

**Resumo da UC - Tópicos Avançados de Redes**

Tópicos Avançados de Redes

Licenciatura de Engenharia Informática

Tomás Ferreira Baptista nº 2170851

Índice

[IPv6 4](#_Toc113025497)

[IPv4 vs IPv6 4](#_Toc113025498)

[Tipos de Endereços IPv6 6](#_Toc113025499)

[OSPF: v3 & Advanced 8](#_Toc113025500)

[OSPFv3 8](#_Toc113025501)

[Tipos de Routers 8](#_Toc113025502)

[Pacotes de Encaminhamento OSPF 9](#_Toc113025503)

[LSAs do OSPFv3 9](#_Toc113025504)

[Áreas: 10](#_Toc113025505)

[Tipos de Areas: 10](#_Toc113025506)

[Sumarização 12](#_Toc113025507)

[Virtual Link 12](#_Toc113025508)

[BGP 13](#_Toc113025509)

[BGP Advanced 14](#_Toc113025510)

[Paths 14](#_Toc113025511)

[Confederações 15](#_Toc113025512)

[MPLS (Multiprotocol Label Switching) 16](#_Toc113025513)

[Terminilogia MPLS: 16](#_Toc113025514)

[Propósitos da VPN: 17](#_Toc113025515)

[Terminologia MPLS L3 VPNs: 17](#_Toc113025516)

[Multicast 19](#_Toc113025517)

[Multicast Listener Discovery (MLD) 20](#_Toc113025518)

[Protocol Independent Multicast (PIM) 20](#_Toc113025519)

[Qualidade de Serviço (QoS) 22](#_Toc113025520)

[Gestão de Congestão 22](#_Toc113025521)

[Introdução às Filas de Espera (Queuing) 22](#_Toc113025522)

[Implementação de Queuing 22](#_Toc113025523)

[Configuração de FIFO e WFQ 22](#_Toc113025524)

[Configuração de CB-WFQ e LLQ 22](#_Toc113025525)

[Controlo de Congestão 22](#_Toc113025526)

[Policing e Shaping de Tráfego 22](#_Toc113025527)

[Segment Routing (SR) 23](#_Toc113025528)

[Control Plane e Data Plane 23](#_Toc113025529)

[Traffic Protection 23](#_Toc113025530)

[Segment Routing e LDP Internetworking 23](#_Toc113025531)

# IPv6

**Características:**

* 128-bit de espaço de endereçamento, escrito em hexadecimal.
* IPv6 Header-40 bytes fixos.
* Autoconfiguração de endereços.
* Não há espaço de endereçamento privado.
* Suporte para mobilidade.
* Maior capacidade para VoIP e QoS.
* Ipv6 origem e sempre unicast.
* Ipv6 destino pode ser uni, multi ou anycast.
* NAT não é necessário.
* IPv6 não tem endereços de broadcast

## IPv4 vs IPv6



Figura 1 - Cabeçalho IPv6

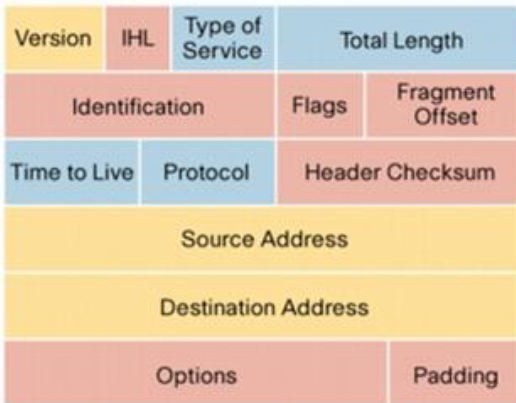
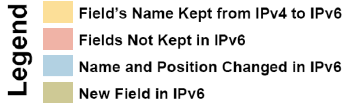


Figura 2 - Cabeçalho IPv4



|  |  |  |
| --- | --- | --- |
| CABEÇALHOS | IPv6 | IPv4 |
| Bits de endereçamento  (source & destination) | 128-bit | 32-bit |
| IHL (Internet Header Length) | 40 fixos | Depende do campo ***Options*** |
| Version | IPv6 Version - valor 6 | IPv4 Version - valor 4 |
| Igual | **Traffic Class** | **Type of Service** |
| Flow Label | - Usado para identificar os pacotes de um fluxo ou stream.  - Os pacotes com o mesmo flow label têm a mesma origem e o mesmo destino, não devem ser reordenados. | X |
| Identifica o tipo de protocolo a seguir ao IP | **Next Header** | **Protocol** |
| Número de bytes do payload | **Payload Length**  - Inclui extension headers + data  - extension headers – usado em fragmentação | **Total Length**  Número de bytes do IPv4 header (options) + data. |
| •Definido na origem  •Cada router decrementa em 1 o valor do Hop Limit  • Ao chegar a 0, o router deita fora o pacote | **Hop Limit** | **TTL (Time to Live)** |
| Header Checksum | Obrigatório em IPv6 UDP | Opcional daí ser um campo |
|  | Options são usadas como **Extension Headers** | **Options e Padding** |

## Tipos de Endereços IPv6

**UNICAST:**

* **Global Unicast** (GUA) – desde *2000::/64 até 3FFF:FFF:FFF:FFF::/64*
  + É sempre unicast a origem
  + O destino é unicast, multicast ou anycast
  + Globalmente único
* **Link-Local** – *desde FE80::/10 até FEBF::/10*
  + Usados para comunicarem com a rede antes de terem um GUA
* **Loopback** – ::1/128
  + Usado num nó para enviar pacotes IPv6 a si próprio, tipicamente para testar a TCP/IP stack
* **Unspecified** – ::/128
  + Usado como endereço IPv6 origem durante o processo de deteção de endereços duplicados
* **Unique Local** – FC00::/7
* **Embedded** - ::/80

**MULTICAST:**

* Assigned – FF00::/8
* Solicited Node – FF02::1:FF00:0000/104

**Anycast**

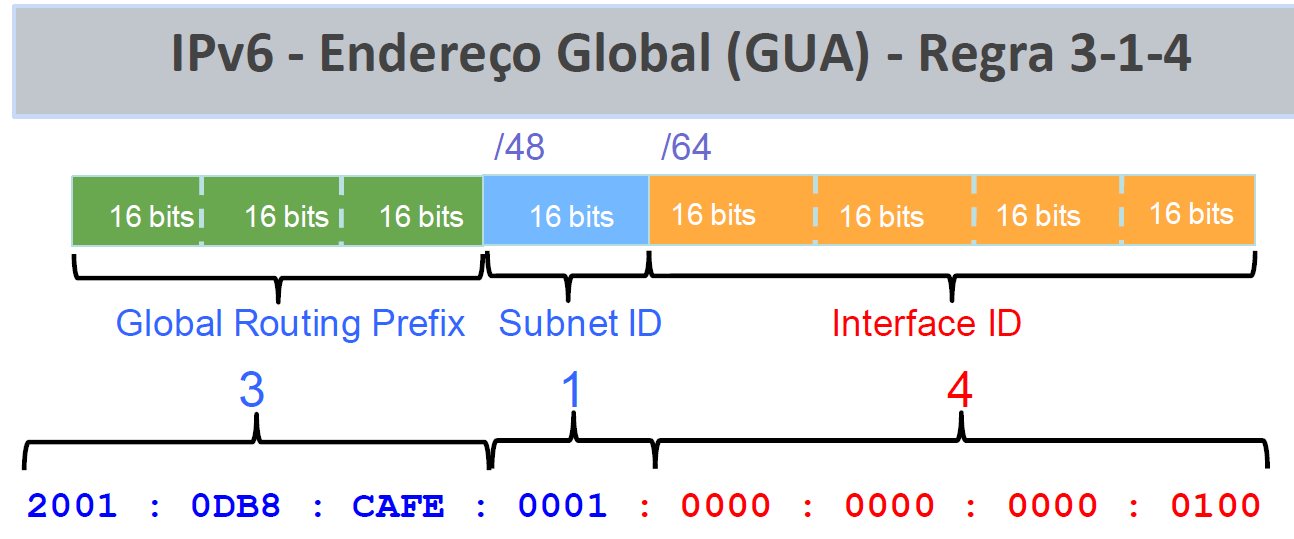


Figura 3 - GUA regra 3-1-4

Os 64 - bit da Interface ID são criados usando o EUI-64:

* 48-bit MAC address da interface
* Inserem-se 16 bits: FF-FE
* Uma imagem com mesa

  Descrição gerada automaticamenteAltera-se o U/L (Universal/Local) bit (7º bit do MAC address)

# OSPF: v3 & Advanced

## OSPFv3

**Características:**

* É um link-state protocol 🡪 Após a inundação inicial de LSAs, os routers link-state passam pequenas atualizações de link-state acionadas por eventos para todos os outros routers.
* O OSPF propaga anúncios link-state em vez de atualizações da tabela de roteamento:
  + Link = interface do router
  + Estado = descrição de uma interface e seu relacionamento com routers vizinhos
* OSPF usa o algoritmo SPF (Shortest Path First) para calcular o caminho mais curto para o destino

## Tipos de Routers

* Internal Routers (IR) 🡪 dentro de uma área
* Backbone Routers (BR) 🡪 dentro da área 0
* Area Border Routers (ABR):
  + Está entre duas áreas
  + Tem de estar ligado à área 0
* Autonomous System Boundary Routers (ASBR):
  + Se o router fizer redistribuição é um ASBR

Uma imagem com texto

Descrição gerada automaticamente

## Pacotes de Encaminhamento OSPF

* Os pacotes são enviados com alta prioridade pelo type of service (IPv4) ou traffic class (IPv6)
* Há 5 tipos de pacotes (e são executados por esta ordem):
  + Hello
  + Database Description (DBD)
  + Link-State Request (LSR)
  + Link-State Update (LSU)
  + Link-State Acknowledgment (LSAck)

## LSAs do OSPFv3

**LSA 2001 –** Cada router dentro da sua área informa o router DR sobre a sua identificação e os seus links.

**LSA 2002** **–** O DR transmite para todos os routers da área a informação sobre a existência de todos os routers da área.

**LSA 2003** **–** Este LSA é criado pelos routers ABR para transmitirem aos routers de uma área a informação sobre os prefixos de outra área.

**LSA 2004** **–** Este LSA é criado pelos routers ABR para transmitirem aos routers de uma área a informação sobre a existência de R1 como router ASBR, ou seja, um router ligado a um sistema externo.

**LSA 2008** **–** Cada router envia aos routers vizinhos, ligados ao mesmo link, o seu prefixo e o seu endereço link-local.

**LSA 2009** **–** Cada router envia aos routers vizinhos todos os seus prefixos.

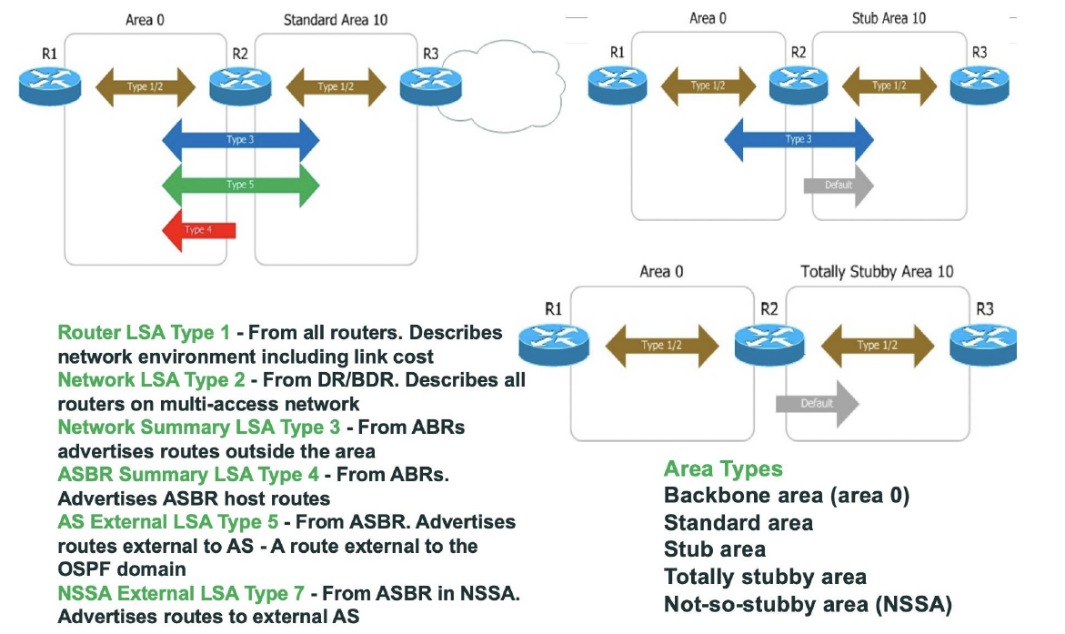
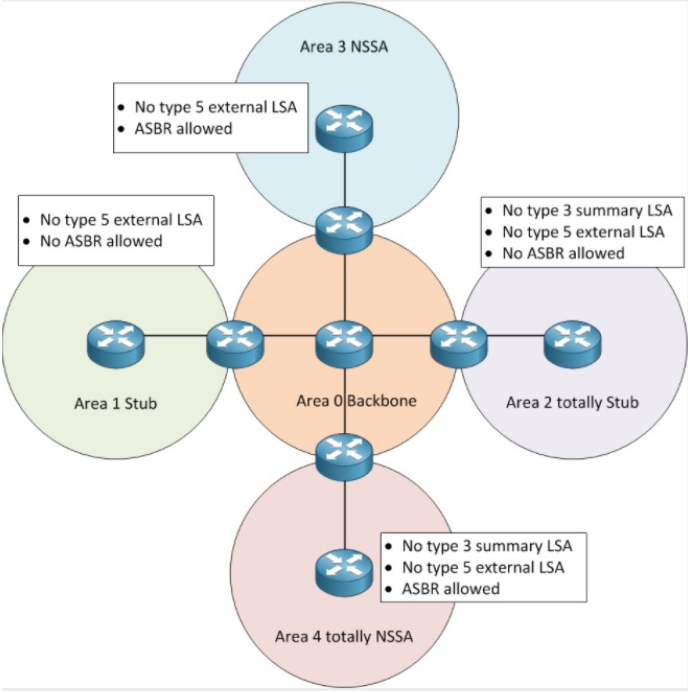
**LSA 4005** **–** O router R1 cria um LSA 4005 para informar os restantes routers da existência do prefixo externo. Na tabela de encaminhamento este prefixo aparece identificado por OE1 ou OE2.

## Áreas:

* Usa um modelo de hierarquia de 2 níveis
* Áreas são definidas com o número de 32 bit
* Área 0 reservada para o backbone
* Todas as áreas têm de estar ligadas à área 0

### Tipos de Areas:

* **Area Standard**
  + Recebe um resumo dos LSAs de outras áreas
  + Os LSAs com informação de ligações para redes externas são distribuídos.
* **Area Stub**
  + Recebe um resumo dos LSAs de outras áreas
  + Os LSA do tipo5 (rotas redistribuídas) não são enviados
  + É enviado um LSA com uma rota default
  + Todos os routers da área são definidos como stub
* **Area Totally Stub** 
  + Enviado para a área um LSA com uma rota default
    - Representa todos os links externos
    - Representa todos os links internos resumidos
    - Representa links internos não resumidos
  + A base de dados LSDB é pequena e estável
  + A tabela de encaminhamento é pequena
* **NSSA**
  + Beneficios da Stub Area mas são permitidos ASBR
  + New External LSA type 7, porque as externas de tipo 5 não são permitidas
  + Filtragem e sumarizações são permitidas nos ABRs
* **Totally NSSA**
  + Mesmo que NSSA mas também restringe type 3 LSA
  + Apenas router, network, Type-7 LSA e default-route permitidos



## Sumarização

* Encaminhamento é feito pelo prefixo mais longo em vez de comunicar vários prefixos específicos, apenas um sumarizado
* Mais pequeno e mais estável
* Drawback é possível sub optimal routing
* Apenas vão ser transmitidos LSA sumarizados e as link-state changes não são propagados
* Pode ser usado o comando “Area-Range” para fazer sumarização apenas quando se envia LSAs PARA a área 0 ou rotas sumarizadas originadas NA área

## Virtual Link

* Todas as áreas têm de estar ligadas à área 0
* Em alguns casos onde não é possível podemos usar um virtual link para ligar o backbone(area 0) através de uma non-backbone área .
* A área que faz esta ligação é denominada de Transit Area e tem que ter toda a informação de encaminhamento.
* Tem de ser configurado entre 2 ABRs
* A Transit Area não pode ser STUB

# BGP

**Características:**

* É um Path Vector protocol 🡪 mantém as informações de caminho que são atualizadas dinamicamente. As atualizações que percorreram a rede e retornaram ao mesmo nó são facilmente detetadas e descartadas.
* Escalabilidade de uma grande rede
  + Hierarquia
  + IGPs periódicos/inundações
  + Isolar a estabilidade da rede
* Políticas Complexas
  + Controlar a acessibilidade aos prefixos
  + Fundir organizações separadas
  + Conecte vários IGPs
* Corre sobre TCP (porto 179)
* Faz updates incrementais
* Trocar mensagens para abrir e confirmar os parâmetros de ligação
* Troca inicial de toda a tabela
* Troca de mensagens Keepalive

**NEXT-HOP:**

Basicamente, o Next-Hop força o router a fazer uma pesquisa recursiva para determinar qual interface de saída deve ser usada para enviar os pacotes.

**NULL 0:**

Null0 é uma interface virtual especial em routers Cisco que funciona como um coletor de dados: o que quer que seja roteado para fora da interface Null0 é perdido (os pacotes são simplesmente descartados). Esta interface não tem endereço IP, está sempre ativa, na maioria das vezes nem é visível na configuração ou nas saídas da interface, e este é seu único propósito: **Atuar como um buraco negro para todos os pacotes que são roteados por ela.**

**Router Reflectors:**

* Forneça controle adicional para permitir que o router anuncie (reflita) rotas aprendidas iBGP para outros pares iBGP
  + Método para reduzir o tamanho da malha iBGP
* Alti-falantes BGP normais podem coexistir
  + Apenas o RR tem de suportar este recurso
  + neighbor x.x.x.x route-reflector-client
* Evita loops

## BGP Advanced

Paths**:**

1. Considerar apenas o caminho com NEXT\_HOPs alcançável **🡪 Well-known mandatory**
2. Não considere o caminho iBGP se não estiver sincronizado
3. Maior WEIGHT
4. Maior LOCAL\_PREF
5. Prefere a rota originada localmente
6. O AS\_PATH mais curto **🡪 Well-known mandatory**
7. Menor ORIGIN code **🡪 Well-known mandatory**
   * IGP < EGP < incomplete
8. Menor Multi-Exit Discriminator (MED)
   * Default (0): Considerado apenas se os caminhos forem do mesmo vizinho AS
9. Prefere um caminho externo ao interno
10. Métrica IGP mais baixa para o NEXT\_HOP
11. Para caminhos eBGP
    * Se o multipath estiver habilitado, o router pode instalar até N caminhos paralelos na tabela de roteamento, mas o melhor caminho ainda deve ser determinado.
    * Selecione o mais antigo, a menos que as atualizações cheguem ao mesmo tempo.
    * Se as atualizações chegaram ao mesmo tempo, veja o próximo marcador
12. ID de roteador mais baixo
    * Originator-ID é considerado para rotas refletidas
13. Cluster-List mais curta
    * O cliente deve estar ciente dos atributos RR!
14. Endereço IP do vizinho mais baixo

### Confederações

* Resolve o problema do BGP mesh;
* Divide as AS em sub-AS
* Visível para o mundo exterior como uma única AS
* Preserva local-preference, MED e next-hop
* Os iBGP speakers dentro de cada sub-AS são fully meshed
* Os RR podem ser usados dentro destas mesmas sub-AS

# MPLS (Multiprotocol Label Switching)

Veio a resolver problemas relacionados com encaminhamento pacotes e tornar mais rápido o processo de encaminhamento.

Tem dois componentes:

* **Control plane:** 
  + responsável pela construção das tabelas MPLS
  + precisa de protocolos como OSPF, EIGRP para o seu funcionamento uma vez que copia informação de outros protocolos para a sua tabela
* **Forwarding plane:** 
  + responsável pelo encaminhamento dos pacotes
  + usa a tabela MPLS

## Terminilogia MPLS:

**Routing Information Base (RIB)** 🡪 é análoga à tabela de roteamento IP.

**Forwarding Information Base (FIB)** a.k.a. CEF 🡪 derivada da tabela de roteamento IP

**Label Information Base (LIB)** 🡪contém todas as ligações de rótulo aprendidas via LDP

**Label Forwarding Information Base (LFIB)** 🡪derivada de entradas FIB e entradas LIB correspondentes

**Forwarding Equivalence Class (FEC):**

* Grupo de pacotes IP encaminhados da mesma maneira (por exemplo, no mesmo caminho de encaminhamento)
* Um FEC pode representar um: prefixo de IP de destino, ID de VPN, VC ATM, ID de VLAN, Traffic Engineering tunnel, Class of Service.

**Label Distribution Protocol (LDP)** 🡪 Um mecanismo pelo qual dois Label Switch Routers (LSR) trocam informações de mapeamento de rótulo. Este protocolo é definido pelo IETF (RFC 5036).

**Label Edge Router (LER)** 🡪 Um router que opera nas extremidades de uma rede MPLS. Um LER determina e aplica os rótulos apropriados e encaminha os pacotes rotulados para o domínio MPLS.

**Provider Edge (PE)** 🡪 O LER que funciona como roteadores de entrada e/ou saída para o domínio MPLS.

**Label Switch Router (LSR)** 🡪 Um roteador que comuta os rótulos usados para rotear pacotes através de uma rede MPLS.

**Provedor (P)** 🡪 O LSR que funciona no núcleo da rede MPLS

## Propósitos da VPN:

* Permite uma conexao flexivel entre inter-sites
* Podem tar no mesmo ou em diferentes organizaçoes
* VPN-ovelap em que um site pode tar em mais do que uma VPN
* Estes sites podem estar em multiplos ISP

## Terminologia MPLS L3 VPNs:

**Route-Target** - Routers Identificadores de 64 bits que recebem os routers

**Route Distinguisher** - Atributos específicos de cada rota para identificar os prefixos de 64 bits e é VRF based

**VPN-IPv4 endereços** - Incluem a RD e o 32bit IP endereço

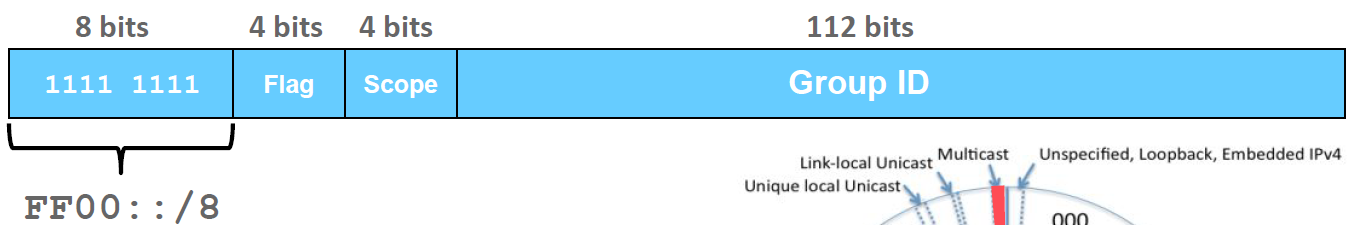
**VRF (VPN Routing and Forwarding)** - Tabela de encaminhamento da VPN e distribuído através do protocolo de encaminhamento

**VPN-Aware Network** - Rede do operador onde esta a funcionar o serviço do MPLS

Para ter uma tabela de encaminhamento MPLS a funcionar é preciso:

* Protocolo ISP a funcionar
* O MPLS constrói a sua própria tabela consultando as tabelas IGP atribuindo uma label numérica a redes diretamente ligadas e uma label não numérica as diretamente ligadas, e troca com os routers vizinhos esta tabela

# Multicast

O campo **scope** é utilizado para definer o alcance do pacote

A **flag** é atribuída pela IANA e inclui os endereços assigned e solicited-note se for 0, e 1 atribuído dinamicamente

A associação dos terminais é feita pelo protocolo **Multicast Listener Discovery (MLD)**

O IPv6 apenas suporta o protocolo de encaminhamento multicast **Protocol Independent Multicast (PIM):**

* O PIM cria a arvore multicast
* Para serem associados os terminais é usado PIM (as funções em IPv6 são iguais às em IPv4)

Está organizado em árvore e uma única cópia sai da fonte até aos recetores, não existindo restrições físicas para estes.

**Tipos de Grupos Multicast:**

* Any Source Multicast(ASM)
* Source-Specific Multicast(SSM)
* Embedded RP groups

## Multicast Listener Discovery (MLD)

* É um componente do IPv6 para descobrir os dispositivos que pretendem escutar Multicast
* Semelhante ao IGMP do Ipv4, mas usa ICMPv6
* Descritos nas RFC 3810 e RFC 4604

## Protocol Independent Multicast (PIM)

PIM é o único protocolo de encaminhamento multicast para IPv6

Constrói a tabela de encaminhamento multicast nos routers

Modos de funcionamento:

* PIM Sparse-Mode (PIM»SM) - RP obrigatório (Multiple sources, single group)
* Bi-directional PIM (PlM-BiDir) - RP obrigatório.
* Bi-Directional many-to-many (hosts can be sources and receivers)
* PIM Source»Specific Multicast (PIM-SSM) - Sem RP (Single source, single group) (S, G)

O encaminhamento Multicast é o oposto de encaminhamento Unicast

* O unicast está preocupado com para onde o pacote vai
* O multicast está preocupado com de onde o pacote vem

O multicast usa **reverse-Path Forwarding (RPF)**

* Verifica se os pacotes multicast de um grupo chegam pela interface que conduz à fonte
* Se não, são descartados
* Este procedimento evita loops

O procedimento RPF para PIM utiliza a tabela de encaminhamento unicast para encontra a fonte

# Qualidade de Serviço (QoS)

**Classificação:** Qualquer mecanismo de QoS orientado à classe tem de suportar algum tipo de classificação. Serve para identificar e separar o tráfego em diferentes classes. Sendo classificado por IP fonte/destino, Porto fonte/destino, protocol, Ip precedence, DSCP,etc

**Marcação:** Utilizado para marcar os pacotes baseado na classificação, medição ou ambos.

**Gestão de Congestão:** Cada interface deve ter um mecanismo de queuing para priorizar a transmissão de pacotes. Faz uso da Marcação para saber em que queue e que vai meter cada pacote, usando mecanismos como WFQ e LLQ.

**Controlo de Congestão:** Utilizado para descartar pacotes aleatórios proactivamente para evitar congestão futura na rede usando tecnologias como Random Early Detection(RED) e Weighted Random Early Detection.(WRED)

**Policing e Shaping:** Utilizado para limitar um débito baseado numa medição (o tráfego em excesso pode ser descartado, marcado com policing ou atrasado com shaping)

**Eficiência da ligação:** Utilizado para melhorar a eficiência da largura de banda através de compressão, fragmentação e interleaving

## Gestão de Congestão

### Introdução às Filas de Espera (Queuing)

A congestão pode ocorrer em qualquer ponto da rede onde existam pontos com velocidades discordantes, agregação ou confluência.

As filas de espera gerem a congestão para providenciar garantias de largura de banda (bandwidth) e atraso (delay)

**Tipos de Filas:**

* FIFO
* PQ
* RR (Round Robin)
  + Utiliza várias filas de espera
  + Não permite priorização
  + Enviar um pacote de cada fila em cada ronda
    1. Um pacote da fila 1
    2. Um pacote da fila 2
    3. Um pacote da fila 3
    4. Repetir
* WRR (Weighted Round Robin)
  + Utiliza várias filas de espera
  + Permite priorização
  + Enviar pacotes de cada fila proporcionalmente a um peso (weight) assignado:
    1. Envia até 4 da fila 1
    2. Envia até 2 da fila 2
    3. Envia 1 da fila 3
    4. Voltar à fila 1

### Implementação de Queuing

O sistema **Hardware Queuing** utiliza sempre um mecanismo FIFO

O sistema **Software Queuing** pode ser selecionado e configurado e depende da plataforma que lhe serve de suporte (i.e., diferente de vendedor para vendedor e mesmo dentro do mesmo vendedor, entre famílias diferentes de equipamentos)

### 

### Configuração de FIFO e WFQ

**FIFO**:

**Positivo**

* Simples e rápido (apenas uma fila de espera com um simples mecanismo de escalonamento (scheduling))
* Suportado em todas as plataformas
* Suportado em todas as versões de SW

**Negativo**

* Causa starvation (fluxos agressivos podem monopolizar os links)
* Causa jitter bursts ou packet trains de pacotes temporariamente saturam a fila de espera)

**WFQ (Weight Fair Queuing):**

* Um algoritmo de queuing deverá partilhar a largura de banda de uma forma justa (fair) entre os fluxos:
  + Reduzindo o tempo de resposta para fluxos interativos escalonando estes pacotes para a frente da fila de espera de HW
  + Prevenindo que fluxos de grande volume de tráfego monopolizem uma interface
* Na implementação de WFQ, os pacotes são ordenados em fluxos
* Unfairness é reintroduzida através de um peso (weight) de forma a dar proporcionalmente mais largura de banda a fluxos com um valor maior de IP Precedence

### Configuração de CB-WFQ e LLQ

**LLQ (Low Latency Queuing):**

LLQ = PQ + CB-WFQ

* Filas de espera prioritárias foram adicionadas ao CB-WFQ para tráfego de tempo real
* As classes de tráfego de alta prioridade garantem:
  + Propagação de pacotes com baixa latência
  + Largura de banda
* As classes prioritárias são muitas vezes policiadas –não podem exceder a sua LB garantida
* As classes menos prioritárias utilizam CB-WFQ

**PQ (Priority Queuing):**

* Utiliza várias filas de espera
* Permite priorização
* Esvazia sempre a 1ª fila de espera antes de passar para a outra:

1. Esvaziar a fila de espera 1
2. Se a fila 1 estiver vazia, enviar um pacote da fila 2
3. Se ambas as filas 1 e 2 estão vazias enviar um pacote da fila 3

* Pode levar à fome (starvation)das filas de espera 2 e 3

**CB-WFQ (Class Based – Weight Fair Queuing)**

* CB-WFQ fornece garantias de largura de banda de acordo com pesos (weights) assignados às classes
* Os pesos podem ser definidos especificando:
  + Largura de Banda (bandwidth), em Kbps
  + Percentagem da LB (percentagem da largura de banda disponível numa interface)
  + Percentagem da largura de banda remanescente

## Controlo de Congestão

São usados:

* Tail Drop
  + Os pacotes adicionais são descartados
  + Apenas são descartados pacotes quando a fila está cheia
* RED (Random Early Detection) – alternativa ao Tail Drop
  + Descartar pacotes antes que a fila encha
  + As filas são mantidas com poucos pacotes
  + Isto permite aumentar o throughput
  + Podemos falar em 3 modos de operação:
    - Sem descartes: quando o tamanho da fila de espera médio está entre 0 e o threshold mínimo
    - Com descartes aleatórios: quando o tamanho da fila de espera médio está entre o threshold mínimo e o threshold máximo
    - Tail Drop : Quando o tamanho da fila de espera médio está igual ou acima do threshold máximo
  + Os descartes aleatórios devem evitar que a fila encha e evita o tail drop.

## Policing e Shaping de Tráfego

**Shaping:**

* Previne e gere a congestão quando são utilizadas larguras de banda assimétricas ao longo do caminho percorrido pelo tráfego.
* Regula o débito a que o tráfego é enviado tendo em conta o débito contratado para o serviço.

***Características:***

* Aplica-se apenas à saída dos Interfaces
* Os pacotes fora do perfil definido são colocados em buffers até um limite definido
* O buffering minimiza as retransmissões TCP
* Não suporta marcação ou remarcação de pacotes

**Policing:**

* Limita o acesso a recursos quando utilizamos uma interface de alto débito, mas pretendemos um acesso sub-rate
* Limita o débito de tráfego para determinadas aplicações ou classes
* Marca (recolorir) o tráfego em excesso (Layer 2 ou Layer 3)

***Características:***

* Aplica-se tanto à entrada como à saída dos Interfaces
* Os pacotes fora do perfil definido são descartados
* O descarte dos pacotes leva a retransmissões TCP
* Suporta marcação ou remarcação de pacotes
* Utiliza menos buffers

# Segment Routing (SR)

As decisões de encaminhamento são tomadas na fonte.

**Segmento:** um identificador para qualquer tipo de instrução

**Non-Disjoint Traffic**

* A envia tráfego com [65]
* Classic ecmp “a la IP”

**Disjoint Traffic**

* A envia tráfego com [111, 65]
* O pacote é atraído no plano azul e depois usa o ecmp clássico “a la IP”

## Control Plane e Data Plane

**Control plane:**

* Roteamento de nível 1, nível 2 e multinível
* ID de segmento de prefixo (Prefix-SID) para prefixos de host em interfaces de loopback
* SIDs de adjacência para adjacências
* Anúncios de mapeamento de prefixo para SID (servidor de mapeamento)
* Sinalização MPLS penúltimo salto de salto (PHP)
* Sinalização de rótulo explícito-nulo MPLS

**Data Plane:**

* Prefix SID
  + Pacote encaminhado ao longo do caminho mais curto do IGP
  + O pacote pode aproveitar o balanceamento de carga ECMP
  + Operação de troca realizada no rótulo de entrada
  + O rótulo de entrada (X) e o rótulo de saída (Y) terão o mesmo valor quando o vizinho downstream tiver o mesmo SRGB
  + O penúltimo salto pode executar uma operação pop (PHP) se sinalizado pela saída
* Adjacency SID
  + Pacote encaminhado ao longo da adjacência IGP
  + Operação pop realizada no rótulo de entrada
  + A etiqueta superior de entrada (X) e a etiqueta de saída (Y) podem ou não ter o mesmo valor
  + O penúltimo salto sempre aparece no último SID da adjacência

## Traffic Protection

**Topology Independent LFA (TI-LFA) – Benefits**:

* Proteção de nó e link de 50 ms com 100% de cobertura
* Simples de operar e entender
  + calculado automaticamente pelo IGP
* Evita congestionamento transitório e roteamento abaixo do ideal
  + aproveita o caminho pós-convergência, planejado para transportar o tráfego
* Implantação incremental
  + também protege o tráfego LDP

## Segment Routing e LDP Internetworking

**LDP to SR**:

* Quando um nó é compatível com LDP, mas seu próximo salto
* ao longo do SPT até o destino não é compatível com LDP
  + nenhum rótulo de saída LDP
* Neste caso, o LDP LSP é conectado ao segmento de prefixo
* C instala a seguinte entrada LDP-to-SR FIB:
  + rótulo de entrada: rótulo vinculado por LDP para FEC Z
  + rótulo de saída: segmento de prefixo vinculado a Z
  + interface de saída: D
* Esta entrada é derivada automaticamente na camada de roteamento

**SR to LDP:**

* Quando um nó é compatível com SR, mas seu próximo salto ao longo do SPT até o destino não é compatível com SR
  + nenhuma etiqueta de saída SR disponível
* Neste caso, o segmento de prefixo é conectado ao LDP LSP
  + Qualquer nó na fronteira SR/LDP instala entrada(s) FIB SR-to-LDP