/24 como source:

```
R2(config)#access-list 100 permit tcp 1.1.1.0 0.0.0.255 ?  
A.B.C.D  Destination address  
any      Any destination host  
eq       Match only packets on a given port number  
gt       Match only packets with a greater port number  
host    A single destination host  
lt       Match only packets with a lower port number  
neq     Match only packets not on a given port number  
range   Match only packets in the range of port numbers
```

Além de selecionar a ‘source’, também podemos selecionar o número da porta de origem. Lembre-se que, quando nos conectarmos a partir do R1 ao servidor HTTP no R2, o número de porta de origem será aleatório, portanto, não é possível especificar um número de porta de origem.

```
R2(config)#access-list 100 permit tcp 1.1.1.0 0.0.0.255 host 2.2.2.2 ?  
ack      Match on the ACK bit  
dscp     Match packets with given dscp value  
eq       Match only packets on a given port number  
established Match established connections  
fin      Match on the FIN bit  
fragments Check non-initial fragments  
gt       Match only packets with a greater port number  
log     Log matches against this entry  
log-input Log matches against this entry, including input interface  
lt       Match only packets with a lower port number
```

neq	Match only packets not on a given port number
precedence	Match packets with given precedence value
psh	Match on the PSH bit
range	Match only packets in the range of port numbers
rst	Match on the RST bit
syn	Match on the SYN bit
time-range	Specify a time-range
tos	Match packets with given TOS value
urg	Match on the URG bit
<cr>	

Vamos selecionar o destino, que é o endereço IP 2.2.2.2. Eu poderia ter digitado “2.2.2.2 0.0.0.0”, porém, é mais prático usar a palavra-chave **host**. Além do endereço IP de destino, podemos selecionar o número da porta de destino com a palavra-chave **eq**:

```
R2(config)#access-list 100 permit tcp 1.1.1.0 0.0.0.255 host 2.2.2.2 eq 80
```

Este será o resultado. Antes de aplicá-lo à interface, adicionaremos uma instrução extra que será bem útil:

```
R2(config)#access-list 100 deny ip any any log
```

Utilizando a afirmação acima, tornamos visível o “deny any” implícito que vem no final de cada access-list. A palavra-chave **log** mostrará no console todos os pacotes que forem negados. Vamos testar!

```
R2(config)#interface fastEthernet 0/0
R2(config-if)#ip access-group 100 in
```

Vamos aplicar a access-list na interface fa0/0 sentido de entrada. Porém, precisamos habilitar o servidor HTTP:

```
R2(config)#ip http server
```

Vamos tentar conectar a porta 80:

```
R1#telnet 2.2.2.2 80
Trying 2.2.2.2, 80 ...
% Destination unreachable; gateway or host down
```

Não é preciso um navegador web para testar se o servidor HTTP está funcionando. Podemos usar o telnet para nos conectarmos à porta TCP 80. O tráfego acima será negado, como podemos ver no console do roteador R2:

```
R2# %SEC-6-IPACCESSLOGP: list 100 denied tcp 192.168.12.1(55419) -> 2.2.2.2(80),
1 packet
```

Podemos verificar as correspondências na access-list:

```
R2#show access-lists
Extended IP access-list 100
10 permit tcp 1.1.1.0 0.0.0.255 host 2.2.2.2 eq www
```

```
20 deny ip any any log (1 match)
```

O pacote foi negado porque o endereço IP de origem era 192.168.12.1 e não foi permitido. Agora, vamos nos conectar a partir do endereço IP 1.1.1.1:

```
R1#telnet 2.2.2.2 80 /source-interface loopback 0
```

```
Trying 2.2.2.2, 80 ... Open
```

Observe que agora o status diz ‘Open’, o que significa que está conectado. Quando usamos telnet, podemos selecionar a interface de origem. O pacote agora é permitido porque corresponde à primeira instrução da lista de acesso.

Agora chegou a hora de resolvemos o grande mistério: Remover uma única instrução da access-list. Temos duas opções:

1. Copiar a access-list para o bloco de notas, editá-la e depois colar de volta no roteador.
2. Usar o editor de access-list.

O editor de lista de acesso parece mais fácil, certo? É assim que funciona:

```
R2(config)#ip access-list extended 100
```

Use o comando **ip access-list** para criar uma nova lista de acesso ou modificar as atuais. O console é semelhante a este:

```
R2(config-ext-nacl)#
```

Agora podemos adicionar ou remover instruções:

```
R2(config-ext-nacl)#?  
Ext Access-list configuration commands:  
<1-2147483647> Sequence Number  
default      Set a command to its defaults  
deny        Specify packets to reject  
dynamic      Specify a DYNAMIC list of PERMITs or DENYS  
evaluate     Evaluate an access-list  
exit        Exit from access-list configuration mode  
no          Negate a command or set its defaults  
permit      Specify packets to forward  
remark      Access-list entry comment
```

Vamos remover a instrução 20 da access-list 100:

```
R2(config-ext-nacl)#do show access-list 100  
Extended IP access-list 100  
 10 permit tcp 1.1.1.0 0.0.0.255 host 2.2.2.2 eq www (21 matches)  
 20 deny ip any any log (1 match)
```

Agora, basta um simples comando:

```
R2(config-ext-nacl)#no 20
```

Basta colocar o comando ‘**no**’ antes do número de sequência e ela desaparecerá:

```
R2(config-ext-nacl)#do show access-list 100  
Extended IP access-list 100  
10 permit tcp 1.1.1.0 0.0.0.255 host 2.2.2.2 eq www (21 matches)
```

Para finalizar, é importante saber que podemos criar uma named access-list (lista de acesso nomeada). Para testar, criaremos uma lista que negue o tráfego ICMP do R2 para a interface loopback0 do R1 e que permita todo o resto:

```
R1(config)#ip access-list extended NEGAR_ICMP  
R1(config-ext-nacl)#deny icmp host 192.168.12.2 1.1.1.0 0.0.0.255  
R1(config-ext-nacl)#deny icmp host 2.2.2.2 1.1.1.0 0.0.0.255  
R1(config-ext-nacl)#permit ip any any  
R1(config-ext-nacl)#exit
```

Está montada nossa named access-list, colocamos o nome dela de “NEGAR_ICMP”. A primeira instrução diz que o tráfego ICMP do endereço IP 192.168.12.2 deve ser negado. A segunda linha nega o tráfego ICMP tendo como origem o endereço IP 2.2.2.2. Qualquer outro tráfego será permitido. Faremos o seguinte para aplica-lo à interface:

```
R1(config)#interface fastEthernet 0/0  
R1(config-if)#ip access-group NEGAR_ICMP in
```

Agora vamos testar:

```
R2#ping 1.1.1.1  
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:  
.....  
Success rate is 0 percent (0/5)
```

```
R1#show access-lists  
Extended IP access-list NEGAR_ICMP  
10 deny icmp host 192.168.12.2 1.1.1.0 0.0.0.255 (15 matches)  
20 deny icmp host 2.2.2.2 1.1.1.0 0.0.0.255  
30 permit ip any any
```

Como esperado, o primeiro ping não teve resposta.

```
R2#ping 1.1.1.1 source loopback 0  
Type escape sequence to abort.  
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
```

```

Packet sent with a source address of 2.2.2.2

.....  

Success rate is 0 percent (0/5)

R1#show access-lists  

Extended IP access-list NEGAR_ICMP  

    10 deny icmp host 192.168.12.2 1.1.1.0 0.0.0.255 (15 matches)  

    20 deny icmp host 2.2.2.2 1.1.1.0 0.0.0.255 (15 matches)  

    30 permit ip any any

```

O segundo ping também não teve resposta.

Vamos tentar um telnet para ver se temos resultado:

```

R2#telnet 1.1.1.1  

Trying 1.1.1.1 ...

```

```

R1#show access-lists  

Extended IP access-list NEGAR_ICMP  

    10 deny icmp host 192.168.12.2 1.1.1.0 0.0.0.255 (27 matches)  

    20 deny icmp host 2.2.2.2 1.1.1.0 0.0.0.255 (18 matches)  

    30 permit ip any any (12 matches)

```

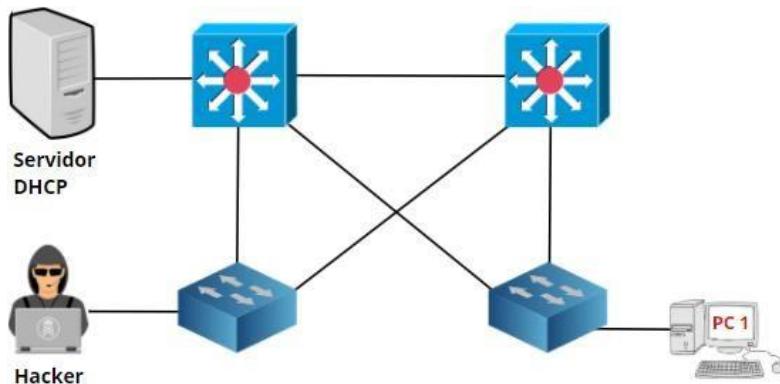
Não configuramos o telnet no R1 e por isso a conexão não completou embora pacotes estejam passando pela lista de acesso. O telnet não tem correspondência nas duas primeiras linhas e será liberado quando chegar na linha 30, que possui um ‘permit any any’.

5.7 Configure Layer 2 security features (DHCP snooping, dynamic ARP inspection, and port security)

Hora de configurarmos os recursos de segurança da camada 02. Começaremos com DHCP Snooping.

DHCP snooping

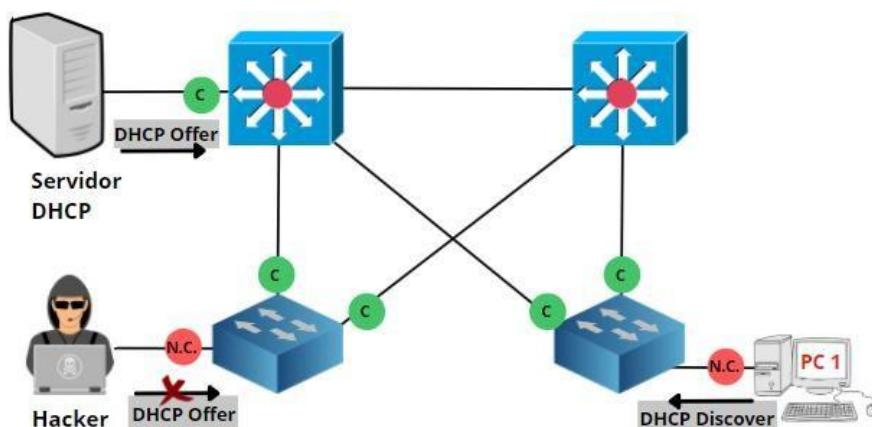
DHCP Snooping é uma ferramenta que configuramos em switches para monitorar o tráfego DHCP e interromper o tráfego de quaisquer pacotes DHCP que não estejam previamente autorizados. Essa parte é melhor explicada através de exemplos, portanto, dê uma olhada na imagem abaixo:



Na imagem acima, temos um servidor DHCP conectado a um switch. No canto inferior direito, temos um cliente legítimo que gostaria de obter um endereço IP. Porém, do outro lado, temos um agente malicioso executando um software que distribui endereços IPs como se fosse um servidor DHCP legítimo. Em casos assim, quem você acha que responderá primeiro à mensagem DHCP Discover? O servidor DHCP legítimo ou o computador com um software malicioso simulando ser um servidor DHCP?

Em redes maiores, provavelmente encontraremos um servidor DHCP central em algum lugar no CPD. Se um invasor executar um servidor DHCP na mesma sub-rede, ele provavelmente responderá mais rápido a mensagem de DHCP Discover do cliente. Com o controle do endereçamento IP, o agente malicioso pode atribuir ao cliente seu próprio endereço IP como o gateway padrão, possibilitando o ataque man-in-the-middle. Outra opção seria enviar seu próprio endereço IP como servidor DNS, dessa forma ele conseguirá falsificar sites, etc.

O invasor também pode enviar mensagens DHCP Discover para o servidor DHCP, e assim, tentar esgotar o pool de endereços IP. Dessa forma, um computador legítimo, não conseguirá obter endereços IPs. O que podemos fazer para deter esse tipo de ataque? Podemos configurar nossos switches para que rastreiem mensagens DHCP discover e DHCP offer. Veja como:

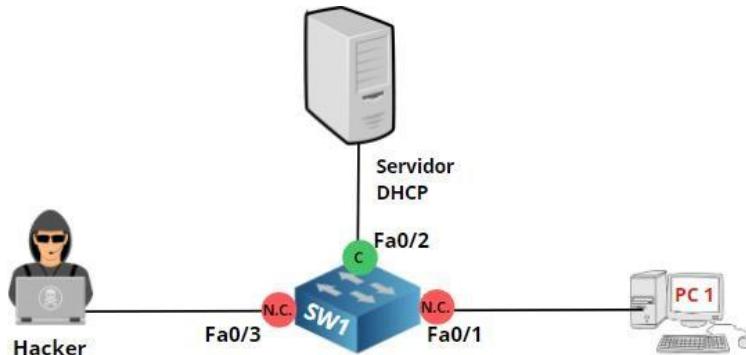


As interfaces que se conectam aos clientes não possuem permissão para enviar mensagens DHCP offer. Podemos garantir tornando-as **não-confiáveis**. Uma interface não-confiável bloqueará as mensagens DHCP offer. Apenas uma interface configurada como **confiável** possui permissão para encaminhar mensagens DHCP offer. Também podemos limitar através do 'rate-limit' a quantidade de mensagens DHCP discover de uma determinada interface. Por padrão, a quantidade permitida de 'DHCP discover' que uma interface pode enviar é ilimitada. Dessa forma, impedimos que ataques esgotem o pool de DHCP.

Vejamos como configurar o DHCP snooping:

DHCP Snooping configuração

Vamos usar a seguinte topologia:



A interface fa0/1 está conectada a um cliente que gostaria de obter um endereço IP através do servidor DHCP conectado a interface fa0/2. Há um invasor conectado a fa0/3 que está executando um software malicioso que distribui endereços DHCP. Vejamos como pará-lo.

Primeiro, é necessário habilitar DHCP snooping globalmente.

```
SW1(config)#ip dhcp snooping
```

Por padrão, o switch adicionará a opção 82 à mensagem DHCP discover antes de encaminhá-la para o servidor DHCP. Alguns servidores DHCP não gostam disso e descartarão o pacote. Se o seu cliente não obtiver mais nenhum endereço IP após habilitar o DHCP snooping globalmente, você deve usar este comando.

```
SW1(config)#no ip dhcp snooping information option
```

Selecione as VLANs para as quais deseja usar DHCP snooping.

```
SW1(config)#ip dhcp snooping vlan 1
```

Depois de habilitar o DHCP snooping, por padrão, nenhuma interface será confiável. É preciso certificar que a interface que o servidor DHCP está plugado seja marcada como confiável.

```
SW1(config)#interface fa0/2
SW1(config-if)#ip dhcp snooping trust
```

Opcionalmente, podemos limitar a taxa de pacotes DHCP que a interface pode receber. Vamos configurar a interface fa0/1 para que não receba mais de 10 pacotes DHCP por segundo.

```
SW1(config)#interface fa0/1
SW1(config-if)#ip dhcp snooping limit rate 10
```

Use o comando ‘show ip dhcp snooping’ para verificar a configuração:

```
SW1#show ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on following VLANs:
1
DHCP snooping is operational on following VLANs:
1
DHCP snooping is configured on the following L3 Interfaces:
```

```

Insertion of option 82 is enabled

circuit-id format: vlan-mod-port

remote-id format: MAC

Option 82 on untrusted port is not allowed

Verification of hwaddr field is enabled

Verification of giaddr field is enabled

DHCP snooping trust/rate is configured on the following Interfaces:

Interface          Trusted      Rate limit (pps)
-----
FastEthernet0/1    no          10
FastEthernet0/2    yes         unlimited

```

```

SW1#show ip dhcp snooping binding

MacAddress      Ip Address      Lease(sec)  Type      VLAN  Interface
-----
00:0C:29:28:5C:6C  192.168.1.1  85655      dhcp-snooping  1     FastEthernet0/1

```

Assim que o cliente receber um endereço IP do servidor DHCP legítimo, você verá que o SW1 rastreará a ligação MAC para IP. As mensagens ‘DHCP offer’ do servidor DHCP vindas de interfaces não-confiáveis serão descartadas.

Dynamic ARP inspection

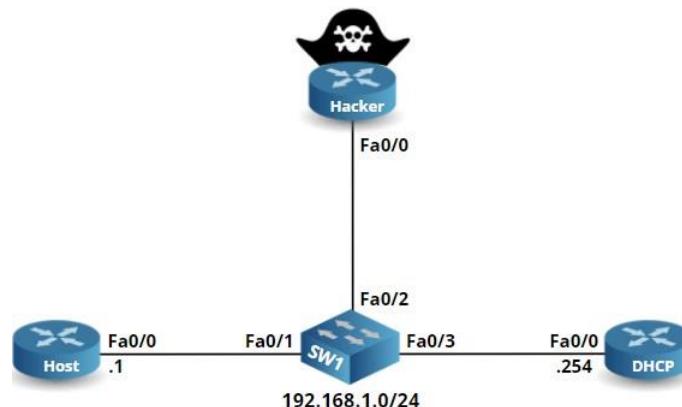
DAI (Inspeção Dinâmica do ARP) é um recurso de segurança que protege o ARP (Protocolo de Resolução de Endereço), que é vulnerável a ataques como envenenamento de ARP (ARP poisoning).

Definição de ARP Poisoning ou ARP Spoofing: É um tipo de ataque no qual uma falsa resposta ARP é enviada a uma requisição ARP original. Enviando uma resposta falsa, o roteador pode ser convencido a enviar dados destinados ao computador 1 para o computador 2, e, por último, o computador redireciona os dados para o computador 1. Se o envenenamento ocorre, o computador 1 não tem ideia do redirecionamento das informações.

A atualização do cache do computador alvo (computador 1) com uma entrada falsa é chamado de Poisoning (envenenamento).

O DAI verifica todos os pacotes ARP em interfaces não-confiáveis comparando as informações do pacote ARP com o banco de dados do ‘DHCP snooping’ ou de uma lista de acesso ARP. Se as informações no pacote ARP não forem relevantes, ele será descartado.

Vamos aprender como configurar o DAI. Eis a topologia que usaremos:



Temos quatro dispositivos, o roteador do lado esquerdo chamado “host” será um cliente DHCP, o roteador do lado direito é o o servidor DHCP, e no topo, temos um roteador que será usado como atacante. O switch no meio será configurado para Dynamic ARP inspection.

Configuração

Começaremos configurando o switch. Primeiro precisamos nos certificar de que todas as interfaces estão na mesma VLAN:

```
SW1(config)#interface range fa0/1 - 3
SW1(config-if-range)#switchport mode access
SW1(config-if-range)#switchport access vlan 123
SW1(config-if-range)#spanning-tree portfast
```

Agora podemos configurar o ‘DHCP snooping’:

```
SW1(config)#ip dhcp snooping
SW1(config)#ip dhcp snooping vlan 123
SW1(config)#no ip dhcp snooping information option
```

Os comandos acima habilitarão o ‘DHCP snooping’ globalmente e também para VLAN 123 desabilitando a inserção da opção 82 nos pacotes DHCP. Agora, temos que tornar a interface que se conecta ao servidor DHCP confiável:

```
SW1(config)#interface FastEthernet 0/3
SW1(config-if)#ip dhcp snooping trust
```

A partir de agora, o switch controlará as mensagens DHCP. Vamos configurar um servidor DHCP no roteador no lado direito:

```
DHCP(config)#ip dhcp pool MEU_POOL
DHCP(dhcp-config)#network 192.168.1.0 255.255.255.0
```

Por agora, isso é tudo de que precisamos configurar. Vejamos se o host consegue obter endereço IP:

```
HOST(config)#interface FastEthernet 0/0
HOST(config-if)#ip address dhcp
```

Após alguns segundos veremos a seguinte mensagem:

```
%DHCP-6-ADDRESS_ASSIGN: Interface FastEthernet0/0 assigned DHCP address 192.168.1.1, mask  
255.255.255.0, hostname HOST
```

Vamos verificar se o switch armazenou essa troca de IP no banco de dados do DHCP snooping:

```
SW1#show ip dhcp snooping binding  
  
MacAddress          IPAddress      Lease(sec)    Type        LAN       Interface  
-----  
00:1D:A1:8B:36:D0  192.168.1.1   86330        dhcp-snooping 123     FastEthernet0/1  
  
Total number of bindings: 1
```

Há uma entrada com o endereço MAC e endereço IP do nosso host. Agora podemos continuar com a configuração do DAI, e para ativá-lo só precisamos de um comando:

```
SW1(config)#ip arp inspection vlan 123
```

O switch verificará todos os pacotes ARP em interfaces não-confiáveis. Lembre-se que todas as interfaces são ‘não-confiáveis’ por padrão. Vamos testar essa configuração configurando o endereço IP do host no ‘dispositivo hacker’:

```
Hacker(config)#interface FastEthernet 0/0  
Hacker(config-if)#ip address 192.168.1.1 255.255.255.0
```

Agora vamos analisar o que acontece quando tentamos enviar um ping do ‘hacker’ para nosso roteador DHCP:

```
Hacker#ping 192.168.1.254  
  
Type escape sequence to abort.  
  
Sending 5, 100-byte ICMP Echos to 192.168.1.254, timeout is 2 seconds:  
  
.....  
  
Success rate is 0 percent (0/5)
```

O ping não obteve resposta. O que o switch nos diz sobre isso?

```
SW1#  
  
%SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Req) on Fa0/2, vlan  
123.([0017.5aed.7af0/192.168.1.1/0000.0000.0000/192.168.1.254/01:20:08 UTC Tue Mar 2 1993])  
  
%SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Req) on Fa0/2, vlan  
123.([0017.5aed.7af0/192.168.1.1/0000.0000.0000/192.168.1.254/01:20:10 UTC Tue Mar 2 1993])  
  
%SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Req) on Fa0/2, vlan  
123.([0017.5aed.7af0/192.168.1.1/0000.0000.0000/192.168.1.254/01:20:10 UTC Tue Mar 2 1993])  
  
%SW_DAI-4-DHCP_SNOOPING_DENY: 1 Invalid ARPs (Req) on Fa0/2, vlan  
123.([0017.5aed.7af0/192.168.1.1/0000.0000.0000/192.168.1.254/01:20:10 UTC Tue Mar 2 1993])
```

Acima, vemos que todas as solicitações ARP do ‘hacker’ foram descartadas. O switch verifica as informações encontradas no ‘ARP request’ e as compara com as informações do ‘DHCP snooping database’ (banco de dados do DHCP Snooping). Como não encontrou correspondência, os pacotes são descartados. É possível ver o número de pacotes ARP descartados com o seguinte comando:

```
SW1#show ip arp inspection
```

Source Mac Validation	: Disabled			
Destination Mac Validation	: Disabled			
IP Address Validation	: Disabled			
Vlan	Configuration	Operation	ACL Match	Static ACL
123	Enabled	Active		
Vlan	ACL Logging	DHCP Logging	Probe Logging	
123	Deny	Deny	Off	
Vlan	Forwarded	Dropped	DHCP Drops	ACL Drops
123	0	5	5	0
Vlan	DHCP Permits	ACL Permits	Probe Permits	Source MAC Failures
123	0	0	0	0
Vlan	Dest MAC Failures	IP Validation Failures	Invalid Protocol Data	
123	0	0	0	0
Vlan	Dest MAC Failures	IP Validation Failures	Invalid Protocol Data	
123	0	0	0	0

Observe o número de drops aumentando. Até agora tudo bem, o ‘hacker’ foi impedido de agir. Porém, ainda temos um problema, mas antes, vamos derrubar a interface do ‘hacker’:

```
HACKER(config)#interface FastEthernet 0/0
HACKER (config-if)#shutdown
```

Observe o que acontece quando tentamos enviar um ping do host para nosso roteador DHCP:

```
HOST#ping 192.168.1.254
Type escape sequence to abort.
```

```

Sending 5, 100-byte ICMP Echos to 192.168.1.254, timeout is 2 seconds:
.....  

Success rate is 0 percent (0/5)

```

Este ping também falhou, mas por quê? Não estamos falsificando nada! Vamos ver o que o switch nos diz:

```

SW1#  

%SW_DAI-4-DHCP_SNOOPING_DENY:      1      Invalid      ARPs      (Res)      on      Fa0/3,      vlan  

123.([0016.c7be.0ec8/192.168.1.254/001d.a18b.36d0/192.168.1.1/01:24:48 UTC Tue Mar 2 1993])  

%SW_DAI-4-DHCP_SNOOPING_DENY:      1      Invalid      ARPs      (Res)      on      Fa0/3,      vlan  

123.([0016.c7be.0ec8/192.168.1.254/001d.a18b.36d0/192.168.1.1/01:24:50 UTC Tue Mar 2 1993])  

%SW_DAI-4-DHCP_SNOOPING_DENY:      1      Invalid      ARPs      (Res)      on      Fa0/3,      vlan  

123.([0016.c7be.0ec8/192.168.1.254/001d.a18b.36d0/192.168.1.1/01:24:52 UTC Tue Mar 2 1993])  

%SW_DAI-4-DHCP_SNOOPING_DENY:      1      Invalid      ARPs      (Res)      on      Fa0/3,      vlan  

123.([0016.c7be.0ec8/192.168.1.254/001d.a18b.36d0/192.168.1.1/01:24:54 UTC Tue Mar 2 1993])  

%SW_DAI-4-DHCP_SNOOPING_DENY:      1      Invalid      ARPs      (Res)      on      Fa0/3,      vlan  

123.([0016.c7be.0ec8/192.168.1.254/001d.a18b.36d0/192.168.1.1/01:24:56 UTC Tue Mar 2 1993])

```

O switch está descartando ARP replies (respostas ARP) do roteador DHCP para o host. Como o roteador DHCP não tem ideia de como alcançar o host, o ping está falhando:

```

HOST#show ip arp  

Protocol Address          Age (min)  Hardware Addr  Type   Interface  

Internet 192.168.1.1           -    001d.a18b.36d0  ARPA   FastEthernet0/0  

Internet 192.168.1.254         0    Incomplete       ARPA

```

```

DHCP#show ip arp  

Protocol Address          Age (min)  Hardware Addr  Type   Interface  

Internet 192.168.1.1           0    001d.a18b.36d0  ARPA   FastEthernet0/0  

Internet 192.168.1.254         -    0016.c7be.0ec8  ARPA   FastEthernet0/0

```

Por que o switch está descartando ‘ARP reply’? O problema é que o roteador DHCP está usando endereços IP estáticos. A DAI verifica o ‘DHCP snooping database’ para todos os pacotes que chegam em interfaces ‘não-confiáveis’, quando não encontra uma correspondência o pacote ARP é descartado. Para corrigir, precisamos criar uma entrada estática para o roteador DHCP:

```

SW1(config)#arp access-list ROTEADOR_DHCP  

SW1(config-arp-nacl)#permit ip host 192.168.1.254 mac host 0016.c7be.0ec8

```

Primeiro, criamos uma access-list ARP com uma declaração de permissão para o endereço IP e o endereço MAC do roteador DHCP. Agora precisamos aplicar ao DAI:

```

SW1(config)#ip arp inspection filter ROTEADOR_DHCP vlan 123 ?  

      static  Apply the ACL statically

```

Usamos o comando ‘**ip arp inspect filter**’, mas é preciso ter cuidado. Se usarmos o parâmetro “static”, estamos dizemos ao switch para não verificar o ‘DHCP snooping database’. Então, ele verificará apenas a access-list ARP e quando não encontrar uma entrada, o pacote ARP será descartado. Certifique-se de adicionar o filtro sem o parâmetro ‘**static**’:

```
SW1(config)#ip arp inspection filter ROTEADOR_DHCP vlan 123
```

O switch agora verificará primeiro a access-list ARP. Quando não encontrar correspondência, verificará o ‘DHCP snooping database’. Vamos tentar o ping novamente:

```
HOST#ping 192.168.1.254
Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.1.254, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms
```

Excelente! O ping agora obteve resposta devido à entrada estática para o roteador DHCP. Outra maneira de lidar com esse problema é configurar a interface como confiável. A DAI permitirá todos os pacotes ARP em interfaces confiáveis:

```
SW1(config)#interface FastEthernet 0/3
SW1(config-if)#ip arp inspection trust
```

Ainda podemos fazer mais algumas coisas com o DAI! Existem algumas verificações de segurança adicionais que podemos ativar:

```
SW1(config)#ip arp inspection validate ?
dst-mac  Validate destination MAC address
ip      Validate IP addresses
src-mac  Validate source MAC address
```

Veja o que essas opções significam:

- **dst-mac**: Verifica o endereço MAC de destino no cabeçalho Ethernet em relação ao endereço MAC de destino no pacote ARP. Esta verificação é executada pelas ‘ARP replies’. As ‘ARP replies’ com endereços MAC diferentes serão descartadas.
- **ip**: Verifica se há endereços IP inválidos e inesperados. Por exemplo: 0.0.0.0, 255.255.255.255 e endereços multicast..
- **src-mac**: Verifica o endereço MAC de origem no cabeçalho Ethernet em relação ao endereço MAC do remetente no pacote ARP. Essa verificação é executada para ‘ARP requests’ e ‘ARP replies’. Os pacotes ARP com endereços MAC diferentes serão descartados.

Só é possível ativar uma dessas opções ao mesmo tempo. Abaixo, um exemplo de como habilitar a verificação dst-mac:

```
SW1(config)#ip arp inspection validate dst-mac
```

Por último, também podemos configurar a ARP rate-limiting (limitação de taxa do ARP). Por padrão, há um limite de 15 pps para o tráfego ARP em interfaces não confiáveis. Veja como podemos alterá-la:

```
SW1(config)#interface FastEthernet 0/1
SW1(config-if)#ip arp inspection limit rate 10
```

Agora, essa interface permitirá 10 pacotes ARP por segundo.

Port Security

Por padrão, não há limite para o número de endereços MAC que um switch pode aprender em uma interface. Todos esses endereços MAC são permitidos. Se quisermos, podemos mudar esse comportamento com o uso do ‘port security’ (segurança da porta). Vamos trabalhar com um cenário bem comum, observe a topologia abaixo:



Alguém conectou um switch barato (não gerenciado) que trouxe de casa à interface FastEthernet 0/1 do switch Cisco. Como resultado, o switch Cisco aprenderá o endereço MAC dos computadores PC1 e PC2 na interface FastEthernet 0/1.

É claro que não queremos que as pessoas tragam seus próprios switches e os conectem na rede da empresa, portanto, precisamos impedir que esse tipo de ‘gambiarra’ funcione:

```
Sw1(config)#interface fa0 / 1
Sw1(config-if)#switchport port-security
Sw1(config-if)#switchport port-security maximum 1
```

Com o comando ‘**switchport port-security**’ nós habilitamos o port security na interface. No caso em tela, configuramos a porta para permitir apenas um endereço MAC. Assim que o switch notar outro endereço MAC na interface, ele identificará que está acontecendo uma violação na política de segurança e poderá tomar algumas atitudes que veremos daqui a pouco.

Além de definir um número máximo de endereços MAC, também podemos usar o ‘port security’ para filtrar qual endereço MAC será permitido. No exemplo a seguir, configuraremos o port security para permitir apenas o endereço MAC aaaa.bbbb.cccc. Este não é o endereço MAC do meu computador, então será perfeito para demonstrar uma violação de segurança:

```
Sw1(config)#interface fa0/1
Sw1(config-if)#switchport port-security mac-address aaaa.bbbb.cccc
```

Use o comando ‘**switchport port-security mac-address**’ para definir o endereço MAC que você deseja permitir. Agora vamos gerar algum tráfego para causar uma violação:

```
C:\Documents and Settings\PC1> ping 1.2.3.4
```

Estamos pingando para um endereço qualquer, não há nada nesse endereço IP 1.2.3.4; o objetivo aqui é só gerar tráfego. Eis o que veremos:

```
Sw1#
%PM-4-ERR_DISABLE: psecure-violation error detected on Fa0/1, putting Fa0/1 in
err-disable state
%PORT_SECURITY-2-PSECURE_VIOLATION: Security violation occurred, caused by MAC
address 0090.cc0e.5023 on port FastEthernet0/1.
```

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to down  
%LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to down
```

Ocorreu uma violação de segurança. Como resultado, a porta entrou em ‘err-disable state’, ou seja, foi desativada por erro. Como vemos abaixo, agora ela está down. Vamos analisar melhor o ‘port security’ com o comando abaixo:

```
Switch#show port-security interface fa0/1  
  
Port Security : Enabled  
Port Status : Secure-shutdown  
Violation Mode : Shutdown  
Aging Time : 0 mins  
Aging Type : Absolute  
SecureStatic Address Aging : Disabled  
Maximum MAC Addresses : 1  
Total MAC Addresses : 1  
Configured MAC Addresses : 1  
Sticky MAC Addresses : 0  
Last Source Address:Vlan : 0090.cc0e.5023:1  
Security Violation Count : 1
```

Este é um comando útil para verificar a configuração do ‘port security’. Use ‘**show port-security interface**’ para ver os detalhes de segurança da porta. Observe que o ‘violation mode’ está em ‘shutdown’ e que a última violação foi causada pelo endereço MAC 0090.cc0e.5023 (PC1).

```
Sw1#show interfaces fa0 / 1  
  
FastEthernet0 / 1 is down, line protocol is down (err-disabled)
```

Desabilitar a interface após uma violação de segurança é uma boa ideia (em termos de segurança), mas o problema é que a interface permanecerá em ‘err-disable state’. Isso provavelmente significa que haverá usuários abrindo incidentes no ‘helpdesk’ e você terá que trazer a interface de volta manualmente! Vamos ativá-la novamente:

```
Switch(config)#interface fa0/1  
Switch(config-if)#shutdown  
Switch(config-if)#no shutdown
```

Para tirar a interface do estado de err-disable, basta digitar “shutdown” seguido de “no shutdown”!

Concorda que seria mais fácil se a interface pudesse se recuperar sozinha após um certo período de tempo. É possível fazer isso, basta habilitar o seguinte comando:

```
Switch(config)#errdisable recovery cause psecure-violation
```

Após 5 minutos (300 segundos), ela sairá automaticamente do estado ‘err-disable’. Certifique-se de resolver o problema, pois caso contrário, haverá outra violação e a interface voltará para o estado ‘err-disable’. Porém, podemos acelerar isso alterando o timer. Vamos defini-lo para 30 segundos:

```
SW1(config)#errdisable recovery interval 30
```

Em vez de digitar o endereço MAC, podemos fazer com que o switch aprenda um endereço MAC sozinho:

```
Sw1(config-if) # no switchport port-security mac-address aaaa.bbbb.cccc  
Sw1(config-if) # switchport port-security mac-address sticky
```

A palavra-chave **sticky** garantirá que o switch utilize o primeiro endereço MAC que aprender na interface. Vamos verificar:

```
Switch#show run interface fa0/1  
  
Building configuration...  
  
Current configuration : 228 bytes  
  
!  
  
interface FastEthernet0/1  
  
switchport mode access  
  
switchport port-security  
  
switchport port-security mac-address sticky  
  
switchport port-security mac-address sticky 000c.2928.5c6c
```

Observe na ‘running-config’ que ele salvará o endereço MAC do PC1.

Utilizar o ‘shutdown’ na interface em caso de violação pode ser uma solução drástica demais. Existem outras opções menos impactantes:

```
Sw1(config-if)#switchport port-security violation ?  
  
protect Security violation protect mode  
restrict Security violation restrict mode  
shutdown Security violation shutdown mode
```

Existem outras opções como **protect** (proteger) e **restrict** (restringir):

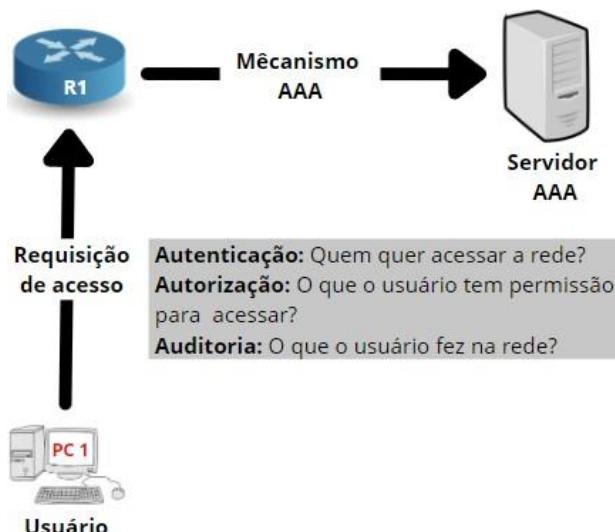
- **Protect:** Frames Ethernet com endereços MAC não permitidos serão descartados, mas você não receberá nenhuma informação sobre isso.
- **Restrict:** Quadros Ethernet com endereços MAC não permitidos serão descartados, mas você verá as informações de registro e um trap SNMP será enviado.
- **Shutdown:** Frames Ethernet com endereços MAC não permitidos farão com que a interface entre no estado de err-disable. Você verá informações de registro e um trap SNMP será enviado. Para a recuperação, temos duas opções:
 - **Manual:** O administrador precisará restaurar a interface com os comandos “shutdown” e “no shutdown”.
 - **Automatic:** Utilizando o comando ‘**errdisable recovery**’ para ativar e configurar a recuperação automática.

5.8 Differentiate authentication, authorization, and accounting concepts

AAA: Autenticação, Autorização e Auditoria é um framework usado para gerenciar a atividade do usuário em uma rede. Com o AAA é possível ter um gerenciamento eficaz da rede, mantendo-a segura, pois ela consegue garantir que apenas usuários que possuem autorização tenham acesso a rede, e todas as atividades que eles realizarem estarão sendo monitoradas e registradas.

AAA utiliza alguns métodos para permitir acesso à rede, por exemplo, exige credenciais autorizadas e autenticadas para provar que os usuários são legítimos. Só então esses usuários obtêm acesso à rede. O AAA é amplamente utilizado em dispositivos de rede como roteadores, switches e firewalls.

Além de todos os benefícios de segurança, o AAA rompe certas limitações de configuração, permitindo um nível de escalabilidade que não seria possível sem ele. Por exemplo, sem o AAA, se for necessário alterar ou adicionar uma senha de usuário, a tarefa deverá ser realizada localmente em todos os dispositivos, o que exigirá muito tempo e recurso. Ter um servidor AAA resolve esse problema já que a tarefa será centralizada em um único servidor, bastando alterar a senha através dele para que ela se replique em todos os dispositivos da rede.



Authentication

A autenticação fornece um método de identificação fazendo com que o usuário insira um nome de usuário e uma senha válidos antes que o acesso à rede seja concedido. A autenticação é baseada em cada usuário, tendo, cada um, um conjunto exclusivo de credenciais de login para obter acesso à rede.

O servidor AAA compara as credenciais de autenticação de um usuário com outras credenciais de usuário armazenadas em um banco de dados. Se as credenciais de login corresponderem, o usuário terá acesso à rede. Se as credenciais não corresponderem, a autenticação falhará e o acesso será negado.

Authorization

Após a autenticação, o usuário deve obter autorização para realizar certas tarefas. Após efetuar login em uma rede, por exemplo, o usuário pode tentar emitir comandos, acessar dispositivos, etc. O processo de autorização determina se o usuário tem autoridade para emitir tais comandos ou acessar determinado lugar.

Simplificando: a autorização é o processo de aplicação de políticas - determinando quais tipos ou qualidades de atividades, recursos ou serviços um usuário tem permissão. Normalmente, a autorização ocorre dentro do contexto de autenticação. Depois de autenticar um usuário, ele pode ser autorizado para diferentes tipos de acesso ou atividades.

Accounting

A última peça da estrutura AAA é a auditoria, que monitora os recursos que um usuário consome durante o acesso à rede. Isso pode incluir a quantidade de tempo que ele pode utilizar o sistema ou a quantidade de dados enviados e recebidos durante uma sessão.

A auditoria é realizada registrando as estatísticas da sessão e as informações de uso. Também é usada para controle de autorização, faturamento, análise de tendências, utilização de recursos e planejamento da capacidade de dados necessária para as operações de negócios. Além disso, temos a função mais utilizada em redes de computadores: monitorar os comandos aplicados e dispositivos acessados pelos administradores e analistas de redes.

Protocolos AAA

Se um único administrador deseja acessar 1000 roteadores e 400 switches, e o banco de dados utilizado para armazenar o nome de usuário e senha (autenticação) for local, o administrador deverá criar a mesma conta de usuário em 1400 dispositivos diferentes. Além disso, se for necessário alterar a senha por alguma questão de segurança, o processo deverá ser realizado manualmente em cada um dos dispositivos. É uma tarefa inglória.

Para facilitar tarefas como essa, podemos utilizar o ACS (Access Control Server - Servidor de Controle de Acesso). O ACS fornece um sistema de gerenciamento central para armazenamento de nome de usuário e senha. Além da ‘autenticação’, o ACS consegue lidar com a ‘autorização’ (o que o usuário está autorizado a fazer). Mas, para isso, temos que configurar o roteador\switch para consultar o ACS para tomadas de decisão sobre autenticação e autorização.

Dois protocolos são usados entre o servidor ACS e o cliente para atender a esse propósito, são eles o RADIUS e TACACS +. Vamos falar de cada um deles:

RADIUS (Remote Authentication Dial-In User Service)

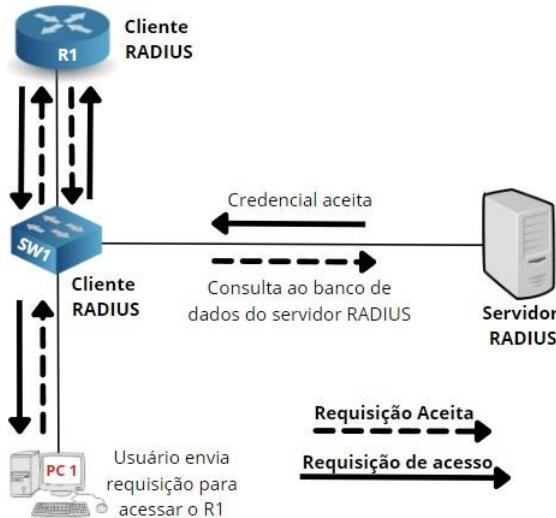
RADIUS é um acrônimo para “Remote Authentication Dial-In User Service”, é um protocolo de segurança usado na estrutura AAA para fornecer autenticação centralizada para usuários que desejam obter acesso à rede.

Recursos:

- Protocolo de padrão aberto para o framework AAA, ou seja, pode ser usado entre dispositivos de qualquer fabricante e o servidor ACS da Cisco.
- Utiliza o protocolo UDP como protocolo de transmissão.
- Utiliza as portas UDP 1812 para autenticação e autorização e 1813 para auditoria.
- Se o dispositivo e o servidor ACS estiverem usando RADIUS, apenas a ‘senha’ dos pacotes AAA serão criptografadas, o restante dos pacotes será transmitido em texto claro, incluindo nome de usuário.
- Fornece mais opções de auditoria que o TACACS +.
- As funções de autenticação e autorização são integradas.

Funcionamento:

Quando outros dispositivos quiserem acessar o Network Access Server (NAS - cliente do RADIUS), ele enviará uma mensagem de solicitação de acesso ao servidor ACS para verificar as credenciais. Em resposta à solicitação de acesso do cliente, o servidor ACS fornecerá uma mensagem de aceitação de acesso (access-accept) se as credenciais forem válidas e rejeição de acesso (access-reject) se as credenciais não corresponderem.



Vantagem:

- Por se tratar de um padrão aberto, pode ser utilizado entre dispositivos de vários fabricantes.
- Mais opções de auditoria que o TACACS +

Desvantagem:

- Como o RADIUS usa UDP, é menos confiável que o TACACS +.
- Nenhuma autorização explícita pode ser implementada.
- O RADIUS criptografa apenas as senhas. Não protege outros dados, como nome de usuário.

TACACS +

TACACS + é um acrônimo para ‘Terminal Access Controller Access Control Server’. É um protocolo de segurança usado na estrutura AAA para fornecer autenticação centralizada para usuários que desejam obter acesso a determinada rede.

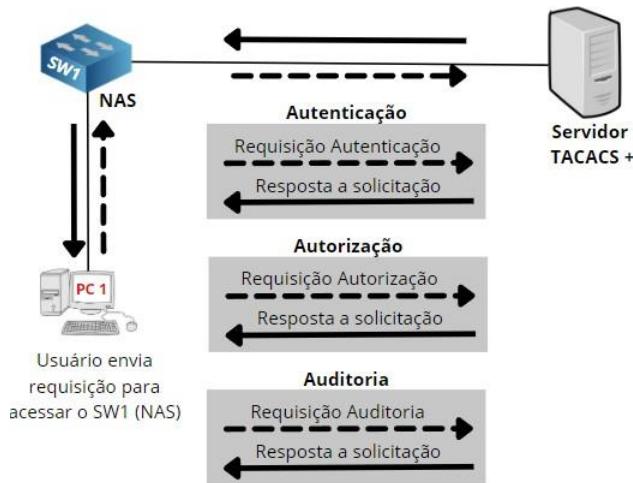
Recursos:

- A Cisco desenvolveu seu próprio protocolo para o framework AAA. Por isso, oferece grande eficiência entre os dispositivos Cisco e o serviços ACS;
- Utiliza o protocolo TCP como protocolo de transmissão.
- Trabalha na porta 49 do TCP.
- Se o dispositivo e o servidor ACS estiverem usando TACACS +, todos os pacotes AAA trocados entre eles serão criptografados.
- Separa o AAA em elementos distintos, ou seja, autenticação, autorização e auditoria são separados.
- Fornece maior controle granular (do que RADIUS) já que os comandos que são autorizados para determinado usuário podem ser especificados.
- Fornece suporte à auditoria, porém, com menos opções que o protocolo RADIUS.

Funcionamento:

Os clientes do protocolo TACACS + são chamados de Network Access Device (Nad - Dispositivo de Acesso à Rede) ou Network Access Device (NAS - Servidor de Acesso à Rede). O funcionamento é bem simples. O dispositivo que deseja acesso à rede entrará em contato com o servidor TACACS + para obter um prompt que solicitará o nome de usuário. Após inserir o nome de usuário, o dispositivo de acesso à rede novamente contactará o servidor TACACS + para obter outro prompt, só que

dessa vez o prompt solicitará a ‘senha’ para o usuário. Após informar a senha, o usuário terá acesso negado ou permitido pelo servidor TACACS+.



Vantagens:

- Fornece controle granular. TACACS + permite que um administrador de rede defina quais comandos um determinado usuário pode executar.
- Todos os pacotes AAA são criptografados incluindo as senhas (No Radius apenas as senhas são criptografadas).
- O TACACS + usa TCP em vez de UDP. O TCP garante a comunicação entre o cliente e o servidor.

Desvantagem:

- Menos opções de auditoria que o RADIUS.

5.9 Describe wireless security protocols (WPA, WPA2, and WPA3)

A Wi-Fi Alliance é uma organização sem fins lucrativos que promove a rede sem fio (wireless) e tem como objetivo uniformizar e padronizar o Wi-Fi. Eles são os responsáveis por fornecer os certificados Wi-Fi Protected Access (WPA).

Atualmente, existem três versões WPA:

- WPA (versão 1)
- WPA2
- WPA3

Para um fabricante de dispositivos wireless obter a certificação WPA, o hardware dos seus equipamentos precisa passar por um processo de teste em laboratórios autorizados. Se o hardware atender os requisitos e critérios, ele receberá o certificado WPA.

WPA oferece suporte a dois modos de autenticação:

- **Personal** - Pessoal
- **Entreprise** – Comercial

Com o modo pessoal, é utilizado uma **chave pré-compartilhada**. Nesse modo, os clientes wireless e o AP utilizam um **handshake de quatro vias**, que usa a chave pré-compartilhada como entrada. Só então as chaves de criptografia são geradas. Quando esse processo é concluído, o cliente wireless e o AP podem, enfim, enviar quadros criptografados um ao outro.

O modo corporativo utiliza **802.1X e um servidor de autenticação**, geralmente um servidor RADIUS. O WPA não especifica um método EAP específico, todos os métodos EAP como PEAP e EAP-TLS são suportados.

WPA

Os primeiros dispositivos sem fio foram certificados para WPA (versão 1) em 2003. WPA é a resposta da Wi-Fi Alliance para substituir o WEP. O WEP em 2003 já era considerado um algoritmo altamente inseguro com diversas vulnerabilidades devido à utilização do RC4.

Existem algoritmos de criptografia muito mais seguros que o WEP, como AES por exemplo. Mas o problema, principalmente em 2003, era que o hardware precisava suportar essa tecnologia. Naquela época, a maioria dos hardwares, seja dos clientes wireless ou Access Points, suportavam apenas o RC4. Portanto, era necessário um algoritmo de software mais seguro e que fosse compatível com os hardwares existentes.

WPA utiliza o **Temporal Key Integrity Protocol** (TKIP), porém, para fins de compatibilidade com o hardware da época, alguns itens do WEP foram reciclados, o que acabou deixando o protocolo inseguro. Algumas coisas foram aperfeiçoadas. O TKIP, por exemplo, utiliza chave de 256 bits em vez das chaves de 64 e 128 bits utilizadas no WEP.

O WPA já nasceu condenado, pois era baseado em partes no padrão 802.11i, que ainda estava em fase de desenvolvimento. Ele era bom o suficiente para substituir o WEP e usar o hardware existente, mas no longo prazo, era evidente a necessidade de um protocolo mais robusto.

WPA2

O WPA2 é o substituto do WPA, é baseado no padrão IEEE 802.11i (ratificado). Embora tenha sido introduzido 2004, só ficou obrigatória para todos os dispositivos que quisessem usar a marca comercial ‘Wi-Fi’ em Março de 2006. A atualização mais significativa, é a utilização de criptografia AES-CCMP no lugar da criptografia RC4 (utilizada pelos protocolos WEP e WPA).

O WPA2 introduziu o **Wi-Fi Protected Setup** (WPS). Antigamente, a única forma de se conectar a uma rede que utilizava PSK (Pre-shared key - chave pré-compartilhada) era conhecendo o SSID e a PSK. O **WPS** abriu novas possibilidades, com ele basta apertar um botão ou inserir um código **PIN**, e o cliente wireless configura automaticamente o SSID e a PSK.

O WPS torna mais fácil para usuários não experientes configurar uma rede sem fio, especialmente quando são utilizadas PSKs longas e complexas, como recomendam as boas práticas de segurança. No entanto, em 2011, os pesquisadores descobriram uma vulnerabilidade no WPS. Através de um ataque de força bruta contra o PIN do WPS, é possível em poucas horas descobrir a PSK.

WPA3

A Wi-Fi Alliance introduziu o WPA3, em substituição ao WPA2, em 2018. O WPA3 ainda usa AES, mas substituiu o CCMP pelo Galois/Counter Mode Protocol (GCMP).

O comprimento da chave do AES aumentou. Embora o WPA3-personal ainda utilize o AES de 128 bits, opcionalmente pode-se utilizar 192 bits. Já para WPA3-enterprise, é obrigatório a utilização de chaves de 192 bits.

O WPA2 introduziu “Protected Management Frame’s” (PMF), mas de forma opcional. O WPA3 tornou obrigatório a utilização do PMF. As duas principais funções do PMF são:

- Proteção extra contra espionagem e falsificação nos frames de gerenciamento unicast.
- Proteção extra contra falsificação nos frames multicast.

Foram implementados novos recursos:

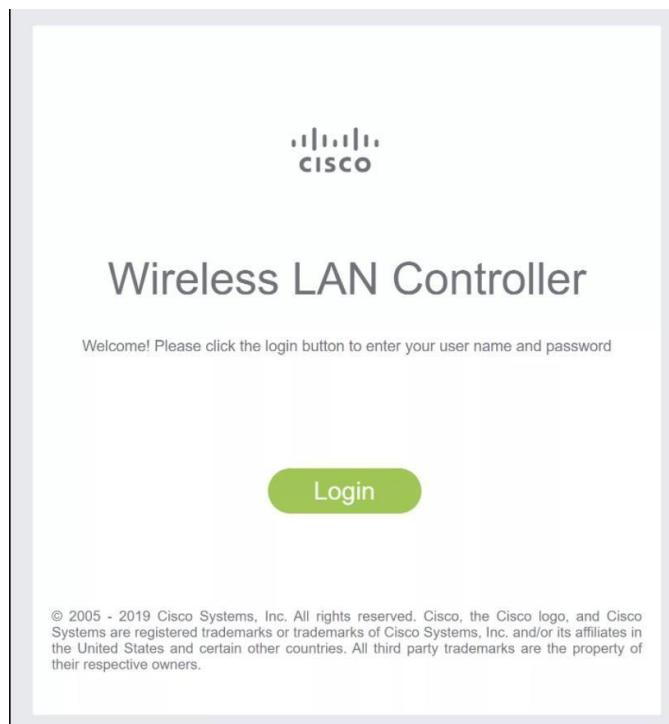
- **Simultaneous Authentication of Equals (SAE) - Autenticação simultânea de iguais:** WPA e WPA2 usam handshake de quatro vias para autenticação, que é vulnerável a um ataque offline. Um invasor pode capturar o handshake de quatro vias e, em seguida, executar um dicionário offline ou um ataque de força bruta. No WPA3, os clientes autenticam com SAE em vez do handshake de quatro vias. SAE é resistente a ataques offline.
- **Forward Secrecy - Sigilo de encaminhamento:** Com WPA ou WPA2, é possível capturar o tráfego sem fio e descriptografá-lo posteriormente, uma vez que você tenha a PSK (chave pré-compartilhada). Com WPA3, isso é impossível. Devido ao sigilo de encaminhamento, não é possível descriptografar o tráfego sem fio posteriormente, mesmo possuindo a PSK (chave pré-compartilhada).
- **Opportunistic Wireless Encryption (OWE):** Esta é uma substituição para autenticação aberta. Com a autenticação aberta, não havia criptografia. OWE adiciona criptografia. A ideia é usar a troca Diffie-Hellman e criptografar o tráfego entre o cliente wireless e o AP. As chaves são diferentes para cada cliente sem fio, portanto, outros clientes não podem descriptografar o tráfego. Ainda não há autenticação, portanto, não há proteção contra APs invasores.
- **Device Provisioning Protocol (DPP) - Protocolo de provisionamento de dispositivo:** Este é o substituto para a solução WPS. Muitos dispositivos de baixo custo (como dispositivos IoT) não possuem interface para configuração de uma PSK (chave pré-compartilhada). Em vez disso, eles contam que o cliente utilize um computador ou smartphone para fazer essa configuração. O DPP permite autenticação de dispositivos através de um QR code ou NFC.

5.10 Configure WLAN using WPA2 PSK using the GUI

Neste tópico vamos aprender a configurar uma WLAN utilizando WPA2 e PSK através da interface gráfica da WLC. Para alcançar esse objetivo criaremos 02 vlans:

- **VLAN 10:** VLAN de gerenciamento.
- **VLAN 20:** Rede wireless para os dados dos usuários.

Primeiro, vamos logar na WLC:



E depois, inserir as credenciais:

Sign in

http://192.168.10.100

Your connection to this site is not private

Username

Password

Na tela seguinte, vamos clicar no botão ‘advanced’, localizado no canto superior direito:

The screenshot shows the Cisco 2500 Series Wireless Controller's Network Summary dashboard. On the left, there's a navigation menu with options like Monitoring, Network Summary, Access Points, Clients, Rogues, Interferers, Wireless Dashboard, AP Performance, Client Performance, and Best Practices. The main area displays a 'NETWORK SUMMARY' card with metrics: 1 Wireless Networks, 2 Access Points, 0 Active Clients (2.4GHz and 5GHz), 25 Rogues (APs and Clients), and 0 Interferers (2.4GHz and 5GHz). Below this is a 'ACCESS POINTS BY USAGE' donut chart and an 'OPERATING SYSTEMS' section. In the top right corner of the dashboard, there's a small 'Advanced' button, which is highlighted with a red box.

O primeiro passo, é configurar uma nova interface dinâmica. Essa interface lógica é como a WLC se conecta à rede cabeadas.

O caminho é: **Controller > Interfaces** e depois clicar em **New**:

This screenshot shows the Cisco Controller interface under the 'Controller' tab. The 'Interfaces' link in the sidebar is highlighted with a red box and labeled '2'. The 'Controller' tab itself is also highlighted with a red box and labeled '1'. In the main pane, there's a table listing two interfaces: 'management' (VLAN 10, IP 192.168.10.100, Static, Enabled) and 'virtual' (N/A, IP 192.0.2.1, Static, Not Supported). At the top right of the interface list, there's a 'New...' button, which is highlighted with a red box and labeled '3'. The top navigation bar includes links for MONITOR, WLANs, CONTROLLER, WIRELESS, SECURITY, MANAGEMENT, COMMANDS, HELP, FEEDBACK, and Home.

Interface Name	VLAN Identifier	IP Address	Interface Type	Dynamic AP Management	IPv6 Address
management	10	192.168.10.100	Static	Enabled	::/128
virtual	N/A	192.0.2.1	Static	Not Supported	

Observe que já temos uma interface de gerenciamento (management) e uma interface virtual. A interface de management é como acessamos a GUI ou CLI (por meio de SSH) da WLC. A interface virtual é usada para DHCP relay, autenticação Web, VPN e outros serviços.

Vamos dar um nome à nova interface e definir o número da VLAN:

Interfaces > New

Interface Name	VLAN20
VLAN Id	20

VLAN 20 foi criada, basta clicar em **Apply** e a WLC apresentará a seguinte tela:

The screenshot shows the 'Interfaces > Edit' configuration page. The left sidebar lists various controller settings like General, Icons, Inventory, Interfaces, and Advanced. The main area has tabs for General Information, Configuration, Physical Information, Interface Address, and DHCP Information. The 'Physical Information' and 'Interface Address' sections are highlighted with red boxes. The 'DHCP Information' section is also highlighted with a red box. The 'Physical Information' section contains fields for Port Number (1), Backup Port (0), Active Port (0), and Enable Dynamic AP Management (unchecked). The 'Interface Address' section contains fields for VLAN Identifier (20), IP Address (192.168.20.100), Netmask (255.255.255.0), and Gateway (192.168.20.254). The 'DHCP Information' section contains fields for Primary DHCP Server (192.168.20.254) and Secondary DHCP Server (empty). The 'Access Control List' and 'mDNS' sections are also visible.

Precisamos inserir algumas informações adicionais para a nova interface dinâmica. O número da porta é a interface física que conecta o WLC à rede cabeada; no caso em tela, é a porta número 1.

Cada interface requer um endereço IP, máscara de sub-rede e gateway padrão. Nessa tela, também configuraremos o servidor DHCP que queremos usar para esta VLAN.

Por fim, clique em **Apply** e teremos uma nova interface dinâmica.

Agora configuraremos uma WLAN.

Clique em **WLANS**, selecione **Create New** e clique em **GO**:

The screenshot shows the 'WLANS' configuration page. The left sidebar has a 'WLANS' section with 'WLANS' and 'Advanced' options. The main area shows a table of existing WLANs. A red circle labeled '1' is over the 'WLANS' tab. A red circle labeled '2' is over the 'Create New' button in the top right corner of the table area. The table columns include WLAN ID, Type, Profile Name, WLAN SSID, Admin Status, and Security Policies. One row is shown with WLAN ID 1, Type WLAN, Profile Name lab, WLAN SSID lab, Admin Status Enabled, and Security Policies [WPA2][Auth(802.1X)].

Na captura de tela acima, você vê que temos a rede wireless “**lab**”. Esta é a rede padrão criada pelo assistente quando configuroi o WLC pela primeira vez. Por padrão, ele usa autenticação 802.1X. Podemos excluir-la ou simplesmente ignorá-la.

Ao selecionar **Create New**, e clicar em **Go**, veremos a seguinte tela:

WLANS > New

Type	WLAN
Profile Name	VLAN20
SSID	VLAN20
ID	2

O nome do perfil é interno, podemos escolher o nome que quisermos. Já o campo SSID é o nome que será anunciado através dos beacons, portanto, este é o nome da rede sem fio que os usuários visualizarão. Vamos chamá-la de “VLAN20”. Clique em **Apply** e você verá esta tela:

WLANS > Edit 'VLAN20'

General Security QoS Policy-Mapping Advanced

Profile Name	VLAN20
Type	WLAN
SSID	VLAN20
Status	<input checked="" type="checkbox"/> Enabled
Security Policies [WPA2][Auth(802.1X)] (Modifications done under security tab will appear after applying the changes.)	
Radio Policy	All
Interface/Interface Group(G)	vlan20
Multicast Vlan Feature	<input type="checkbox"/> Enabled
Broadcast SSID	<input checked="" type="checkbox"/> Enabled
NAS-ID	none

Na guia **General**, existem dois itens importantes:

- **Status:** Clique na caixa de seleção para **habilitar** a WLAN.
- **Interface:** Selecione a interface dinâmica que criamos para esta VLAN.

Clique na guia **Security** e selecione a subguia **layer 2**:

WLANS > Edit 'VLAN20'

General **Security** **QoS** **Policy-Mapping** **Advanced**

Layer 2 **Layer 3** **AAA Servers**

Layer 2 Security: WPA+WPA2
MAC Filtering:

Fast Transition:
Fast Transition: Adaptive
Over the DS:
Reassociation Timeout: 20 Seconds

Protected Management Frame:
PMF: Disabled

WPA+WPA2 Parameters

WPA Policy: <input type="checkbox"/>
WPA2 Policy: <input checked="" type="checkbox"/>
WPA2 Encryption: <input checked="" type="checkbox"/> AES
TKIP: <input type="checkbox"/>
CCMP256: <input type="checkbox"/>
GCMP128: <input type="checkbox"/>
GCMP256: <input type="checkbox"/>

OSEN Policy:

Authentication Key Management

802.1X: <input type="checkbox"/> Enable
CCKM: <input type="checkbox"/> Enable
PSK: <input checked="" type="checkbox"/> Enable
FT 802.1X: <input type="checkbox"/> Enable
FT PSK: <input type="checkbox"/> Enable
PSK Format: ASCII *****
SUITEB-1X: <input type="checkbox"/> Enable
SUITEB192-1X: <input type="checkbox"/> Enable
WPA gtk-randomize State: <input type="checkbox"/> Disable

Lobby Admin Configuration

Lobby Admin Access:

A política padrão é utilizar criptografia WPA2 e AES, que é justamente o que queremos. Precisamos fazer algumas alterações no **Authentication Key Management** (gerenciamento de chave de autenticação). Por default o WLC usa a autenticação 802.1X. Clique na caixa de seleção **PSK** e insira o **PSK format** selecione ASCII. Clique no botão Apply e a configuração estará completa.

Exercícios:

1. Qual comando evita que as senhas sejam armazenadas na configuração do roteador ou switch como texto simples?
 - a) enable secret
 - b) service password-encryption
 - c) username Cisco password encrypt
 - d) enable password
2. Qual conjunto de ações satisfaz os requisitos de autenticação multifator?
 - a) O usuário insere a senha e recebe uma pergunta previamente cadastrada.
 - b) O usuário insere um nome de usuário e uma senha e, a seguir, clica em uma notificação em um aplicativo de autenticação em um dispositivo móvel.
 - c) O usuário insere um PIN em um token RSA e, em seguida, insere a chave RSA na tela de login.
 - d) O usuário insere um nome de usuário e senha e, em seguida, insere novamente as credenciais em uma segunda tela de login.
3. Qual tipo de criptografia wireless é usada pelo WPA2 no modo de chave pré-compartilhada (pre-shared key mode)?
 - a) TKIP with RC4
 - b) RC4
 - c) AES-128
 - d) AES-256
4. Qual configuração é necessária em um roteador para gerar uma chave RSA a ser utilizada em conexões SSH?
 - a) Configura a versão do SSH.
 - b) Configurar o acesso VTY
 - c) Criar usuário e senha
 - d) Atribuir um nome de domínio DNS.
5. Qual é a principal diferença entre 'autenticação' e 'autorização' no protocolo AAA?
 - a) Autenticação verifica o nome de usuário e senha, e a autorização trata da comunicação entre o agente de autenticação e o banco de dados do usuário.
 - b) Autenticação identifica o usuário que está tentando acessar o sistema e a autorização valida a senha do usuário.
 - c) Autenticação identifica e verifica o usuário que está tentando acessar o sistema, e a autorização controla as tarefas que o usuário pode executar.
 - d) Autenticação controla os processos do sistema que o usuário pode acessar e autorização realiza os registros das atividades que o usuário inicia.
6. Um administrador de rede precisa proteger as interfaces não utilizadas que estão configuradas na VLAN padrão de um switch. Quais são as duas etapas que atenderão essa solicitação?
 - a) Configurar as interfaces em um EtherChannel.
 - b) Aplicar shutdown nas interfaces.
 - c) Configurar as interfaces em modo de acesso e coloca-las na VLAN 99.
 - d) Configurar as interfaces como trunk.
 - e) Ativar o Cisco Discovery Protocol (CDP)
7. Quando uma VPN site a site está sendo utilizada, qual protocolo é responsável pelo transporte dos dados do usuário?
 - a) IKEv2
 - b) IKEv1
 - c) IPsec
 - d) MD5
8. Um engenheiro precisa configurar uma WLAN usando o tipo de criptografia mais forte para WPA2-PSK. Qual cifra atende ao requisito dessa configuração?
 - a) WEP
 - b) RC4
 - c) AES
 - d) TKIP
9. AAA significa authentication, authorization e accounting?
 - a) Verdadeiro
 - b) Falso
10. Qual efeito do comando 'aaa new-model'?
 - a) Habilita serviços AAA no dispositivo
 - b) Configura o dispositivo para se conectar a um servidor RADIUS através do AAA

- c) Associa um servidor RADIUS a um grupo.
 - d) Configura um usuário local no dispositivo.
11. Quais as duas maneiras um gerenciador de senhas reduz a chance de um hacker roubar a senha de um usuário?
- a) Fornecendo automaticamente um segundo fator de autenticação desconhecido para o usuário original.
 - b) Através de um firewall interno, protegendo assim o repositório de senhas de acesso não autorizado.
 - c) Ele protege contra o keystroke logging (ação de gravar/registrar as teclas pressionadas em um teclado) em um dispositivo ou site comprometido.
 - d) Armazenando o repositório de senhas na estação de trabalho com funcionalidade antivírus e anti-malware embutida
 - e) Incentiva os usuários a criarem senhas mais fortes.
12. Quando uma WLAN WPA2-PSK é configurada na Controladora wireless, qual o número mínimo de caracteres que necessário para formar ASCII?
- a) 6
 - b) 8
 - c) 12
 - d) 18
13. Qual tipo de ataque pode ser mitigado pelo "dynamic ARP inspection"?
- a) Worm
 - b) Malware
 - c) DDoS
 - d) Man-in-the-middle
14. Qual prática protege uma rede de ataques 'VLAN hopping'?
- a) Habilitar o 'dynamic ARP inspection'
 - b) Configurar uma ACL para evitar que o tráfego mude de VLAN
 - c) Alterar a VLAN nativa para uma VLAN ID não utilizada
 - d) Implementar port security nas VLANs conectadas a Internet
15. Qual programa de segurança é violado quando um grupo de funcionários entra em um prédio usando o crachá de apenas uma pessoa?
- a) Detecção de intrusão
 - b) Conscientização do usuário
 - c) Controle de acesso físico
 - d) Autorização de rede
16. Qual dispositivo rastreia o estado das conexões ativas para tomar decisões de encaminhamento de pacotes?
- a) Wireless access point
 - b) Firewall
 - c) Wireless LAN controller
 - d) Roteador
17. Qual combinação fornece a criptografia mais forte para um ambiente wireless?
- a) WPA2 + AES
 - b) WPA + AES
 - c) WEP
 - d) WPA + TKIP
18. O que faz uma interface entrar no estado de 'err-disabled'?
- a) Latência
 - b) Violações de segurança na interface
 - c) Aplicar o comando de 'shutdown' na interface
 - d) Não ter nada plugado na interface.
19. Qual elemento do programa de segurança envolve a instalação de leitores de crachá nas portas do data center para permitir que os trabalhadores entrem e saiam com base em suas funções de trabalho?
- a) Tokens
 - b) Biometria
 - c) Autenticação multifator
 - d) Controle de acesso físico

Respostas: 1) b, 2) b, 3) d, 4) d, 5) c, 6) b, c 7) c, 8) c, 9) a, 10) a, 11) c, e, 12) b, 13) d, 14) c, 15) c, 16) b, 17) a, 18) b, 19) d

6.0 Automation and Programmability

Esse é o último grande bloco, nele vamos falar sobre a parte de automação e programação. É uma parte bem teórica e curta. É importante lembrar que aqui é apresentado noções básicas de programação e automação, a Cisco não espera ou deseja que você se torne programador para fazer a prova CCNA. Tenha isso em mente ao ler esse último bloco.

6.1 Explain how automation impacts network management

Redes de computadores é um universo que está sempre em constante evolução, com novas ferramentas e conceitos. Com essa constante evolução, as exigências aumentaram e com ela a necessidade de novas metodologias para suprir esse crescimento. Uma das tecnologias emergente que cada dia ocupa mais espaço é a automação de redes. Neste tópico, veremos como a automação afeta o gerenciamento da rede.

Com a chegada da automação, várias atividades de gerenciamento ficaram bem mais simples de serem executadas, sejam atividades de teste, implementação ou até mesmo configuração. A automação torna às tarefas menos complexas e repetitivas, por exemplo: Configurações e alterações de centenas de dispositivos podem ser feitas automaticamente com meia dúzia de linhas de comando em uma controladora central.

Atualmente, boa parte das grandes empresas utilizam automação em pelo menos uma área. Pesquisa recente apresentada na Cisco Live de 2019, mostrou que pelo menos 85% das empresas utilizam automação. Diferentes ferramentas de automação são utilizadas, por exemplo, 53% das empresas pesquisadas utilizam ferramentas de automação para configuração de dispositivos (Configuration Automation Tool). Em segundo lugar na pesquisa com 40%, veio automação para configuração, e em terceiro a automação para gerenciamento de políticas (policy Management Automation).

Benefícios da automação de rede

Existem vários benefícios na Automação de Rede, sendo uma das principais, facilitar a função dos engenheiros e administradores de rede. Abaixo, os principais benefícios da Automação de Redes:

- Redução nos custos de operação;
- Maior disponibilidade;
- Diminuição dos erros;
- Melhor controle da rede;
- Maior agilidade nos negócios.

Com automação da rede, o custo para manutenção diminui, pois muitas das atividades agora são realizadas de forma automática, e com maior velocidade. Isso reduz a complexidade do trabalho, ao mesmo tempo que diminui as horas de trabalho necessárias para realização de determinada tarefa. Um efeito que pode ser visto como negativo, é a diminuição da mão de obra necessária para determinadas tarefas, uma equipe com dez analistas de rede poderá ser substituída tranquilamente por apenas um, que com auxílio da automação será capaz de resolver problemas de atualização, configuração, etc.

Até o momento, 95% das atividades relacionadas a rede são realizadas manualmente por humanos, e como sabemos, humanos são falhos, principalmente quando realizam tarefas repetitivas. É assustador o número de paralisações e quedas ocasionadas por erro humano. A automação diminuirá a possibilidade de erro causado por falha humana.

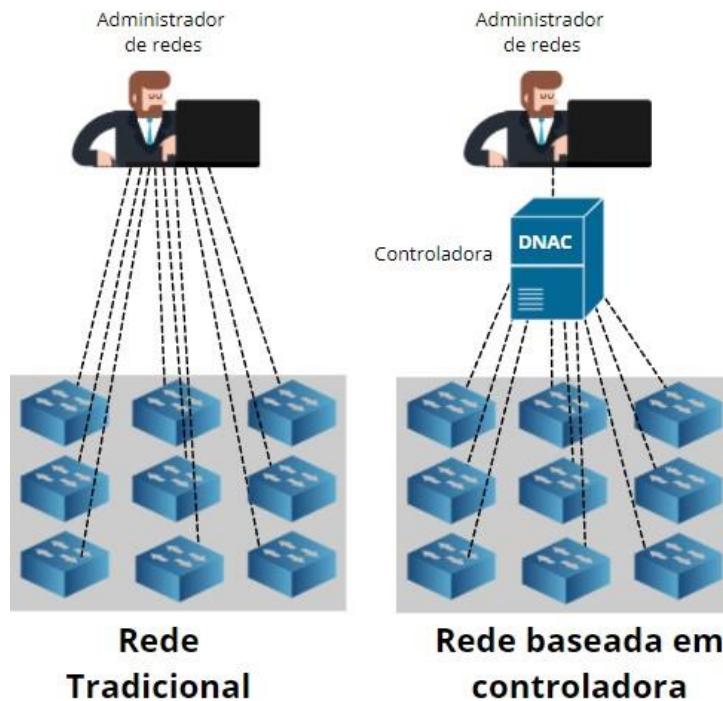
Isso influencia diretamente no ‘up time’ de uma rede. O ‘up time’ é o principal parâmetro para medir a qualidade de uma rede. Com atividades manuais, sujeitas a erro, aumenta o tempo de indisponibilidade de uma rede, justamente por erros causados por administradores\engenheiros. Com a automação, a tendência é que a rede opere sem erros humanos, aumentando assim o ‘up time’.

Usando programas de automação, será possível ter uma visão mais precisa da rede e em tempo real. Isso, facilitará o acesso a todo tipo de relatórios, como: Dados da rede, relatórios de desempenho, possíveis gargalos, etc.

No futuro, as funções de Administrador\Engenheiro de Redes mudarão, exigindo novas qualificações e habilidades, é sobre esse futuro que este bloco trata.

6.2 Compare traditional networks with controller-based networking

Redes baseadas em controladoras fornecem um único ‘portal’, uma forma centralizada de gerenciamento de rede. Em vez de gerenciar os dispositivos de rede individualmente, tendo que configurar caixa por caixa, podemos simplesmente fazer login na controladora e de lá provisionar novos equipamento, realizar troubleshooting:



Observe o desenho acima, o administrador de redes conectado na controladora DNAC, consegue ter uma visão geral da rede e acessar todos os dispositivos de forma centralizada. Isso facilita muito a vida do administrador, por exemplo, em uma rede tradicional, o administrador para adicionar uma vlan teria que entrar em todos os switches de forma individual e digitar os comandos. Utilizando uma controladora, basta ‘dizer’ a controladora o que deve ser feito na rede, e ela configurará todos os dispositivos.

Não é difícil perceber com o exemplo acima que redes baseadas em controladora reduz a complexidade da configuração, enquanto redes baseadas no modelo tradicional o potencial de erro aumenta consideravelmente.

6.3 Describe controller-based and software defined architectures (overlay, underlay, and fabric)

Redes corporativas podem ser bem complexas. Geralmente há um ponto central, filiais remotas, funcionários trabalhando de casa, e tudo isso está conectado através de conexões WAN. Há diversos dispositivos atuando fisicamente para que tudo isso funcione, incluindo roteadores, switches, firewalls, controladoras wireless, etc.

Do ponto de vista da topologia lógica vemos que há muito mais coisas acontecendo: VLANs, VRFs, protocolos de roteamento, access list, regras de firewall, etc. E quase todas essas ferramentas são configuradas manualmente, em cada um desses dispositivos.

Em 2007/2008, o SDN (software defined network) apareceu com a promessa de automatizar tudo, eliminando de forma definitiva a CLI, e fazendo com que todos essa parte lógica relatada anteriormente fosse definida (configurada) por softwares.

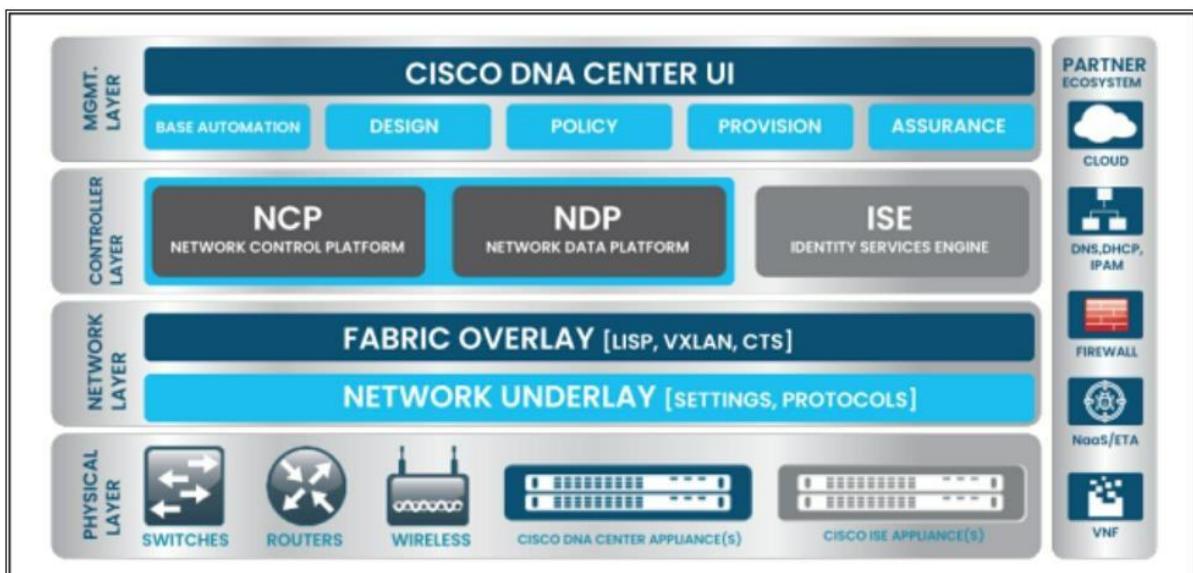
Porém, SDN se aplica mais para datacenters, afinal, em um data center quase tudo gira em torno de aplicações. Quando tratamos da rede de uma empresa, lidamos com um cenário diferente em que quase tudo gira em torno de usuários e dispositivos fixos e móveis. Por exemplo, temos usuários trabalhando em todos os lugares usando laptops, tablets e smartphones.

Hoje em dia, as redes corporativas usam praticamente apenas dispositivos de hardware. A empresa precisa de um novo firewall? Ela simplesmente compra uma nova caixa do Cisco ASA. A Rede cresceu e precisa de mais uma Controladora wireless (WLC)? E empresa compra outro appliance WLC.

Só que as empresas chegaram à conclusão que faz mais sentido financeiro e operacional a substituição de novos dispositivos por serviços, de forma semelhante aos serviços em nuvem! Neste caso, se precisarmos de um novo firewall, basta clicar em um botão e iniciar uma nova instância do firewall vASA. Precisa de outro WLC? Basta clicar em um botão e um novo WLC virtual irá surgir.

Esta é uma das promessas do Cisco SD-access: Automação completa da rede, semelhante ao funcionamento das soluções SDN/nuvem. Cisco SD-Access é um dos elementos mais importantes da Cisco Digital Network Architecture (**Cisco DNA**).

A solução Cisco SD-Access pode ser dividida em cinco camadas básicas cada uma com suas subdivisões. Enfatizaremos os relacionamentos entre essas cinco camadas de uma perspectiva arquitetural:



1. **Camada física:** Compreende os elementos de hardware, como roteadores, switches e dispositivos wireless, interfaces, clusters ou switches virtuais, bem como servidores.
2. **Camada de rede:** Compreende os planos de controle e de dados, e os elementos do plano de política que constituem a network underlay e fabric overlay.
3. **Camada da controladora:** Compreende os elementos do sistema de software de gerenciamento e orquestração dos subsistemas associados, como automação, identidade e análise (automation, identity e analytics).
4. **Camada de gerenciamento:** Compreende os elementos com os quais os usuários interagem, em particular a interface gráfica do usuário (GUI), bem como APIs e interfaces de linha de comando (CLIs).
5. **Ecossistema de parceiros:** Compreende todos os sistemas da Cisco e de parceiros que são capazes de aumentar e aproveitar os serviços do SD-Access.

Overlay, underlay, and fabric

Vamos começar pelo **Fabric**. Aqui que encontramos todos os componentes de hardware com os quais estamos familiarizados: Roteadores, switches, controladores wireless, access point, etc. Importante ressaltar que inclui dispositivos que executam IOS e IOS XE.

SD-access utiliza **underlay** e **overlay**, nela cria-se um overlay (via software) sobre o underlay (camada física). E como passamos a operar no overlay ganhamos flexibilidade e agilidade. É muito mais fácil segmentar e alterar a rede sem precisar mudar a estrutura física.

Nós não vemos essa separação (rede underlay e overlay) na maioria das redes corporativas, mas ela faz todo sentido. Por exemplo, se uma nova aplicação for implementada na rede, pode ser necessário realizar algumas alterações, como uma access-list. Porém, ao alterar esta access-list, poderemos afetar outras aplicações na rede. Através da separação entre underlay e overlay, esse risco desaparece. Vamos a uma definição mais direta:

- **Underlay:** É a rede física (switches, roteadores e suas conexões), com configurações que permitem a comunicação entre equipamentos e gerência (DNA-Center). O underlay pode, teoricamente, ter qualquer topologia, mas a recomendação é criarmos uma estrutura baseada em camada 3, usando ISIS como protocolo de roteamento. Inclusive este é o cenário que o DNA-Center cria quando o utilizamos para criar o underlay automaticamente.
- **Overlay:** É a rede definida por software, criada sobre o underlay, onde temos a abstração da parte física. É possível inclusive termos várias redes “virtuais” sobre o mesmo underlay. Na camada de overlay o SD-Access utiliza VXLAN (Data Plane) para encapsular e transportar os pacotes IPs pela Fabric. Também é utilizado o LISP – Locator/ID Separation Protocol (Control Plane), para rotear o tráfego VXLAN.

Com a separação entre redes **underlay** e **overlay**, as alterações na rede overlay não afetarão a rede underlay. É semelhante ao uso de protocolos de tunelamento como GRE ou DMVPN. Você pode mexer nos túneis, mas não afetará a rede subjacente (underlying network).

Usamos APIs para configurar os dispositivos de hardware e implementar novos serviços. Porém, ainda é possível usar a CLI para solucionar problemas.

O Fabric consiste em três componentes principais, que ainda veremos mais detalhadamente:

- **Control Plane (Plano de controle):** Baseado no Locator Identity Separator Protocol (LISP)
- **Data Plane (Plano de dados):** Baseado em Virtual Extensible LAN (VXLAN)
- **Policy Plane (Plano de política):** Baseado em Cisco TrustSec (CTS)

O LISP simplifica o roteamento removendo informações de destino da tabela de roteamento e movendo-as para um sistema de mapeamento centralizado. É semelhante ao DNS, um roteador envia uma consulta a um sistema de mapeamento central, chamado LISP, perguntando onde está o endereço de destino. Isso resulta em tabelas de roteamento menores e requer menos ciclos de CPU.

Poderíamos utilizar LISP na ‘data plane’, mas ele cria túneis apenas para o tráfego L3. SD-access utiliza uma versão modificada do **VXLAN** no data plane, um dos motivos para isso, é que VXLAN oferece suporte ao encapsulamento L2.

No policy plane, usamos Cisco TrustSec, Scalable Group Tags (SGT) e SGT Exchange Protocol (SXP). Adicionamos dispositivos finais a um grupo e através do SGT forçamos que esses dispositivos tenham uma política de rede semelhante. O SGT é uma tag separada do endereço de rede, e como ele podemos anexar políticas de rede como QoS, PBR, etc.

Isso nos dá a flexibilidade de criar políticas de rede sem mapeá-las por endereços IP ou sub-redes. O SGT é adicionado ao cabeçalho VXLAN.

6.3.a Separation of control plane and data plane

Para falarmos da separação do ‘data plane’ e ‘control plane’ precisamos rever o funcionamento de uma rede ‘tradicional’.

A TI avançou nos mais diversos campos nos últimos 30 anos, porém, a parte de ‘networking’ permanece semelhante ao que era 30 anos atrás. Ainda temos os mesmos dispositivos de rede com finalidade específicas, como roteadores, switches e firewalls.

Esses dispositivos de rede são vendidos por fabricantes de equipamento de redes como a Cisco, e costumam usar hardware proprietário. A maioria desses dispositivos é configurada principalmente por meio da CLI, embora existam alguns produtos que utilizam GUI, como CCP (Cisco Configuration Protocol) para os roteadores e ASDM para os firewalls Cisco ASA.

Um dispositivo de rede, como um roteador, executa diferentes funções em diferentes níveis para atividades corriqueiras. Vamos ver os passos que um roteador faz para encaminhar um pacote IP:

- Verifica o endereço IP de destino na tabela de roteamento com a finalidade de descobrir para onde deve encaminhar o pacote IP.
- Utiliza protocolos de roteamento como OSPF, EIGRP ou BGP para aprender as redes e coloca-las na tabela de roteamento.
- Utiliza o protocolo ARP para descobrir o endereço MAC de destino do próximo salto e se necessário altera o endereço MAC de destino no quadro Ethernet.
- O TTL (Time to Live) no pacote IP deve ser diminuído em 1 e a soma de verificação do cabeçalho IP deve ser recalculada.
- A soma de verificação do quadro Ethernet deve ser recalculada.

Todas essas tarefas diferentes são separadas por planos diferentes. Existem três planos como dito anteriormente:

1. Control Plane (Plano de controle)
2. Data Plane (Plano de dados)
3. Management plane (Plano de gerenciamento)

Veremos a função cada um deles:

Control plane

O plano de controle é responsável por **trocar informações de roteamento**, construir a tabela ARP, etc. Eis algumas tarefas que são realizadas pelo plano de controle:

- Aprender endereços MAC para construção da tabela de endereços MAC no switch.
- Executar o STP para criar uma topologia livre de loops.
- Construir tabelas ARP.
- Executar protocolos de roteamento como OSPF, EIGRP e BGP e construir a tabela de roteamento.

Data Plane

O plano de dados é responsável por **encaminhar o tráfego**. Ele se baseia nas informações fornecidas pelo plano de controle. Eis algumas tarefas que o plano de dados realiza:

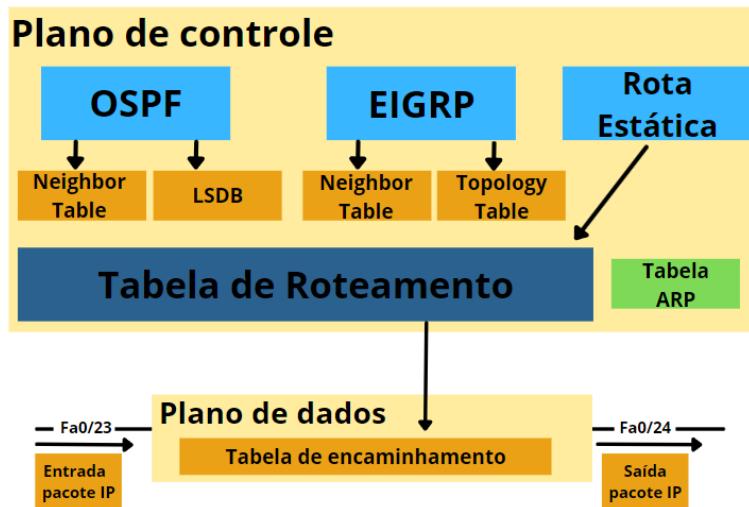
- Encapsula e desencapsula pacotes.
- Adiciona ou remove cabeçalhos, como o cabeçalho 802.1Q.
- Encaminha frames para o endereço MAC correspondente na tabela MAC.
- Encaminha pacotes IPs para os destinos que tenham correspondência na tabela de roteamento.
- Altera os endereços de origem e destino ao usar o NAT.
- Elimina o tráfego bloqueado pelas ‘access-list’.

As tarefas do **plano de dados devem ser executadas o mais rápido possível**, por isso o encaminhamento de tráfego é executado por hardware especializado como ASICs e tabelas TCAM.

Management plane

O plano de gerenciamento é usado para acesso e gerenciamento dos dispositivos de rede. Por exemplo, para acesso os dispositivos através de telnet, SSH ou porta do console.

Para aprofundarmos no SDN é preciso ter um entendimento claro sobre o plano controle e o plano de dados. Observe a ilustração abaixo, onde fica fácil visualizar as diferenças entre os dois planos e como cada um deles funciona:



Acima é possível ver o ‘control plane’, onde são usados os protocolos de roteamento como OSPF e EIGRP e até mesmo roteamento estático. As melhores rotas são instaladas na tabela de roteamento. Outra tabela que o roteador deve construir é a tabela ARP.

As informações da tabela de roteamento e ARP, são usadas para construir a tabela de encaminhamento (forwarding table). Quando o roteador receber um pacote IP, ele será capaz de encaminhá-lo rapidamente, pois a tabela de encaminhamento já foi construída.

6.3.b North-bound and south-bound APIs

Para falarmos da North-bound e South-bound precisamos entrar ainda mais em SDN.

SDN (Software Defined Networking)

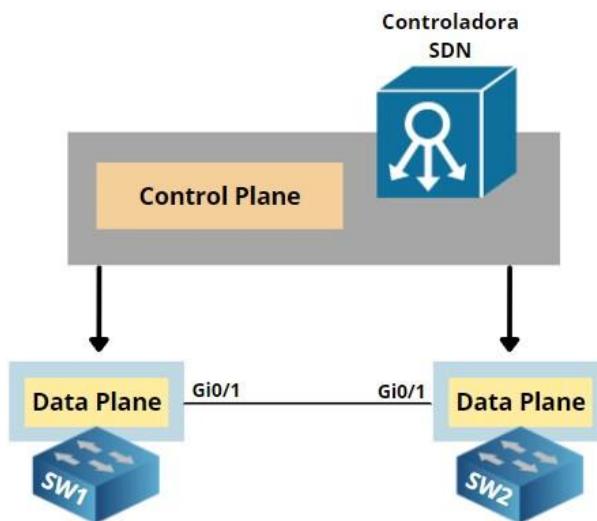
Assim como a palavra da moda “nuvem”, cada organização ou fornecedor tem uma opinião diferente sobre o que exatamente é SDN, e os diferentes produtos que oferecem sobre essa tecnologia.

A rede tradicional usa um modelo distribuído no plano de controle. Protocolos como ARP, STP, OSPF, EIGRP, BGP e outros são executados separadamente em cada dispositivo da rede. Esses dispositivos se comunicam entre si, mas não há um dispositivo central que tenha uma visão geral ou que controle toda a rede.

Uma exceção são as controladoras wireless (WLC). Ao configurar uma rede sem fio, tudo pode ser configurado diretamente na WLC, e ela então controlará e configurará os access point. Não é necessário configurar cada access point separadamente, é tudo realizado pela WLC.

Com SDN, usamos uma controladora central para o ‘control plane’. Dependendo da solução SDN adotada, a controladora SDN poderá assumir 100% do ‘control plane’ ou obter apenas algumas informações sobre os dispositivos da rede. A controladora SDN pode ser um dispositivo físico de hardware ou uma máquina virtual.

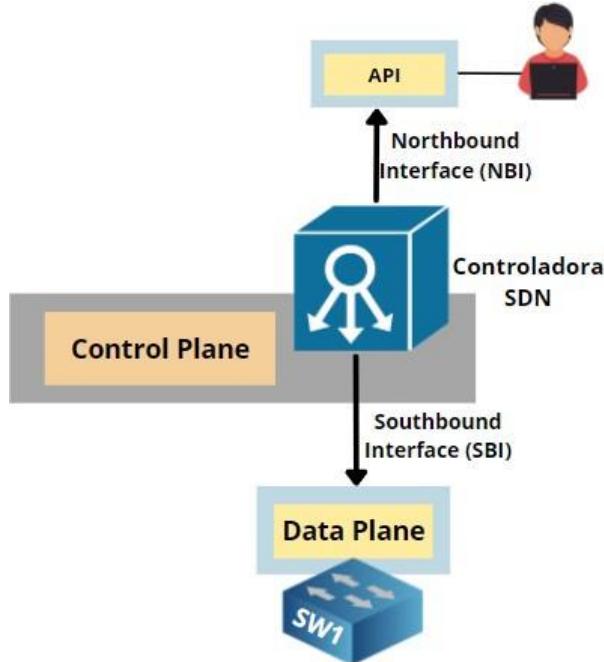
Eis uma ilustração para ajudá-lo a visualizar melhor todo esse processo:



Observe que a controladora SDN é a responsável pelo ‘control plane’. Os switches se tornaram dispositivos “burros”, que apenas encaminham pacote através do ‘data plane’. A controladora SDN é responsável por alimentar o ‘data plane’ dos switches com informações fornecidas pelo ‘control plane’.

Existem vantagens e desvantagens em ter um ‘control plane’ centralizado. Uma das vantagens de ter uma controladora central é que podemos configurar toda a rede a partir de um único dispositivo. Essa controladora terá acesso total e visão de tudo o que está acontecendo na rede.

Vamos adicionar mais alguns detalhes a essa estrutura. É importante saber que a controladora SDN utiliza duas interfaces especiais. Observe a imagem abaixo:



Essas interfaces de comunicação são chamadas de **Northbound interface (NBI)** e **Southbound interface (SBI)**:

Southbound Interface

A controladora SDN deve se comunicar com os dispositivos de rede para programar o ‘data plane’. Isso é feito por meio da Southbound interface. Esta não é uma interface física, mas uma interface lógica, geralmente uma API (Application Programming Interface).

Uma API é uma interface de software que permite que um aplicativo dê acesso a outros aplicativos usando funções e estruturas de dados predefinidas. Explicarei mais API em instantes.

As Southbound interface mais utilizadas são:

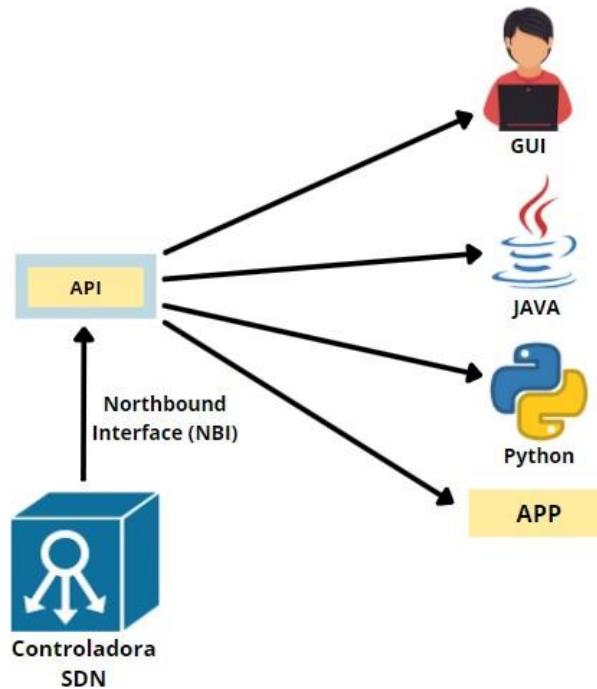
- **OpenFlow:** Este é provavelmente o SBI mais popular, é um protocolo de código aberto da Open Networking Foundation. Existem alguns dispositivos de rede e controladoras SDN que oferecem suporte integral a OpenFlow.
- **Cisco OpFlex:** É a resposta da Cisco ao OpenFlow. É um protocolo de código aberto que foi submetido ao IETF para padronização.
- **CLI:** Cisco oferece APIC-EM, é uma solução SDN para a geração atual de roteadores e switches. Utiliza protocolos que estão disponíveis no hardware da geração atual, como telnet, SSH e SNMP.

Northbound Interface

A northbound interface é usada para acessar a própria controladora SDN. Isso permite que um administrador de rede accesse o SDN para configurá-la ou extraír informações. Pode ser realizado por meio de uma GUI, mas também oferece uma API que permite que outros aplicativos accedam a controladora SDN. Podemos usa-la para escrever scripts e automatizar a administração da rede. Eis alguns exemplos:

- Listar informações de todos os dispositivos que compõe a rede.
- Mostrar o status de todas as interfaces físicas.
- Adicionar uma nova VLAN em todos os switches.
- Mostrar a topologia de toda a rede.
- Configurar automaticamente endereços IP, roteamento e access-list quando uma nova máquina virtual for criada.

Abaixo, uma ilustração para ajudá-lo a visualizar todo esse processo:



Por meio da API, várias aplicações podem acceder a controladora SDN:

- Um usuário que está usando uma GUI para extraír informações sobre a rede. Nos bastidores, a GUI está usando uma API.
- Scripts escritos em Java ou Python podem usar a API para extraír informações da controladora SDN ou configurar a rede.

6.4 Compare traditional campus device management with Cisco DNA Center enabled device management

Cisco DNA Center é um software de gerenciamento e uma controladora para SD-Access. No momento, está disponível apenas como um dispositivo de hardware (appliance). O DNA Center está posicionado na linha de produtos da Cisco como substituto do Cisco APIC-EM. Ele também pode substituir a Cisco Prime Infrastructure.

Existem 3 opções disponíveis:

- DN2-HW-APL (C220 M5, 44 núcleos) - até 1000 dispositivos
- DN2-HW-APL-L (C220 M5, 56 núcleos) - até 2.000 dispositivos
- DN2-HW-APL-XL (C480 M5, 112 núcleos) - até 5.000 dispositivos

A Cisco oferece suporte à implantação de um único nó ou cluster com até 3 dispositivos para alta disponibilidade.

O DNA Center pode operar com 2 tipos de redes:

- Traditional campus networks
- SD-Access fabric

Redes Tradicionais

O DNA Center pode funcionar com redes não SD-Access, semelhantes ao software de gerenciamento de rede tradicional. A automação orientada por política (Policy-driven automation) está disponível neste modo, mas é opcional. A funcionalidade de análise pode funcionar até mesmo com acesso somente leitura (read-only) à rede. Essa opção fornece uma maneira segura dos administradores se familiarizarem e avaliarem os recursos do DNA Center.

SD-Access Fabric

A Cisco introduziu um novo paradigma para redes corporativas de médio a grande porte, denominado ‘intent-based networking’ (rede baseada em intenção). Nessa arquitetura, um administrador comunica a intenção a controladora ou solicita “o que” ele deseja alcançar. Ele não precisa especificar instruções específicas para dispositivo para que as alterações sejam aplicadas. A controladora aceita instruções do administrador via GUI ou de uma aplicação via API e, em seguida, aplica a configuração aos dispositivos que ela controla.

O acesso definido por software ou SD-Access é uma implementação desse paradigma. O SD-Access tem vários protocolos subjacentes para fornecer infraestrutura de rede virtualizada escalonável e flexível, mas relativamente complexa. O DNA Center é a chave que esconde a complexidade ao fornecer um nível de abstração que permite que os operadores de rede concentrem sua atenção em conceitos de configuração de alto nível, como políticas.

Papel do DNA Center na rede

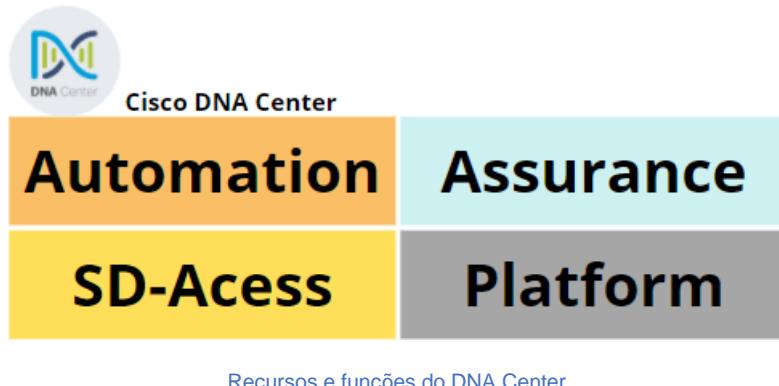
Com SD-Access Fabric, o DNA Center desempenha um papel essencial. Enquanto a rede underlay pode ser construída manualmente, as redes overlay são criadas e operadas através do DNA Center. Na rede tradicional, um administrador pode decidir quais tarefas devem ser realizadas pelo DNA Center e quais devem ser feitas diretamente no dispositivo.

O DNA Center possui capacidade de realizar diversas tarefas no ciclo de vida da rede:

- **Dia 0.** Integração e descoberta. Durante este estágio, o DNA Center pode ser usado para provisionamento zero-touch (ZTP) com protocolo Plug and Play (PnP);
- **Dia 1.** Provisionamento. Os modelos baseados em políticas podem ser definidos e aplicados a vários dispositivos agrupados em uma hierarquia de sites;
- **Dia 2, N.** Operação por meio de configuração de política, monitoramento, solução de problemas e correção de software. O DNA Center tem vários recursos que simplificam as tarefas de operações de rede, incluindo Software e Gerenciamento de Imagens.

Recursos e funções

As funções e recursos do DNA Center podem ser divididos em 4 grupos, conforme mostrado na figura abaixo:



Automation

Este grupo de recursos é responsável por executar tarefas operacionais e de provisionamento sem que seja necessário aplicar a configuração manualmente nos dispositivos. Veja alguns desses exemplos abaixo.

Network Design and Profiles

Separação lógica da rede em uma hierarquia de regiões e sites. Os perfis, que incluem parâmetros comuns, como DNS, servidores DHCP, são associados a esses contêineres lógicos, de forma que todos os sites sob eles herdam as configurações.

Software Image Management (SWIM)

Esse recurso garante que versões das imagens dos softwares sejam implantadas nos dispositivos na rede. O DNA Center realiza verificações antes e após a instalação. Por exemplo, verifica o espaço livre na memória flash, dentre outros.

Plug and Play de rede (PnP)

Um recurso muito útil quando o número de dispositivos a serem implementados na rede é alto. Com esse recurso, um dispositivo só precisa ser conectado a rede para receber um endereço IP e demais configurações automaticamente. Vários métodos de descoberta são suportados pelo DNAC, como o uso de DHCP option e DNS.

QoS Configuration Automation

Um dos aspectos desafiadores da operação diária da rede é a implementação das políticas de QoS. As aplicações na rede podem mudar ou novas aplicações podem ser adicionadas. Se a rede for gerenciada manualmente, manter a configuração atualizada com o tráfego classificado corretamente e com tratamento adequado pode consumir muito tempo. A implementação de QoS em dispositivos com modelos de hardware diferentes aumenta ainda mais a complexidade.

O DNA Center fornece interface de usuário intuitiva, que permite ao administrador selecionar as aplicações de acordo com modelos predefinidos, assim definindo se aquela aplicação é relevante para o negócio ou não.

Assurance

Outra função de uma controladora é fornecer monitoramento centralizado. O componente do DNA Center responsável por isso é chamado de DNA Assurance. Ele fornece recursos exclusivos, como a correlação de diferentes tipos de informações; focada em uma visualização 360° graus de todos os dispositivos da rede e clientes; além de visão retrospectiva com recurso de ‘viagem no tempo’ para descobrir o histórico de determinado dispositivo.

Dashboards

Existem vários painéis disponíveis, cada um focando diferentes aspectos da integridade da rede. O desempenho de aplicações relevantes para os negócios, clientes e dispositivos de rede são monitorados e os principais problemas são exibidos.

Device 360, Client 360 and Network Time Travel

Esses recursos fornecem uma visão centralizada do dispositivo ou cliente. Ele fornece ao administrador a capacidade de acessar rapidamente informações relevantes de um endpoint ou dispositivo e sua integridade. Por exemplo, ele simplifica a solução de problemas quando um usuário reclama sobre o desempenho de uma aplicação. Ao usar a função de pesquisa para localizar rapidamente o usuário e seu dispositivo, um administrador pode identificar se há problemas com a acessibilidade da rede, como sinal de RF fraco ou perdas de pacotes.

Geralmente, também há um ‘gap’ de tempo entre a ocorrência de um problema e a ação do administrador começar a trabalhar no incidente. Nesse momento, um alerta ou log pode ser apagado, tornando a solução do problema mais difícil. O ‘Network Time Travel’ permite a visualização do dispositivo em um momento específico no passado (até 14 dias) para ver eventos e alertas que estavam ativos naquele momento.

Path Trace

O Path Trace exibe visualmente cada dispositivo no caminho entre dois endereços IP na rede. Opcionalmente, pode incluir informações sobre dispositivos que podem estar bloqueando o tráfego com ‘access-list’, bem como estatísticas das interfaces e QoS (Quality of Service).

AI Network Analytics

A análise fornecida por algoritmos de Inteligência Artificial/Aprendizado de Máquina ajuda a identificar problemas de forma proativa. Primeiro, uma baseline da rede é coletada e então o aprendizado ocorre. Essas informações são então usadas para avaliar anomalias e alertar o administrador sobre um possível problema.

SD-Access

Este grupo de funções é específico para SD-Access. Inclui funções necessárias para executar os recursos de gerenciamento da controladora SD-Access, seja para fabric ou wireless fabric

Fabric Assurance

O DNA Center fornece recursos de monitoramento adicionais para fabric, como a correlação da fabric underlay e overlay, acessibilidade entre fabric edge.

Group-based Policy Configuration

O DNA Center se integra ao Cisco ISE (Identity Service Engine) para permitir o uso de políticas baseadas em identidade usando Cisco TrustSec. A configuração de política baseada em grupo permite que um administrador configure o gerenciamento de grupo e política a partir da interface do usuário no DNA Center, que então se comunica com o ISE e a fabric.

O objetivo principal desse recurso é permitir que os dispositivos de rede deduzam a identidade do usuário sem depender do endereço IP ou do mapeamento de informações de VLAN. Por exemplo, quando um usuário ou um dispositivo é autenticado com o ISE, o tráfego deste dispositivo é marcado com uma tag especial chamada SGT (Security / Scalable Group Tag). O SD-Access coloca essa tag no cabeçalho VXLAN, para que outros dispositivos possam dizer a qual usuário ou grupo este datagrama pertence e a partir daí usar essas informações para aplicar as políticas de segurança e QoS.

Platform

Já introduzimos o conceito de protocolos Southbound e como eles são utilizados no SDN (Software Defined Networks), para ‘conectar’ a controladora aos dispositivos finais, como switches e roteadores. Protocolos Northbound, como o nome sugere, trabalha na direção oposta, e são responsáveis pela comunicação dos serviços externos para a controladora, realizando a integração de terceiros. O Cisco DNA Center suporta APIs REST (Representational State Transfer) para tal integração

Integration with Service Management Platforms

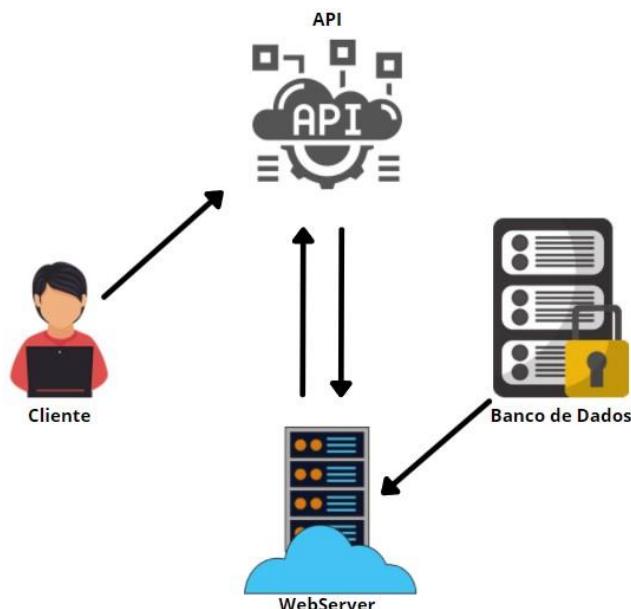
O Cisco DNA Center pode ser integrado via API com Service Management Platforms (Plataformas de gerenciamento de serviços). Essa integração fornece a capacidade de interagir com plataformas como ServiceNow. Por exemplo, o recurso Software Imaging (Criação de Imagens de Software) do DNA Center pode registrar uma solicitação de alteração no ServiceNow e realizar o envio de imagem apenas depois de aprovado. Outro caso de uso pode ser o registro automatizado de tíquetes (abertura de chamados) quando o DNA Center descobre\encontra algum problema.

IPAM Integration

IPAM (IP Address Management - gerenciamento de endereço IP) fornece gerenciamento centralizado de alocação de pools de endereços IP. A integração com esse sistema permite que o DNA Center reserve pools de acordo com o fluxo de trabalho da organização.

6.5 Describe characteristics of REST-based APIs (CRUD, HTTP verbs, and data encoding)

Hoje em dia, quase todas as aplicações na Internet exigem interoperabilidade como um recurso basilar. Em todo momento, as aplicações estão colaborando com outras aplicações (por exemplo, uma aplicação móvel se comunicando com um aplicação web). Assim, é essencial que todos essas aplicações sejam capazes de se comunicar entre si, independente do sistema operacional e de linguagens de programação. Os ‘Web Service’ são usados para formar essas ligações.



Web Service

Um webservice é um grupo de padrões e protocolos usados por aplicativos e sistemas para trocar informações na Internet. O Web service é independente do sistema operacional e pode ser escrito em qualquer linguagem de programação.

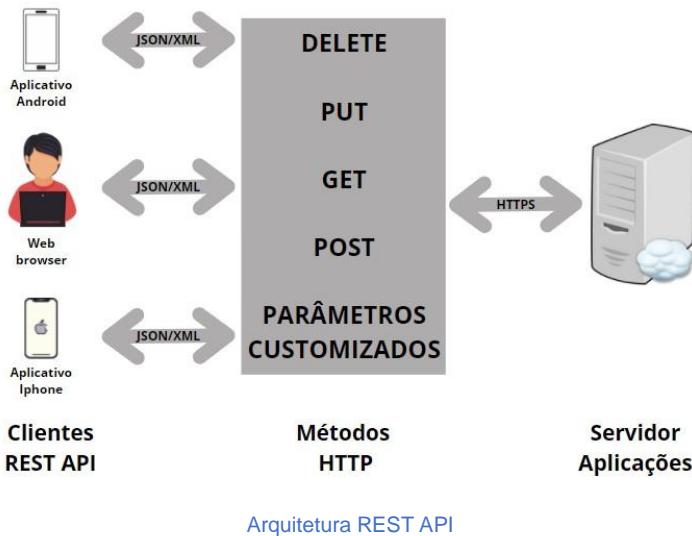
Por exemplo, um aplicativo construído em PHP executado em um servidor Linux pode se comunicar com o aplicativo Android que está sendo executado em um sistema operacional Android.

REST:

Rest é a abreviação de ‘Representational State Transfer’ (Transferência de Estado Representacional). REST é uma arquitetura de software que fornece várias características e protocolos subjacentes, controlando o comportamento de clientes e servidores.

REST API:

Podemos afirmar que ‘REST API’ pode ser usada por qualquer aplicativo, independentemente da linguagem em que está escrita, isto porque as solicitações são baseadas no protocolo HTTP que é universal e os dados são normalmente retornados no formato JSON para que possam ser lidos por quase todas as linguagens de programação.



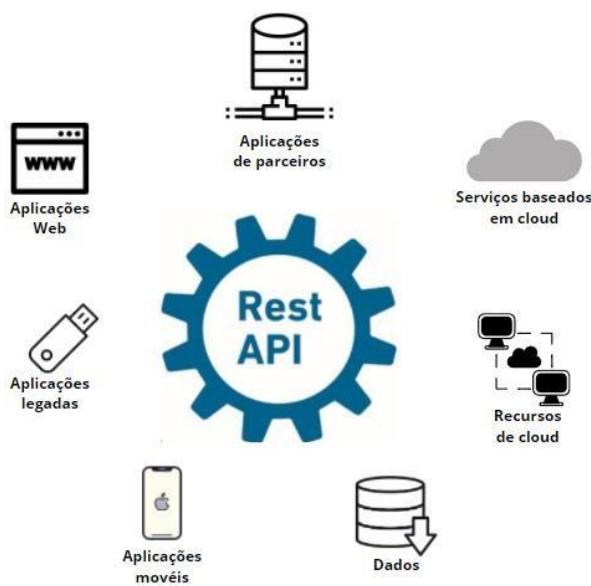
Uma API é considerada RESTful se contiver os seguintes recursos:

- **Arquitetura servidor-cliente:** O servidor é o back-end e o cliente é o front-end do serviço. É significativo notar que essas duas entidades são independentes uma da outra;
- **Stateless (Sem estado):** Nenhum dado ou informação deve ser armazenado no servidor durante o processamento da transmissão solicitada. O estado da sessão deve ser salvo na extremidade do lado do cliente;
- **Cacheable:** O cliente possui a capacidade de armazenar respostas em um cache. Isso aumenta significativamente o desempenho da API;
- **Isolamento:** o cliente é isolado no caminho da solicitação;
- **Idempotence:** Solicitações idênticas não tem nenhum efeito colateral

O que significa API RESTful?

Uma API RESTful é um ‘web service’ que é implementado usando o protocolo HTTP e os princípios REST. É uma coleção de recursos que serve aos métodos HTTP (PUT, GET, POST, DELETE).

A coleção de recursos é então representada em uma forma padronizada (geralmente XML) que pode ser qualquer tipo de mídia na Internet, desde que seja um padrão de hipertexto válido.



Por que devemos usar a API RESTful?

Uma API RESTful é usada para tornar as aplicações independentes, com objetivo de aprimorar o desempenho, simplicidade, escalabilidade, visibilidade, modicabilidade, confiabilidade e portabilidade do aplicativo.

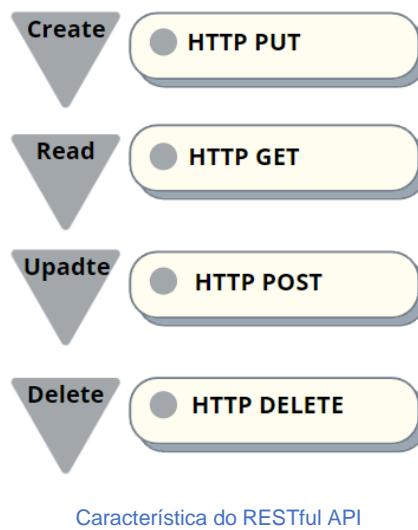
Exemplos de API RESTful no mundo real:

Todos os sites populares e plataformas de mídia social oferecem API RESTful. Dentre os mais diversos exemplos, temos:

- Twitter REST API
- Cloudways REST API
- Google Translate REST API
- Facebook REST API
- Magento REST API

CRUD

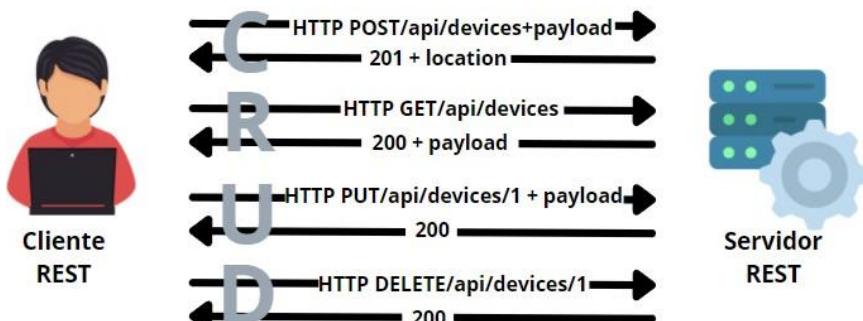
REST é uma API que permite aos clientes realizarem operações de leitura/gravação de dados de informações armazenadas em um servidor. REST utiliza HTTP para realizar um conjunto de ações comumente conhecido como CRUD, que significa:



Supondo que desejamos manipular um objeto ‘dispositivo’ em um servidor, podemos enviar uma solicitação HTTP GET para `/api/devices` e obter uma resposta com payload contendo uma lista completa de todos dispositivos conhecidos.

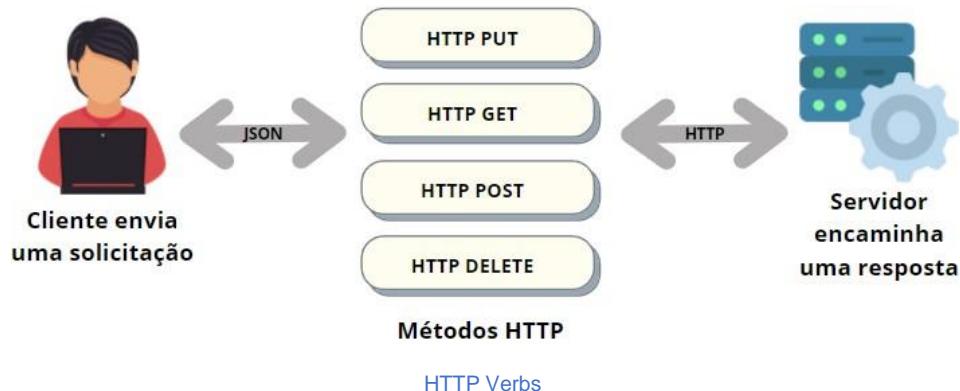
Se precisarmos adicionar um novo dispositivo, precisamos construir um payload com os atributos do dispositivo (por exemplo, endereço IP, nome do host) e enviá-lo anexado a uma solicitação HTTP POST.

Para atualizar um dispositivo, precisamos enviar um payload atualizado e completo com o método HTTP PUT.



Lembre-se de que as atualização e exclusão da ‘API calls’ de referem-se a um número específico em `url/api/devices/{ID}`. Esse é um **UUID** que o servidor atribui a cada novo objeto, e é retornado no cabeçalho ‘Location’ da mensagem ‘**201 Created**’ enviada em resposta à solicitação Create.

HTTP Verbs



Protocolo HTTP

Se você já usou a Internet, com certeza já possui uma noção de como o HTTP funciona. Ele envia solicitações (request) da aplicação que está rodando no seu desktop e recebe informações de servidores remotos. Essa é a internet em poucas palavras. A Internet só é viável porque todos os computadores que usam a rede, falam a mesma língua, e o mesmo protocolo: HTTP.

A Internet é baseada no protocolo HTTP. Ele permite que computadores de qualquer lugar enviem solicitações a servidores remotos e recebam respostas que podem ser exibidas em navegadores.

Métodos HTTP

A seguir estão os quatro principais métodos HTTP:

HTTP GET: Usamos o método GET para recuperar dados de um servidor remoto. Pode ser um recurso ou uma lista de recursos. Sem que for enviado API HTTP GET, se o recurso for encontrado no servidor, ele retornará o código de resposta HTTP 200 (OK) - junto com o corpo da resposta, que geralmente é um código XML ou JSON.

Caso o recurso NÃO seja encontrado em um servidor, ele deve retornar o popular código de resposta HTTP 404 (Not Found).

Exemplos de URIs de solicitação:

HTTP GET `http://www.appdomain.com/users`

HTTP GET `http://www.appdomain.com/users?size=20&page=5`

HTTP GET `http://www.appdomain.com/users/ 123`

HTTP GET `http://www.appdomain.com/users/123/address`

HTTP POST:

Usamos o método POST para criar um novo recurso no servidor remoto. De preferência, se um recurso for criado no servidor de origem, a resposta deve ser o código de resposta HTTP 201 (created) e conter uma entidade que descreva o status da solicitação e o novo recurso, além de um cabeçalho de localização.

Exemplos de URIs de solicitação

HTTP POST <http://www.appdomain.com/users>

HTTP POST <http://www.appdomain.com/users/123/accounts>

HTTP PUT:

Usamos o método PUT para atualizar os dados no servidor remoto. Caso uma nova fonte tenha sido criada pela API PUT, o servidor de origem DEVE informar o agente do usuário através do código de resposta HTTP 201 (created) e se um recurso existente for modificado, ele deve enviar os códigos 200 (OK) ou 204 (no content). Os códigos de resposta devem ser enviados para especificar a conclusão bem-sucedida da solicitação.

Exemplos de URIs de solicitação:

HTTP PUT <http://www.appdomain.com/users/123>

HTTP PUT <http://www.appdomain.com/users/123/accounts/456>

6.6 Recognize the capabilities of configuration management mechanisms Puppet, Chef, and Ansible

Servidores e dispositivos de rede não permanecem do mesmo jeito depois que os instalamos na rede, eles estão sempre em mudança, pois, instalamos atualizações nos sistemas operacionais e pacotes de segurança, fazemos alterações nas configurações. Imagine que precisemos instalar uma atualização para 100 ou 1000 servidores ou criar uma nova VLAN em 20 switches. Esses são os tipos de tarefas que ninguém deseja fazer manualmente, caixa por caixa. É um processo demorado, cansativo e por isso, propenso a falhas e erros do operador.

As ferramentas de gerenciamento de configuração oferecem um método automatizado para implementar e monitorar mudanças nos sistemas. Esses sistemas podem incluir servidores, Storages, dispositivos de rede e software. **O objetivo é manter esses sistemas em estados conhecidos e determinados.** Definimos o estado necessário em uma configuração e a ferramenta de gerenciamento de configuração usa automação para corresponder a esse estado nos sistemas de destino.

O gerenciamento da configuração é importante porque nos permite dimensionar a infraestrutura e o software sem ter que aumentar a equipe para gerenciar esses sistemas.

Features (recursos)

Em um alto nível, as ferramentas de gerenciamento de configuração oferecem os seguintes recursos:

- Execução
- Cooperação
- Forma de controle amigável
- Controle de mudança
- Abstração

Vamos entender cada um desses recursos em detalhes.

Execução:

Ao usar uma ferramenta para configurar os dispositivos, garantimos que o dispositivo está configurado para o estado desejado. Isso evita desvios de configuração. O desvio de configuração ocorre quando as pessoas instalam pacotes manualmente ou alteram arquivos de configuração. A variação na configuração torna a solução de problemas demorada e difícil.

Cooperação:

Armazenar configurações em uma ferramenta de gerenciamento de configuração torna mais fácil cooperar com outras pessoas. Ter todos os arquivos de configuração em um único lugar torna mais fácil implementá-los em todos os sistemas necessários e compartilhá-los.

Forma de controle amigável:

Ter todos os arquivos de configuração em um único lugar significa que podemos colocá-los sob controle de versão em um Version Control System (Sistema de Controle de Versão (VCS)) como o Git. Isso é ótimo para cooperação, pois várias pessoas podem trabalhar juntas nos mesmos arquivos. Todos podem ver quem adicionou ou editou quais arquivos e quando.

Controle de mudança:

Como as ferramentas de gerenciamento de configuração são baseadas em texto, podemos colocá-las sob controle de versão. Com um VCS, podemos ver facilmente as mudanças entre os arquivos de configuração, o que torna a revisão de código simples. Isso torna mais fácil decidir quais alterações enviar para a rede de produção.

Abstração:

Ao utilizar Linux, você terá diversas distribuições diferentes como Ubuntu e CentOS para escolher. Existem diferenças entre as distribuições. A instalação de um pacote requer comandos diferentes e a localização dos arquivos de configuração pode ser diferente. Por exemplo, o webserver apache usa diferentes arquivos e pastas no CentOS ou Ubuntu. As ferramentas de gerenciamento de configuração abstraem esses itens específicos do sistema operacional para que você possa usar os mesmos arquivos de configuração sem se preocupar com o sistema operacional subjacente.

Agent vs Agentless

Existem dois tipos de ferramentas de gerenciamento de configuração:

- **Agent based tools (ferramentas baseadas em agente)**
- **Agentless tools (ferramentas sem agente)**

Agent based tools requerem a instalação de um agente no sistema que você deseja gerenciar. Agentless tools não requerem um agente ou software no sistema que você deseja gerenciar.

Puppet e Chef são dois exemplos de agent based tools. O Ansible é uma agentless tools.

Puppet

Puppet é uma ferramenta de gerenciamento de configuração usada para implementar, configurar e gerenciar dispositivos. Ele utiliza arquitetura mestre-escravo. Existe um servidor mestre Puppet, e os clientes executam um agente Puppet, que extrai a configuração do mestre.

O Puppet utiliza uma linguagem própria de configuração, chamada puppet DSL.

Chef

Chef é uma plataforma de automação que configura e gerencia infraestrutura. Ele usa uma arquitetura mestre-escravo, semelhante ao Puppet. O servidor Chef gerencia os hosts e armazena as configurações de cada um desses hosts. Cada host executa um ‘client’ chef e extrai tarefas de configuração do servidor.

Chef usa Ruby DSL para arquivos de configuração.

Ansible

Ansible é uma ferramenta de configuração e orquestração, escrita em Python que usa YAML para tarefas de configuração. Chamamos essas tarefas de “playbook”. O Ansible não trabalha com agentes, ele usa SSH para se conectar aos dispositivos. Ele envia programas chamados ‘Ansible modules’, os executa e remove quando concluídos.

Ansible é uma ótima maneira de começar a automação de rede. Os playbook são fáceis de ler e escrever e, por não necessitar da instalação de um agente (é o único agentless tools), não temos o trabalho de instalá-lo em um servidor e depois instalar agentes nos hosts.

6.7 Interpret JSON encoded data

JSON (JavaScript Object Notation) é um formato de dados amigável para humanos, usado por aplicativos para armazenamento, transferência e leitura de dados. É fácil de analisar e pode ser usado com a maioria das linguagens de programação modernas, incluindo Python.

Algumas características básicas do **JSON** são:

- JSON é texto simples;
- É "auto-descritivo" (legível);
- É hierárquico (valores dentro de valores);
- Pode ser analisado pelo JavaScript;
- Os dados podem ser transportados usando AJAX

A sintaxe JSON é um subconjunto da linguagem JavaScript (sendo independente desta).

As principais regras de sintaxe JSON são:

- Dados JSON estão definidos aos pares no formato: nome : valor
- Os dados são separados por vírgulas (,)
- As chaves {} contém objetos
- Os colchetes [] expressam matrizes/vetores

Basicamente o JSON se baseia na notação NOME : VALOR, onde NOME pode ser o nome que você deseja usar para identificar um objeto e VALOR o valor deste objeto.

Exemplificando: A sintaxe JSON usa o par NOME : VALOR onde nome é definido entre aspas, seguido por dois pontos, seguido por um valor: Ex: "nome" : "Luiz" , "font-size" : "14px;"

Em JSON os valores usados podem ser:

- Um número (inteiro ou ponto flutuante)
- Uma string (entre aspas)
- Um booleano (verdadeiro ou falso)
- Uma matriz (entre colchetes [])
- Um objeto (entre chaves {})
- Nulo

Os objetos JSON são definidos entre chaves {} e podem conter múltiplos pares nome:valor:

Ex:

```
var pessoa = { "nome" : "Luiz" , "sobrenome" : "Silverio" };
var produto = {"ProdutoID":1, "Descricao":"CCNA", "ProdutoNumero":"200-301"};
```

Ex:

Os arrays em JSON são definidos entre colchetes [] e podem conter múltiplos objetos:

```
var cores = [ "Azul" , "Branco", "Vermelho", "Amarelo" ];
```

Vamos para mais um exemplo:

```
{
  "Nome": "Luiz",
```

```
"Idade": 40,  
"Certificacoes": ["CCNA", "CCNP", "CCIE"]  
}
```

No exemplo acima, “certificacoes” é uma matriz que contém três valores “CCNA”, “CCNP” e “CCIE”.

Se tivermos uma string JSON, podemos convertê-la (parse) para Python usando o método json.loads(), que retorna um dicionário Python:

```
import json  
  
minhavar = '{"nome":"Luiz","idade":40,"certificacoes":["CCNA", "CCNP", "CCIE"]}'  
  
parse_minhavar = json.loads(minhavar)  
  
print(parse_minhavar["certificacoes"][0])
```

O resultado:

```
CCNA
```

O zero dentro do argumento modificador de saída distingue qual parte da matriz [CCNA, CCNP, CCIE] será mostrada. A matriz posiciona da seguinte forma:

- CCNA = posição 0
- CCNP = posição 1
- CCIE = posição 2

Portanto, se você alterar o zero na última linha, para qualquer um dos números acima, receberá a saída correspondente.

Note que os conhecimentos exigidos sobre programação são básicos, a ideia do CCNA não é que você se torne um programador (até porque existem certificações específicas para isso), mas sim, que você tenha uma noção de programação.

Exercícios

1. Quais os dois principais benefícios da automação?
 - a) Reduz custos operacionais
 - b) Mudanças mais rápidas com resultados mais confiáveis
 - c) Reduz falhas na rede
 - d) Aumenta a segurança da rede
2. Quais as duas APIs abaixo são southbound?
 - a) CORBA
 - b) DSC
 - c) OpenFlow
 - d) NETCONF
 - e) Thrift
3. Quais são as duas características de uma rede baseada em controladora (controller-based network)?
 - a) O administrador pode fazer atualizações na configuração através da CLI.
 - b) Utiliza APIs para o northbound e southbound para se comunicar entre as camadas.
 - c) Move o plano de controle para um ponto central.
 - d) Descentraliza o plano de controle, o que permite que cada dispositivo tome suas próprias decisões de encaminhamento.
 - e) Utiliza Telnet para relatar problemas no sistema.
4. Quais são os dois métodos suportados pelas APIs REST?
 - a) YAML
 - b) JSON
 - c) EBCDIC
 - d) SGML
 - e) XML
5. Uma organização decidiu começar a usar serviços fornecidos pela nuvem. Qual serviço de nuvem permite que a organização instale seu próprio sistema operacional em uma máquina virtual?
 - a) platform-as-a-service
 - b) software-as-a-service
 - c) network-as-a-service
 - d) infrastructure-as-a-service
6. Qual afirmação compara corretamente as redes tradicionais e as redes baseadas em controladora?
 - a) Apenas as redes tradicionais oferecem um plano de controle centralizado.
 - b) Somente as redes tradicionais oferecem suporte nativo ao gerenciamento centralizado.
 - c) As redes tradicionais e baseadas em controladoras abstraem as políticas de configurações dos dispositivos.
 - d) Apenas as redes baseadas em controladoras separam o plano de controle e o plano de dados.
7. Qual API é usada em arquiteturas baseadas em controladoras para interagir com dispositivos de ponta?
 - a) overlay
 - b) southbound
 - c) underlay
 - d) northbound
8. Qual das opções a seguir é a codificação JSON de um dicionário ou hash?
 - a) {"key": "value"}
 - b) [{"key": "value"}]
 - c) {"key", "value"}
 - d) ("key": "value")
9. Qual a finalidade de uma API northbound em uma arquitetura de rede baseada em controladoras?
 - a) Realizar a comunicação entre a controladora e o hardware de rede
 - b) Relatar erros dos dispositivos para a controladora
 - c) Gerar estatísticas dos hardwares de rede e de tráfego
 - d) Facilitar a comunicação entre a controladora e as aplicações
10. Qual é uma característica da API REST?
 - a) Evoluiu para o que se tornou o SOAP
 - b) Usado para trocar informações XML estruturadas sobre HTTP ou SMTP
 - c) É considerado lento e complexo

- d) API mais amplamente utilizada para serviços da web
11. Qual plano de arquitetura definida por software auxilia os dispositivos de rede na tomada de decisões de encaminhamento de pacotes, fornecendo acessibilidade de Camada 2 e informações de roteamento de Camada 3?
- a) data plane
 - b) control plane
 - c) policy plane
 - d) management plane
12. Qual operação CRUD modifica uma tabela ou visualização existente?
- a) Read
 - b) Create
 - c) Replace
 - d) Update
13. Em arquiteturas definidas por software, qual plano é responsável pelo encaminhamento de tráfego?
- a) data plane
 - b) control plane
 - c) policy plane
 - d) management plane
14. Qual mecanismo de gerenciamento de configuração utiliza a porta TCP 22 por default ao se comunicar com hosts gerenciados?
- a) Python
 - b) Ansible
 - c) Puppet
 - d) Chef
15. Qual a função das controladoras wireless em uma rede corporativa?
- a) Centralizar o gerenciamento dos pontos de acesso da rede corporativa
 - b) suportar arquiteturas autônomas ou baseadas em controladoras
 - c) servir como a primeira linha de defesa da rede
 - d) fornecer logins de usuário para dispositivos na rede.
16. Qual operação CRUD corresponde ao método HTTP GET?
- a) Read
 - b) Update
 - c) Create
 - d) Delete
17. Qual plano de rede é centralizado e gerencia as decisões de roteamento?
- a) policy plane
 - b) management plane
 - c) control plane
 - d) data plane
18. Qual é o propósito de uma API southbound em uma arquitetura de rede baseada em controladora?
- a) Facilita a comunicação entre a controladora e os aplicativos
 - b) Facilitar a comunicação entre a controladora e o hardware de rede
 - c) Permitir que desenvolvedores de aplicações interajam com a rede
 - d) Integrar uma controladora com outras ferramentas de automação
19. Qual tecnologia é apropriada para a comunicação entre uma controladora SDN e aplicações em execução na rede?
- a) OpenFlow
 - b) REST API
 - c) NETCONF
 - d) Southbound API

Respostas: 1) a, b 2) c, d, 3) b, c, 4) b, e 5) d 6) d 7) d 8) a 9) d 10) d 11) b 12) d 13) a 14) b 15) a 16) a 17) c 18) b 19) b