***Maratona Backend do RoodMap***

**Rede de computadores | Certificado de suporte de TI do Google**

<https://www.youtube.com/watch?v=Z_hU2zm4_S8>

Na carreira de TI todos nós temos uma história do porque começamos e o que realmente nos insetivou a começar essa careira que hoje em dia é uma careira tão importante na vida das pessoas.

Porém, além de insentivo e uma história própria, é necessário como em qualquer outra careira botarmos a mão na massa e nos especializarmos naquilo que realmente importa... e na TI para começo de conversa, é necessário entendermos como as redes de internet funcionam, e como os computadores conversam entre si, então entenderemos o que são **Bases Numéricas**.

***Bases Númericas***

**Bases Númericas** ou **Sistemas de numeração**, são diferentes maneiras de representar os símbolos que utilizamos para representar números, e executar a contagem destes.

Por padrão usamos no dia á dia o **Sistema de Base Decimal** ou **Base 10**, onde a contagem dos números são de **0** até **9**, sendo representado **10** símbolos.

Porém no mundo da tecnologia utilizaremos outras bases como, **Base Hexadecimal**, **Base Octal** e principalmente a **Base Binária**, que é a base númerica que o computador entende, representada por **2** símbolos, sendo o **0** e **1**.

**Base Decimal:**

**178910 =** 1 **x** 10 **+** 7 **x** 10**^**2 **+** 8 **x** 10**^**1 **+** 9 **x** 10**^**0

**1000 + 700 + 80 + 9**

**Base Binária:**

**0** **=** 0 **11 =** 1011 **22** = 10110

**1 =** 1 **12 =** 1100 **23 =** 10111

**2 =** 10 **13 =** 1101 **24 =** 11000

**3 =** 11 **14 =** 1110 **25 =** 11001

**4 =** 100 **15 =** 1111 **26 =** 11010

**5 =** 101 **16 =** 10000 **27 =** 11011

**6 =** 101 **17 =** 10001 **28 =** 11100

**7 =** 111 **18 =** 10010 **29 =** 11101

**8 =** 1000 **19** **=** 10011 **30 =** 11110

**9 =** 1001 **20 =** 10100 **31 =** 11111

**10** **=** 1010 **21** **=** 10101 **32 =** 100000

**Binário para Decimal:**

**2710 = 11011 = 1 1 0 1 1**

2**^**4 2^3 2^2 2^12^0

**(** 1 **x** 2**^**4 **)** **+ (** 1 **x** 2**^**3 **)** **+ (** 0 **x** 2**^**2 **)** **+** **(** 1 **x** 2**^**1 **)** **+ (** 1 **x** 2**^**0 **)**

**(** 1 **x** 16 **)** **+ (** 1 **x** 8 **)** **+ (** 0 **x** 4 **)** **+ (** 2 **x** 2 **) + (** 1 **x** 1 **)**

16 **+** 8 **+** 0 **+** 2 **+** 1

**2710**

**Decimal para Binário:**

**1º Maneira:**

**75510 =** 755 **/** 2

**1 |** 377 **/** 2

**1 |** 188 **/** 2

**0 |** 94 **/** 2

**0 |** 47 **/** 2

**1 |** 23 **/** 2

**1 |** 11 **/** 2

**1 |** 5 **/** 2

**1 |** 2 **/** 2

**0 |** 1 **/** 2

**1 | 0**

**10111100112**

**2º Maneira:**

**75510 =** 1024 512 256 128 64 32 16 8 4 2 1

**0 1 0 1 1 1 1 0 0 1 1**

**10111100112**

**Decimal para Octal:**

**37110 =** 3 7 1

8^2 8^1 8^0

**(** 3 **x** 8**^**2 **) + (** 7 **x** 8**^**1 **) + (** 1 **x** 8**^**0 **)**

**(** 3 **x** 64 **) + (** 7 **x** 8 **) + (** 1 **x** 1 **)**

192 **+** 56 **+** 1

**24910**

**Hexedecimal para Decimal:**

**1 2 3 4 5 6 7 8 9 A B**

**C D E F G**

**1FA16 =** 1 F A

16^2 16^1 16^0

**(** 1 **x** 16**^**2 **) + (** 15 **x** 16**^**1 **) + (** 11 **x** 16**^**0 **)**

256 **+** 240 **+** 1

**49710**

**Decimal para Octal:**

**17710 =** 177 **/** 8 **=** 22

22 **x** 8 **=** 176

177 **–** 176 **=** 1

22 **/** 8 **=** 6

6 **x** 8 **=** 48

22 – 48

**Decimal para Hexadecimal:**

**68510 =** 685 **/** 16

D **|** 42 **/** 16

A **|** 2 **/** 16

2 **|** 0

**2AD16**

**Octal para Binário:**

**278 =** 2 7

0 1 0 1 1 1

**4 2 1 4 2 1**

**0101112**

**Binário para Octal:**

**1111011012 =** 1 1 1 1 0 1 1 0 1

**4 2 1 4 2 1 4 2 1**

**7 5 5**

**7558**

**Hexadecimal para Binário:**

**1B16 =** 1 B

0 0 0 0 1 0 1 1

**8 4 2 1 8 4 2 1**

00011011

**110112**

**Binário para Hexadecimal:**

**1101112 =** 0 0 1 1 0 1 1 1

**8 4 2 1 8 4 2 1**

**2 + 1 4 + 2 + 1**

**3 7**

**3716**

***Modelo OSI***

O Modelo OSI (Open Systems Interconnection) e um framework de referência para a comunicação em redes de computadores, criado pela ISO (International Organization for Standardization).

Esse modelo organiza as funções de uma rede de comunicação em sete camadas distintas, cada uma responsável por processo específico de comunicação, desde a transmissão física de dados até a interação final com o usuário.

O modelo OSI ajuda a padronizar e facilitar a comunicação entre diferentes sistemas e tecnologias de rede, como o TCP/IP que iremos também abordar neste maratona. Além disso resolver os problemas na comunicação se torna um processo menos árduo, pois o processo de comunicação é divido em camadas, assim a especialização de cada camada se torna mais fácil.

O ponto negativo é implementar todas as camadas, pois isso acaba se tornando um processo complexo e caro; e nem sempre algumas redes modernas do Modelo TCP/IP seguem o processo do modelo OSI por completo. Alguns exemplos são:

1. **Internet.**
2. **Redes Privadas Virtuais (VPNs).**
3. **Redes de Data Centers.**
4. **Redes de IoT (Internet das Coisas).**
5. **Redes Móveis (4G/5G)**

**As Sete Camadas do Modelo OSI:**

1. **Camada Física.**
2. **Camada de Enlace de Dados.**
3. **Camada de Rede.**
4. **Camada de Transporte.**
5. **Camada de Sessão.**
6. **Camada de Apresentação:**
7. **Camada de Aplicação:**

**1 - Camada Física (Physical) :**

Esse protocolo retrata o lado físico no contexto de **Redes de Internet**, aqui nós vamos falar sobre os dados brutos, como cabos de cobre, fibras ópticas e sinais de rádio. É aonde será feita a transmissão de **Bits Brutos**.

Segue abaixo alguns exemplos de Protocolos e Padrões da Camada Física:

* **Ethernet**: Define como os dados são formatados e transmitidos por cabos de rede, ou cabos físicos.
* **Wi-Fi:** Protocolo para redes sem fio, que define como os dados são transmitidos usando ondas de rádio.
* **Bluetooth (IEEE 802.15.1)**: Protocolo para comunicação de curta distância usando ondas de rádio.
* **USB (Universal Serial Bus)**: Padrão para comunicação e fornecimento de energia entre computadores e dispositivos periféricos.
* **Fibras Ópticas**: Utiliza luz para transmissão de dados a longas distâncias com alta velocidade e baixa perda de sinal.

**2 - Camada de Enlace de Dados (Data Link):**

É nessa camada que definimos como os dados serão transmitidos e controlados. Aqui os dados são envelopados em unidades que chamamos de frames; fazemos o controle de fluxo; corrigimos os erros que podem ocorrer durante a transmissão dos dados; e fazemos o controle da transmissão, permitindo que cada dispositivo tenha acesso um de cada vez ao meio, evitando possíveis colisões.

Segue abaixo alguns exemplos de Protocolos e Padrões de Camada de Enlace de Dados:

* **Ethernet**: Um dos protocolos mais comuns para redes locais (LAN). Define como os dados são formatados e transmitidos através de cabos físicos.
* **Wi-Fi**: Protocolo utilizado para redes sem fio. Define como os dados são transmitidos através de ondas de rádio.
* **PPP (Point-to-Point Protocol)**: Utilizado para estabelecer uma conexão direta entre dois nós de rede, frequentemente em conexões de internet dial-up.
* **HDLC (High-Level Data Link Control)**: Um protocolo de controle de enlace que garante a entrega confiável de dados entre dois pontos.
* **Token Ring (IEEE 802.5)**: Um protocolo de enlace de dados em que os dispositivos na rede formam um anel lógico e um token circula para controlar o acesso ao meio.

**3 - Camada de Rede (Network):**

Essa camada é essencial na comunicação dos dados, pois aqui determinamos as melhores rotas para o envio dos dados, usando o protocolo **IP ( Internet Protocol )** para identificar cada dispositivo com um endereço específico, identificamos esses dados com um cabeçalho em seus respectivos pacotes ( frames ) que são envelopados na camada de enlace, e por fim aqui definimos a quantidade de dados que serão transmitidos, evitando assim congestionamentos na transmissão dos dados, e garantimos maior eficiência na transmissão.

Segue abaixo alguns exemplos de Protocolos e Padrões de Camada de Rede:

* **IP (Internet Protocol)**: Responsável por endereçar e encaminhar pacotes de dados entre dispositivos em diferentes redes. Existem duas versões amplamente utilizadas: **IPv4** e **IPv6**.
* **ICMP (Internet Control Message Protocol)**: Usado para enviar mensagens de erro e operacionais, como mensagens de destino inalcançável ou pedidos de eco (ping).
* **IGMP (Internet Group Management Protocol)**: Gerencia membros de grupos multicast, permitindo que dispositivos participem ou saiam de um grupo multicast.
* **RIP (Routing Information Protocol)**, **OSPF (Open Shortest Path First)**, **BGP (Border Gateway Protocol)**: Protocolos de roteamento que determinam o caminho que os dados devem seguir na rede.

**4 - Camada de Transporte (Transport):**

Essa camada é responsável pela chegada dos dados em seus respectivos endereços de entrega, de forma eficiente e confiável. Aqui faremos mensão de dois protocolos principais: o **TCP ( Transmission Control Protocol )** e o **UDP ( User Datagram Protocol )**; e dois protocolos adicionais: **TLS ( Transport Layer Security ) e o SCTP (Stream Control Transmission Protocol).**

Aqui os dados são divididos em segmentos menores e depois ao chegar no local de destino são reagrupados novamente, garantindo assim uma melhor transmissão, permitindo também que vários dispositivos utilizem a mesma conexão ao mesmo tempo, identificando seus fluxos por meio de portas.

Segue abaixo alguns exemplos de Protocolos e Padrões de Camada de Transporte:

* **TCP ( Transmission Control Protocol ):** O TCP neste momento tem um papel extremamente importante, pois ele faz:
* O gerenciamento dos números em sequência garantindo a entrega correta dos dados.
* Ajusta a quantidade de dados a serem enviados evitando sobrecarga, fazendo além disso controle de congestionamento na rede.
* **UDP ( User Datagram Protocol ):** O UDP por sua vez também tem suas peculiaridades:
* Este faz transmissão de dados de maneira mais rápida, porém a entrega dos dados e sua respectiva organização não é garantida.
* Tem baixa latência, então faz entregas mais rápidas, geralmente é usado em vídeos ao vivo e jogos online.
* Ao contrário do TCP ele não faz uma conexão antecipada.
* Devido sua simplicidade evitamos maior número de sobrecarga na rede e possíveis congestionamentos.
* Os dados são enviados em pacotes chamados datagramas.

| **Característica** | **TCP (Transmission Control Protocol)** | **UDP (User Datagram Protocol)** |
| --- | --- | --- |
| Tipo de Protocolo | Orientado à conexão | Sem conexão |
| Confiabilidade | Alta (usa confirmações e retransmissões) | Baixa (não garante entrega) |
| Controle de Fluxo | Sim | Não |
| Controle de Congestionamento | Sim | Não |
| Latência | Maior devido ao controle de fluxo e confirmações | Menor, ideal para tempo real |
| Uso Comum | Transferência de arquivos, web, email | Streaming de vídeo/áudio, jogos online, VoIP |

* **TLS (Transport Layer Security)**: Adiciona segurança à comunicação da Camada de Transporte, oferecendo criptografia, integridade dos dados e autenticação. Embora operando geralmente na Camada de Aplicação, pode ser considerado parte da Camada de Transporte quando se discute segurança.
* **SCTP (Stream Control Transmission Protocol)**: Um protocolo mais moderno que combina características do TCP e UDP, oferecendo suporte para múltiplos fluxos de dados independentes dentro da mesma conexão.

**Características Herdadas do Protocolo UDP:**

* Usa o mesmo sistema de “datagramas” porém aqui chamamos de “mensagens”.

**Características do Protocolo TCP:**

* **Controle de congestionamento**: Tanto o SCTP quanto o TCP implementam algoritmos para ajustar a taxa de envio de dados com base nas condições da rede, prevenindo a congestão e garantindo uma transmissão mais eficiente.
* **Garantia de Entrega**: O SCTP, como o TCP, garante que os dados sejam entregues corretamente ao destino. Isso e feito através da retransmissão de pacotes perdidos e da confirmação da recepção.
* **Ordem de Dados**: O SCTP e o TCP mantêm a ordem dos dados, garantindo que os pacotes sejam entregues na sequência correta.
* **Controle de Fluxo**: Ambos os protocolos utilizam mecanismos para controlar a quantidade de dados que podem ser enviados antes de receber uma confirmação, evitando a sobrecarga da rede e do receptor.

**5 - Camada de Sessão (Session):**

E a camada responsável pelo tempo de comunicação e manutenção da comunicação entre dois dispositivos em uma rede, gerenciando assim a troca de dados de maneira organizada e no tempo correto, garantindo que a comunicação não seja interrompida no meio do caminho; ela só será finalizada no final da conversação.

Primeiro ela estabelece uma conexão entre os dois dispositivos, discutindo como será o método de conversação e se há autenticação de ambas as partes.

Após a conexão estar ativa ela acompanha o processo de comunicação gerando manutenção se necessário garantindo que a comunicação não seja interrompida. Para isso em certos momentos ela gera checkpoints; assim em casos de erro, o processo é retomado a partir dos checkpoints ao invés de ser necessário voltar do começo.

Aqui determinamos também se os dados podem ser transmitidos de ambos os lados simultaneamente ( full-duplex ), ou se eles só podem ser transmitidos de um lado de cada vez como em uma conversação civilizada ( half-duplex ).

Segue abaixo alguns exemplos de Protocolos e Padrões de Camada de Sessão:

* **SIP (Session Initiation Procol)**: Usado para administrar a comunicação em redes IP; usado em videoconferências, chamadas de voz e mensagens multímidia.
* **5060 (TCP)** Comunicação SIP não criptografada.

**5061 (TCP/TLS)** Comunicação SIP segura (criptografada via TLS).

* **RPC (Remote Procedure Call)**: Permite que um programa execute um procedimento em um computador remotamente como se estivesse sendo executado localmente.
* **NetBIOS (Network Basic Input/Output System)**: Proporciona serviços de sessão, como o estabelecimento e a manutenção de conexões.

**6 - Camada de Apresentação (Presentation):**

É a camada responsável por garantir que os dados sejam apresentados de forma compreensível e de forma segura para a próxima camada que é a camada de aplicação.

Aqui os dados são formatados de forma específica... Os caracteres EBCDIC são convertidos para ASCII, ou vice-versa; e também é feita a compressão de imagens para JPEG, e vídeos/áudios para MPEG por exemplo.

Aqui os dados são criptografados e quando chegam no seu local de destino voltam ao formato original.

Segue abaixo alguns exemplos de Protocolos e Padrões de Camada de Apresentação:

* **SSL/TLS (Secure Sockets Layer / Transport Layer Security)**: Protocolo de criptografia usado para assegurar a comunicação segura pela rede, especialmente na web (HTTPS).
* **MIME (Multipurpose Internet Mail Extensions)**: Padrão que permite o envio de diferentes tipos de dados (como texto, áudio, vídeo, imagens) através de e-mail.
* **JPEG (Joint Photographic Experts Group)**: Padrão de compressão para imagens digitais.
* **MPEG (Moving Picture Experts Group)**: Padrão de compressão para áudio e vídeo.
* **ASCII (American Standard Code for Information Interchange) e EBCDIC (Extended Binary Coded Decimal Interchange Code)**: Conjuntos de códigos para a representação de texto.
* **GIF (Graphics Interchange Format)**: Formato de imagem que suporta animações e compressão sem perdas.
* **PNG (Portable Network Graphics)**: Formato de imagem que suporta compressão sem perdas e transparência.
* **XML (Extensible Markup Language) e JSON (JavaScript Object Notation)**: Formatos de dados usados para a troca de informações estruturadas entre sistemas.

### 7 - Camada de Aplicação (Application):

Está e a camada responsável por fornecer serviços de rede diretamente às aplicações do usuário final. Esta camada facilita a comunicação entre software de aplicação e outros sistemas de rede, permitindo que os dados sejam trocados de maneira eficiente e segura.

É neste momento que proporcionamos uma interface aos usuário final, gerenciamos e estabelecemos conexões entre diferentes dispositivos, e implementamos mecanismos de autentificação e criptografia aos serviços de rede como Transferência de Arquivos, Correio Eletrônico, Navegação Web e outros...

Abaixo seguem os protocolos principais da cama de Aplicação:

* **HTTP (Hypertext Transfer Protocol)**:
* **Função**: Usado para transferir páginas web.
* **Porta Padrão**: 80 (HTTP) e 443 (HTTPS para HTTP seguro).
* **FTP (File Transfer Protocol)**:
* **Função**: Permite a transferência de arquivos entre sistemas.
* **Porta Padrão**: 21.
* **SMTP (Simple Mail Transfer Protocol)**:
* **Função**: Usado para envio de emails.
* **Porta Padrão**: 25.
* **IMAP (Internet Message Access Protocol)**:
* **Função**: Permite o acesso e gerenciamento de emails em um servidor.
* **Porta Padrão**: 143 (IMAP) e 993 (IMAP seguro).
* **POP3 (Post Office Protocol version 3)**:
* **Função**: Utilizado para baixar emails do servidor para o cliente.
* **Porta Padrão**: 110 (POP3) e 995 (POP3 seguro).
* **DNS (Domain Name System)**:
* **Função**: Converte nomes de domínio em endereços IP.
* **Porta Padrão**: 53.
* **Telnet**:
* **Função**: Protocolo de rede que permite comunicação bidirecional interativa de texto através de uma conexão TCP.
* **Porta Padrão**: 23.
* **SSH (Secure Shell)**:
* **Função**: Protocolo de rede para operação segura de serviços de rede sobre uma rede insegura.
* **Porta Padrão**: 22.
* **LDAP (Lightweight Directory Access Protocol)**:
* **Função**: Protocolo para acesso e manutenção de serviços de diretório distribuídos.
* **Porta Padrão**: 389.
* **DHCP (Dynamic Host Configuration Protocol)**:
* **Função**: Protocolo de rede que permite que dispositivos obtenham endereços IP automaticamente.
* **Porta Padrão**: 67 (servidor) e 68 (cliente).
* **SNMP (Simple Network Management Protocol)**:
* **Função**: Usado para monitorar e gerenciar dispositivos em redes, como roteadores, switches, servidores e outros equipamentos de rede. Ele permite que administradores de rede coletem dados de desempenho, configurem dispositivos e recebam alertas de falhas.
* **Porta Padrão: 161** para comunicação padrão (solicitações e respostas).

**162** para **trap messages** (notificações de eventos enviados pelos dispositivos gerenciados para o gerente de rede).

* **RTP (Real-time Transport Protocol)**:
* **Função**: Usado para transmitir chamadas de áudio, vídeo e transmissões ao vivo (streaming). Esse protocolo está sendo utilizado dentro do protocolo de transporte **UDP (User Datagram Protocol)**.
* **1024 - 65535 (UDP)** É transmitido neste intervalo de portas evitando conflitos ao serém iniciadas várias chamadas.
* **RTCP (RTP Control Protocol)**:
* **Função**: Usado para verificar as estátisticas de uma transmissão feita em **RTP (Real-time Transport Protocol)**, monitorando as perdas de pacotes, a latência da transmissão, ou seja quanto tempo os recursos de um IP estão demorando pra chegar no seu local de destino, e até mesmo o jitter que é a variação entre esse tempo de entrega.
* **1024 - 65535 + 1 (UDP)** É transmitido em neste intervalo de portas evitando conflitos ao serém iniciadas várias chamadas. Aqui é adicionado mais 1 número evitando conflitos também com o protocolo **RTP (Real-time Transport Procol)**.

### Modelo TCP/IP

O Modelo TCP/IP segue algumas particularidades do Modelo OSI. Por isso não será precisa entrar em tantos detalhes. Vamos exemplificar apenas suas diferenças; assim você verá que esse modelo é mais simples e melhor de ser implementado.

**Camada de Rede (Link Layer):**

* Combina as funções das camadas Física e de Enlace de Dados do modelo OSI.
* **Exemplos:** Ethernet, Wi-Fi.

**Camada de Internet (Internet Layer):**

* Equivale à Camada de Rede do modelo OSI.
* Principal protocolo: IP (Internet Protocol).

**Camada de Transporte (Transport Layer):**

* Equivale à Camada de Transporte do modelo OSI.
* Protocolos principais: TCP (Transmission Control Protocol) e UDP (User Datagram Protocol).

**Camada de Aplicação (Application Layer):**

* Combina as funções das camadas de Sessão, Apresentação e Aplicação do modelo OSI.
* Exemplos: HTTP, FTP, SMTP, DNS.

**Modelo OSI:**

1. **Aplicação:** O usuário envia o e-mail.
2. **Apresentação:** O conteúdo do e-mail e formatado e criptografado.
3. **Sessão:** Uma sessão de comunicação e estabelecida entre o cliente de e-mail e o servidor.
4. **Transporte:** O e-mail e segmentado e transportado de forma confiável.
5. **Rede:** Os pacotes de dados são roteados através da rede.
6. **Enlace de Dados:** Os dados são preparados para transmissão física.
7. **Física:** Os bits são transmitidos através do meio físico (cabos, etc.).

****No Modelo TCP/IP:****

1. **Aplicação:** O usuário envia o e-mail.
2. **Transporte:** O e-mail e segmentado e transportado de forma confiável (TCP).
3. **Internet:** Os pacotes de dados são roteados através da rede (IP).
4. **Rede:** Os dados são preparados e transmitidos através do meio físico (Ethernet).

****Roteadores e o Protocolo BGP:****

Os roteadores se encaixam na linha de dispositivos mais importantes quando se trata de tráfego de dados na internet. Pois eles são os responsáveis por enviar os dados e direcioná-los para o destino correto.

O protocolo que lida com todo esse trabalho é chamado de **BGP ( Border Gateway Protocol )**, que é o protocolo de gerenciamento de rotas da internet. Este mantém uma tabela com as rotas disponíveis na internet, trabalhando de forma dinâmica, e se adaptando a cada tipo de situação no envio dos dados quando os IP’s estão se comunicando.

### ****Cabo Ethernet (Cat5, Cat5e, Cat6, Cat6a, Cat7, Cat8)****

**Cat5:** Suporta velocidades de até 100 Mbps e e adequado para redes básicas. Menos comum hoje em dia.

**Cat5e (Category 5e):** Uma versão aprimorada do Cat5, suportando velocidades de até 1 Gbps e melhorando a redução de crosstalk e interferências.

**Cat6:** Suporta velocidades de até 10 Gbps a distâncias mais curtas (até 55 metros) e oferece melhor isolamento contra interferências.

**Cat6a (Category 6a):** Melhorado para suportar 10 Gbps a distâncias de até 100 metros e com melhor proteção contra interferências.

**Cat7:** Suporta 10 Gbps a até 100 metros e possui uma blindagem adicional para minimizar a interferência.

**Cat8:** Suporta velocidades de até 25-40 Gbps a distâncias de até 30 metros. Ideal para ambientes de data centers e aplicações de alta velocidade.

### ****Cabo de Par Trançado Não Blindado (UTP)****

**UTP (Unshielded Twisted Pair):** O tipo mais comum de cabo Ethernet, usado em redes domésticas e comerciais. Não possui blindagem adicional, tornando-o mais suscetível a interferências, mas e suficiente para a maioria das aplicações.

Os cabos Ethernet possuem quatro pares trançados de fios de cobre. Cada par e torcido para reduzir interferências e melhorar a qualidade da transmissão de dados.

### ****Cabo de Par Trançado Blindado (STP)****

**STP (Shielded Twisted Pair):** Semelhante ao UTP, mas com uma camada adicional de blindagem para proteger contra interferências eletromagnéticas. Usado em ambientes com altos níveis de interferência.

### ****Fibra Óptica:****

**Fibra Óptica Monomodo:**  Ideal para longas distâncias e altas velocidades. Utiliza um único fio de vidro ou plástico para transmitir sinais de luz, permitindo transmissões muito rápidas e sem interferências.

**Fibra Óptica Multimodo:** Usada para distâncias menores e geralmente em redes locais. Possui múltiplos fios de vidro ou plástico e é menos cara que a fibra monomodo, mas tem menor capacidade de largura de banda e maior perda de sinal em longas distâncias.

**Obs.:** Cada tipo de cabo tem suas próprias especificações e é escolhido com base nas necessidades de velocidade, distância e ambiente de instalação.

### ****Endereço MAC (Media Access Control)****

O endereço **MAC** é atribuído diretamente a placa de rede do dispositivo, dando ao dispositivo um identifcador para conexão em rede.

Esse endereço possui normalmente **48 bits** formado por 6 grupos de bits hexadecimais.

**Exemplo →** 00:1A:2B:3C:4D:5E **.**

Os três primeiros digitos indentificam o fabricante da placa de rede, e os 3 últimos digítos garantem a identicação do dispositivo, evitando assim, duplicidade de identifações do endereço e gerando **unicidade global**.

O endereço MAC em si é acompanhado com protocolo **ARP (Adress Resolution Protocol)** que é o protocolo que faz a intermediação entre a rede LAN com o endereço MAC dos dispositivos. O ARP basicamente ao se conectar com o IP pergunta para a rede em qual endereço MAC conectado naquele IP haverá a transferência de arquivos. Então o dispositivo por meio do IP responde, e a conexão é estabelecida.

### ****Tipos de comunicações em Redes: Simplex, Half-Duplex e Full-Duplex.****

Neste tópico falaremos um pouco sobre a transmissão dos dados quando chegamos na Camada 1 do Modelo OSI, **Camada Física (Physical)**.

O processo de comunicação pode ocorrer de três maneiras diferentes, chamadas de **Simplex, Half-Duplex e Full-Duplex**.

### ****Simplex (Comunicação Simples):****

Essa é a comunicação mais básica, pois acontece apenas em um sentido de transmissão, onde um dispositivo sempre envia os dados e outro apenas recebe. O dispositivo que recebe nunca dará uma resposta ao seu remetente.

Um exemplo clássico seria a televisão, onde a emissora de TV envia os dados do programa que está sendo transmistido e sua TV em casa apenas recebe esses dados.

### ****Half-Duplex (Comunicação Semi-Duplex):****

Na comunicação **Half-Duplex** o processo é um pouco mais robusto, pois ambos os dispositivos podem enviar e receber dados, porém não simultaneamente. O dispositivo que estiver recebendo informações deve esperar o seu remente terminar o envio para enviar uma resposta.

Para evitar esse tipo de colisão antes de chegar na **Camada Física (Physical)**, após os dados serém empacotados na **Camada de Enlace de Dados (Data Link)**,é utilizado dois protocolos na Camada de Enlace para evitar essas colisões, controlando o acesso ao meio. Chamamos esse protocolos de **CSMA/CD** e **CSMA/CA**.

**CSMA/CD →** O **CSMA/CD** é utilizado no protocolo **Ethernet (Com fio)**.

Nos dias atuais é praticamente obsoleto pois é usado em hubs, sendo que as redes mais modernas é utilizado mais switches.

Ele escuta o meio da transmissão para verificar se alguém está enviando alguma informação, permitindo o ínicio de uma nova transmissão apenas se o meio estiver livre. Caso ocorra alguma colisão a transmissão é interrompida, sendo enviado um sinal de interferência chamado **jan signal**. Após isso, é esperado um tempo aleatório, calculado através de um algorotimo chamado **backoff exponencial**.

Funciona como se duas pessoas falando ao mesmo tempo. Elas percebem que não estão conseguindo, então param e iniciam a conversa novamente, dando sequência na conversa sem interrupções enquanto uma estiver falando.

**CSMA/CA →** Diferente do **CSMA/CD** o **CSMA/CA** é utilizado no protocolo **Wi-Fi (Sem fio)**, então por muitas das vezes o sinal não chega a todos os dispositivos, ocorrendo o que chamamos de **nó escondido**. Aqui é priorizado evitar que a colisão ocorra.

O dispositivo escuta o meio para ver se está livre. Caso estiver ocupado, ele espera um tempo chamado de **IFS – Interframe Space**.

Em alguns casos... quando um dispositivo vê que o meio está ocupado, este pede permissão para transmitir seus dados, enviando uma requisição chamada **RTS (Request to Send)**, e o receptor responde com uma resposta chamada **CTS (Clear to Send)**.

Imagine o professor explicando uma matéria na sala de aula. O aluno levanta sua mão para tirar um dúvida, e só poderá falar após o professor deixar.

Um exemplo prático de comunicação Half-Duplex seria o usado de rádios walkie-talkie. Onde ambos podem se comunicar, mas não juntos. Então é dito frases como “câmbio” ou “câmbio desligo”.

### ****Full-Duplex (Comunicação Duplex Completa):****

Esse tipo de comunicação é o mais complexo, pois ambos os dispositivos podem transmitir e receber respostas simultaneamente. Aqui não existem colisões.

Um exemplo clássico seria a ligação telefônica, onde ambas podem conversar e falar ao mesmo tempo de forma natural e contínua.

### ****Modos de comunicação:****

Nesse tópico abordaremos o que são modos de comunicação, que são a maneira, e o como os dispositivos trocam informações e se comunicacam entre eles em uma rede de internet. Esses modos funcionam na Camada 2 do modelo OSI; **Camada de Enlace de Dados (Data Link)**. Existem basicamente 4 modos de comunicação, sendo… **Unicast, Multicast, Broadcast** e **Anycast**.

## UNICAST — Comunicação “um para um”:

Esse é o modo de comunicação mais usado em comunicação em redes, sendo também o mais simples.

Aqui os dados são enviados de um dispositivo para o outro de forma direta, onde o dispositivo envia um terminado pacote de dados diretamente para o remetente identificado pelo seu endereço IP ou endereço MAC de destino.

Um exemplo seria o envio de email. Você envia o email diretamente para um servidor **SMTP **(Simple Mail Transfer Protocol)****.

## BROADCAST — Comunicação “um para todos”

Aqui o dispositivo envia os dados, e eles são transmitidos para todos os dispositivos conectados naquela rede local. Todos recebem os dados mesmo que não tenham solicitado.

Esse tipo de comunicação é comumente usado onde um dispositivo quer descobrir o endereço MAC ou endereço IP de um determinado dispositivo na rede local. Ele basicamente pergunta ao switche, e este distribuí a todos outros dispositivos conectados na rede. O dispositivo que tiver o IP retorna a resposta.

## MULTICAST — Comunicação “um para muitos (selecionados)”:

Esse tipo de comunicação é um intermediário entre o Unicast e o Brodcast, pois aqui os dados são enviados para todos, porém todos os que tiverem inscritos naqueles dados que estão sendo enviados.

Um dispositivo basicamente envia os dados para um endereço multicast. E os dispositivos que querem receber aqueles dados, se inscrevem naquele grupo por meio do procolo **IGMP (Internet Group Managment Protocol)** por exemplo, e então os dados são enviados somente para esses.

## ****Anycast**** — Comunicação "1 para o mais próximo":

Esse modo de comunicação é bem parecido com o modo **Unicast**, pois aqui a comunicação também funciona 1:1, porém o dados são enviados para dispositivo mais próximo. E quem determina qual dispositivo está mais próximo é o roteador.

Esse modo de comunicação é amplamente usado em redes **WAN (Wide Area Network)**, ou redes a longa distância.

Um exemplo prático desse modo seria o uso de servidores DNS ao enviar uma requisição para o nevegador. Você será atendido pelo servidor mais próximo, gerando baixa latência na entrega, e alta disponibilidade dos dados.

### ****Ethernet II:****

Ethernet II é um protocolo da Camada 2 do modelo OSI, a **Camada de Enlace de Dados (Data Link)**, usado para enviar pacotes para dispositivos em uma rede local **LAN (Local Area Network)**. Esse é sem dúvidas o protocolo mais utilizado nos dias atuais quando se trata de rede local LAN.

Sua estrutura contém 7 quadrados, sendo o **Preâmbulo**, **SFD (Start Frame Delimiter)**, **Endereço MAC de Destino**, **Endereço MAC de Origem**, **Type (Tipo de Protocolo)**, **Payload (Dados)** e o **FCS (Frame Check Sequence)**. Vamos entender melhor um a um:

**1. Preâmbulo (7 bytes):**

É basicamente uma sequência de bits que vem antes do ínicio do quadro, como se fosse um aquencimento para a comunicação entre os dispositivos.

Este quadro serve como um sinal de aviso para os switches e placas de rede, indicando que os dados serão transmitidos ali. Esses **7 bytes** que ele carrega não tem uma informação propriamente dita, sendo apenas um aviso de preparação dos dados que serão enviados.

**2. SDF (Start Frame Delimiter) (1 byte):**

Aqui que começa a transmissão dos dados, ele envia a mesma sequência do preâmbulo, porém com mais um byte de valor 1 no final indicando que a transmissão começou.

**3. Endereço MAC de Destino (6 bytes):**

É o endereço da placa de rede do dispositivo cujo receberá o dado enviado.

**4. Endereço MAC de Origem (6 bytes):**

É o endereço da placa de rede do dispositivo que fará o envio dos dados.

**5. Type (Tipo de Protocolo) (2 bytes):**

Aqui é enviado qual o formato do protocolo que está sendo enviado, sendo o **IPv4**, **IPv6** ou **ARP ( Adress Resolution Protocol )** que é o protocolo comumente usado para descobrir qual dispositivo tem o exato endereço MAC por meio de um IP.

Se na definição do tipo o valor for menor que **0x600 (1536 em decimal)** o protocolo pode ser **Ethernet IEE 802.3**, que é um protocolo mais antigo.

**6. Payload (Dados) (46 – 1500 bytes):**

São os dados que serão transmitidos. Aqui os dados precisam ter no mínimo **46 bytes**. Caso sejam menores que **46 bytes** é acrecentado valores **0**. Isso é feito para evitar problemas na detecção de colisões.

Caso os dados passem de **1500 bytes** é necessário fazer a fragmentação desses dados, ou usar **Jumbo Frames**, geralmente usados em Datacenters por exemplo.

**7. FCS (Frame Check Sequence) (4 bytes):**

**É uma sequência de bits gerado por um algoritmo chamado **CRC (Cyclic Redundancy Check)**,usado para verificar se os dados chegaram corretamente até o dispositivo de destino.**

**Caso os valores não foram condizentes com o valor enviado no dispositivo de origem, os pacotes serão descartados.**

### ****Enderaçamento IP:****

**Os endereços Ips são dividos entre **IPV4** e **IPV6**.**

* **IPv4 (Internet Protocol version 4)**: Aqui os endereçamentos de IP são compostos de 32 bits, por exemplo ( 192.168.1.1 ) suportando 4,3 bilhões de endereços únicos.

O endereço é dividido em 4 bytes ( 8 bits ) cada um variando de 0 a 255, separados em 5 tipos de classes ( A, B, C, D, E ), permitindo assim uma organização e alocação de endereços sistêmica, reduzindo a fragmentação de endereços, suportando até mesmo a criação de sub-redes e seus respectivos gerenciamentos, permitindo assim que múltiplos dispositivos em uma única rede local compartilhem entre si o mesmo endereço de IP público. Porém com o tempo, devido a limitação de números de endereço foi necessário adotar o **Protocolo IPv6**.

* **IPv6 (Internet Protocol version 6)**: Aqui os endereçamentos de IP são compostos de 128 bits em formato hexadecimal, por exemplo ( 2001:0db8:85a3:0000:0000:8a2e:0370:7334 ), permitindo assim um número vasto de endereços, tendo uma limitação muito alta. O IPv6 permiti que os dispositivos façam sua **auto configuração sem estado ( SLAAC - Statelles Address Autoconfiguration )**, assim os dispositivos criem seus próprios IPs automaticamente. E para garantir a segurança foi adotado o método de **suporte IPsec ( Internet Protocol Security )**, trazendo suporte para as futuras tecnologias.

Os IPs são dividos em classes para determinar quantas possibilidades de registros podem ser gerados.

**A –** 0 á 127 **–** 2**^**24 **–** 16.777.216

**B –** 128 á 191 **–** 2**^**16 **–** 65.536

**C –** 192 á 223 **–** 2**^**8 **–** 256

**D –** 224 á 239 **–** Classe reservada para Multicast.

**E –** 240 á 255 **–** Classe reservada para testes.

Além desses IPs também temos aqueles que são restritos, usados para uso interno das empresas ou provedoras de internet...

10.0.0.0

172.16.0.0

192.168.0.0

127.0.0.0

169.254.0.0

0.0.0.0

Os endereços IPs também possuem máscaras de subredes, que ajudam a verificar a quantidade de possibilidades de registro que uma rede pode ter:

**A –** 255.0.0.0

**B –** 255.255.0.0

**C –** 255.255.255.0

Os números **255** representam os octetos como cheios, chamados em seu conjuto como **Redes**, e os números **0**, representam os octetos como livres, chamados em seu conjunto como **Hosts**.

### ****Cálculo de Sub-redes:****

**O **Cálculo de Sub-redes** é uma ferramenta utilizada na rede computadores para dividir uma rede em vários grupos de redes menores.**

**Em uma empresa por exemplo as rede principal é divídida entre os setores, havendo uma Sub-rede para RH, outra para a contabilidade, outra para a administração, outra para a portaria e assim por diante.**

**Dividir uma rede em Sub-redes tem vários benefícios como:**

1. ****Otimização:** Evita desperdicios, pois são utilizados apenas a quantidade de IPs que o departamento necessita.**
2. ****Desempenho:** O trafégo de uma Sub-rede não interfere nas outras.**
3. ****Segurança:** Você pode criar uma rede para os clientes e clientes. Assim se um vírus ou ataque ocorrer ela não afetará todas as Sub-redes, apenas aquela que estiver conectada.**
4. ****Gerenciamento e Organização:** Identifica cada departamento, e facilita a identificação e resolução de problemas.**
5. ****Controle de Broadcast:** Faz um controle nas mensagens broadcast, possibilitando que sejam enviadas apenas para a Sub-rede de destino.**

**Para fazer o cáculo de demonstração usaremos o seguinte o seguinte Endereço IP, e a seguinte Máscara de Sub-rede:**

****IP Host:** 192.168.1.10 .**

****Máscara de Sub-rede:** 255.255.255.192 .**

* ****1º Passo:** O primeiro passo é subtrair 256 pelo último octeto com valor na máscara de Sub-rede, para o obter o salto.**

**256 – 192**

****64****

* ****2º Passo:** Montamos a tabela de Rede, Host, Broadcast. Somando primeiro as **Redes**, depois os **Broadcasts**, e por últimos os **Hosts**.**

|  |  |  |
| --- | --- | --- |
| **Rede** | **Host** | **Broadcast** |
| 192.168.1.0  **+ 1** | 192.168.1.1  **até**  192.168.1.62 | 192.168.1.63  **- 1** |
| 192.168.1.64  **+ 1** | 192.168.1.65  **até**  192.168.1.126 | 192.168.1.127  **- 1** |
| 192.168.1.128  **+ 1** | 192.168.1.129  **até**  192.168.1.190 | 192.168.1.191  **- 1** |
| 192.168.1.192  **+ 1** | 168.168.1.193  **até**  192.168.1254 | 192.168.1.255  **- 1** |
| 192.168.1.256  **+ 1** | **--------**  **--------**  **--------** | **--------**  **--------**  **--------** |