

# Administração de Redes 2023/24

Encaminhamento estático

Princípios do encaminhamento dinâmico

# Routers

- Vamos trabalhar com dois tipos de routers
  - Routers Cisco com sistema operativo IOS
  - Routers Linux

## **Router**

- Funcionalidade centrada na camada de Rede
- Mais de uma interface de rede (mesmo que virtual)
- Faz reenvio de pacotes recebidos e não destinados a si próprio
- Normalmente corre protocolos de encaminhamento
- Normalmente usa múltiplas rotas específicas

vs

## **Terminal**

- Funcionalidade centrada na camada de aplicação
- Uma ou mais interfaces de rede (pode ser multi-homed)
- Não faz reenvio de pacotes
- Não corre protocolos de encaminhamento
- Normalmente tem apenas rota-padrão (default)

# Encaminhamento estático vs dinâmico

- Encaminhamento estático
  - Baseado em rotas estáticas, configuradas manualmente pelo administrador de rede
  - Rotas não se alteram quando há mudanças na rede (falhas, equipamento adicionado, alterações da topologia)
- Encaminhamento dinâmico
  - Programa distribuído determina rotas a usar
  - Adaptação automática a mudanças na rede
  - Protocolos de encaminhamento
    - Intra-domínio (Intra-AS)
      - Vectores de distâncias (distance-vector, DV)
      - Estado das ligações (link-state, LS)
    - Inter-domínio (Inter-AS)

# Encaminhamento estático

## Vantagens

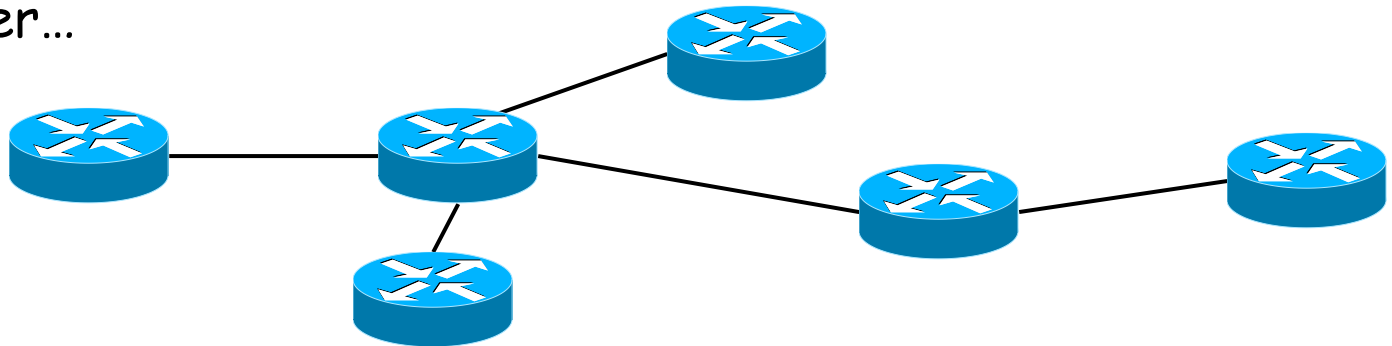
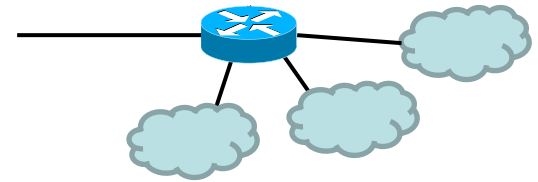
- Mais simples e previsível
- Maior controlo do administrador
- Não consome recursos de rede
- Menor consumo de CPU e memória
  - Sem processo de encaminhamento a correr
- Mais seguro
  - Sem troca de mensagens de encaminhamento na rede

## Desvantagens

- Exige intervenção humana em caso de alterações
- Susceptível a erros humanos
- Sem tolerância a falhas\*
- Normalmente têm métrica administrativa inferior às rotas dinâmicas
  - Podem impedir o encaminhamento dinâmico de funcionar correctamente

# Encaminhamento estático — casos de uso

- Em terminais
  - Não participam nos protocolos de encaminhamento
  - Normalmente têm apenas a rota-padrão
- Quando todas as ligações de um router excepto uma são pontas (*stubs*)
  - Uma rota-padrão é suficiente
- Em topologias de rede não emalhadas (árvores)
  - Sem caminhos alternativos, se uma ligação falhar não há nada a fazer...



# Configuração de rotas estáticas

- Linux

Comando route:

```
route add -net 172.16.1.0 netmask 255.255.255.0 gw 192.168.1.1
```

Comando ip:

```
ip route add 172.16.1.0/24 via 192.168.1.1
```

Ficheiros de configuração (e.g., NM *keyfiles*):

```
/etc/NetworkManager/system-connections/*.nmconnection
```

- Cisco IOS

```
ip route 172.16.1.0 255.255.255.0 192.168.1.1
```

# Rotas estáticas flutuantes

- Rotas têm associada uma métrica (distância administrativa)
  - Sem relação com o custo calculado pelos algoritmos de encaminhamento
- É possível ter várias rotas para o mesmo destino com métricas diferentes
  - É usada a que tiver métrica mais baixa
  - Se esta ficar inactiva (e.g., perda de ligação na interface de saída), passa a ser usada a outra (rota flutuante)
  - Tolerância básica a alguns casos de falha
  - Também pode ser usada para casos de falha de protocolos dinâmicos
- Encaixe de prefixo mais longo tem precedência sobre a métrica (distância administrativa)
  - Uma rota mais específica é preferida mesmo que tenha métrica mais alta
- *Métrica default* para rotas estáticas
  - 0 no Linux
  - 1 no Cisco IOS

# Rotas estáticas flutuantes

- Configuração de rotas estáticas flutuantes
  - No Linux, adicionar opção `metric N` (comandos `route` e `ip`)
  - No Cisco IOS, acrescentar o valor da distância administrativa
- Distância administrativa de alguns protocolos de encaminhamento no Cisco IOS:

Protocolo	Distância Administrativa
Ligação directa	0
Rota estática	1
eBGP	20
OSPF	110
RIP	120

- Usar dist. adm. superior à indicada para a rota ficar flutuante



# Equal Cost MultiPath (ECMP)

- Havendo vários caminhos com igual custo para um dado destino, pode ser interessante distribuir o tráfego entre eles
- Distribuição pode ser
  - Por pacote — distribuição mais equitativa do tráfego
  - Por fluxo — evita reordenação de pacotes que
    - em fluxos TCP pode originar retransmissões desnecessárias
    - não é bem tolerada por alguns protocolos (e.g., de VoIP)
  - Por par <IP de origem, IP de destino> — também evita reordenação de pacotes

# ECMP em Linux

Distribuição de carga por fluxo:

```
ip route add 172.16.1.0/24 nexthop via 192.168.1.1 nexthop via 10.0.0.1
```

Distribuição de carga por pacote (só com kernel alterado):

```
ip route add 172.16.1.0/24 equalize nexthop via 192.168.1.1  
nexthop via 10.0.0.1
```

É possível atribuir um parâmetro weight a cada nexthop para distribuição proporcional de carga

# Reenvio de pacotes em Cisco IOS

- Process switching
  - Consulta normal à tabela de encaminhamento por cada pacote
  - É lento e consome bastante CPU ("Route Processor")
- Fast switching
  - Após consulta à tabela para o primeiro pacote, guarda entrada na *routing cache*
  - Pacotes subsequentes usam essa entrada
  - Mais rápido e eficiente que *process switching*
- Cisco Express Forwarding (CEF)
  - Router constrói previamente tabelas para reenvio (FIB) e de adjacências com base na tabela de encaminhamento
  - Mais eficiente que o *fast switching*
  - Nenhum pacote precisa de ser *process switched*
- CEF distribuído (dCEF)
  - Tabelas FIB e de adjacências em cada interface de entrada
  - Reenvio de pacotes sem intervenção do "Route Processor"
  - Método mais rápido e eficiente

# ECMP em Cisco IOS

- Activado automaticamente ao adicionar multiplas rotas para o mesmo destino com a mesma distância administrativa
- Distribuição de carga por pacote com *process switching*
- Distribuição de carga por par <IP de origem, IP de destino> com *fast switching*
- Distribuição de carga configurável com CEF
  - Normalmente por par <IP de origem, IP de destino>
  - Também possível por pacote:  
ip load-sharing per-packet  
na configuração da interface de entrada

# Encaminhamento dinâmico

- É difícil e trabalhoso configurar rotas em redes de maiores dimensões / com topologias mais complexas
- É necessário reconfigurar rapidamente as rotas em caso de falha de ligação ou router
- Solução: software distribuído que configure automaticamente as rotas → Encaminhamento dinâmico
- Tarefas mínimas do encaminhamento dinâmico:
  - Enviar informação de acessibilidades a outros routers
  - Processar a informação de acessibilidades recebida
  - Determinar caminhos óptimos e criar tabela de encaminhamento
  - Reagir a alterações à topologia
- Resultado final deve ser consistente entre todos os routers
  - Convergência

# Encaminhamento dinâmico — métricas

- Conceito não relacionado com a distância administrativa
- Permitem atribuir valores numéricos aos diferentes caminhos possíveis para escolher o melhor (sob algum ponto de vista)
- Podem ser
  - Estáticas — características que não se alteram ao longo do tempo
  - Dinâmicas — variáveis ao longo do tempo
    - Podem conduzir a oscilações de rotas e tornar a rede instável
- Exemplos de métricas
  - Número de saltos
  - Capacidade
  - Carga — métrica dinâmica
  - Atraso
  - Fiabilidade
  - Custo (métrica genérica sem dimensões)

# Endereçamento IPv4 com classes

Classe	Primeiros bits	Nº de bits na parte de rede (prefixo)	Nº de bits na parte de máquina	Nº de redes da classe	Nº de endereços por rede	Primeiro endereço	Último endereço
A	0...	8	24	$126^1$ ( $2^7-2$ )	16 777 216 ( $2^{24}$ )	1.0.0.0	126.255.255.255
B	10...	16	16	16 384 ( $2^{14}$ )	65 536 ( $2^{16}$ )	128.0.0.0	191.255.255.255
C	110...	24	8	2 097 152 ( $2^{21}$ )	256 ( $2^8$ )	192.0.0.0	223.255.255.255
D (multicast)	1110...	—	—	—	—	224.0.0.0	239.255.255.255
E (reservada)	1111...	—	—	—	—	240.0.0.0	255.255.255.255

<sup>1</sup> Só há 126 redes classe A porque a rede 0 é inválida e a 127 é reservada para *loopback*

- Os primeiros bits determinam a classe
- A classe determina a máscara (comprimento de prefixo)

# Encaminhamento com e sem classes

- Protocolos de encaminhamento com classes (*classful*)
  - Não enviam o comprimento dos prefixos anunciados
  - Máscara de rede *classful* determinada pelos primeiros bits de cada prefixo
  - Possibilidades de *subnetting*<sup>†</sup> severamente limitadas
  - *Supernetting*<sup>‡</sup> impossível
- Protocolos de encaminhamento sem classes (*classless*)
  - Comprimento de cada prefixo é enviado nos anúncios
  - Permite *subnetting* / *supernetting* sem restrições

<sup>†</sup>Subdivisão de redes *classful*

<sup>‡</sup>Agregação de redes *classful* contíguas