# Path control implementation - Policy based routing (PBR)

**ROUTE modul 5 -**

# V tejto kapitole

- Cisco Express Forwarding Switching
- Implementácia Path Control Using Policy-Based Routing
- Implementácia Path Control Using Cisco IOS IP SLAs

# Čo sa deje pri smerovaní?
-
# Smerovanie do detailu

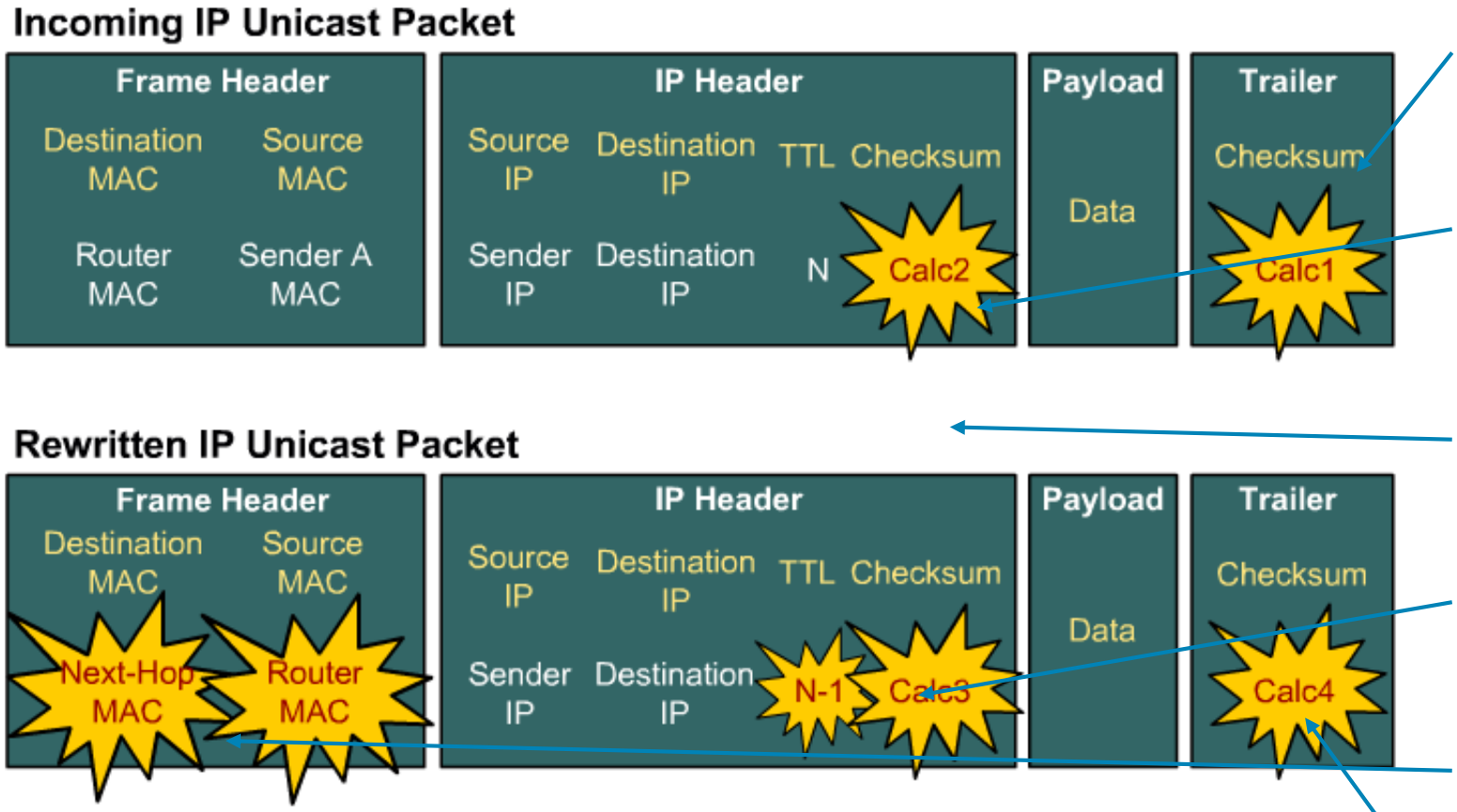
# Riadenie smerovania - Control a data plane L3 zariadení

- L3 zariadenia majú vnútorne distribuovanú architektúru
  - Control plane
    - Zodpovedná za výmenu a budovanie smerovacej informácie
    - Vykonávané tzv. *route processorom*
  - Data forwarding plane
    - Preposielanie paketov
    - Vykonávané procesorom na rozhraní (micro code controller)
  - Vzájomne prepojených rozhraním

# Control a data plane L3 zariadení

- Funkcie riadiaceho rozhrania smerovania medzi vrstvou smerovacieho protokolu a mikrokódom dataplane
  - Riadenie interných dát a riadiacich obvodov pre riadiace funkcie a data plane prepínacie funkcie
  - Extrakcia a príprava všetkých informácii nevyhnutných pre smerovanie
    - L2 bridging (napr. ARP), Out rozhranie,
    - L3 routing (next hop IP)
  - Doručenie týchto informácii na modul ovládania rozhrania pre riadenie preposielania
  - Spracovávania a doručovania špeciálnych paketov na route processor
- Existujú rôzne implementácie tohto rozhrania

# Čo sa deje s paketom pred a po smerovaní?




1. Kontrola FCS prijatého rámca

2. Deenkaps . payload a kontrola zavezpeč. IP paketu

3. Vykonané smerovacie rozhodnutie

4. TTL-1, výpočet CRC pre IP hlavičku

5. Vytvor frame, vyplň adresy

6. Vypočítaj FCS pre frame a pošli outgoing

# Z akých krokov sa skladá unicastové smerovanie v IPv4 sieťach?

1. Má rámec korektnú veľkosť a je jeho FCS správne?
   - Ak áno, pokračujeme ďalšími krokmi
   - Ak nie, rámec zahodíme bez pokračovania
2. Má hlavička IP paketu správnu kontrolnú sumu?
   - Ak áno, pokračujeme ďalšími krokmi
   - Ak nie, paket zahodíme bez pokračovania
3. Je paket podľa IP adresy príjemcu určený pre lokálnu IP adresu?
   - Ak áno, patrí samotnému routeru – nebudeme ho smerovať
   - Ak nie, pokračujeme ďalšími krokmi
4. Je hodnota TTL v hlavičke IP paketu väčšia ako 1?
   - Ak áno, pokračujeme ďalšími krokmi
   - Ak nie, paket zahodíme a odosielateľa upozorníme ICMP správou

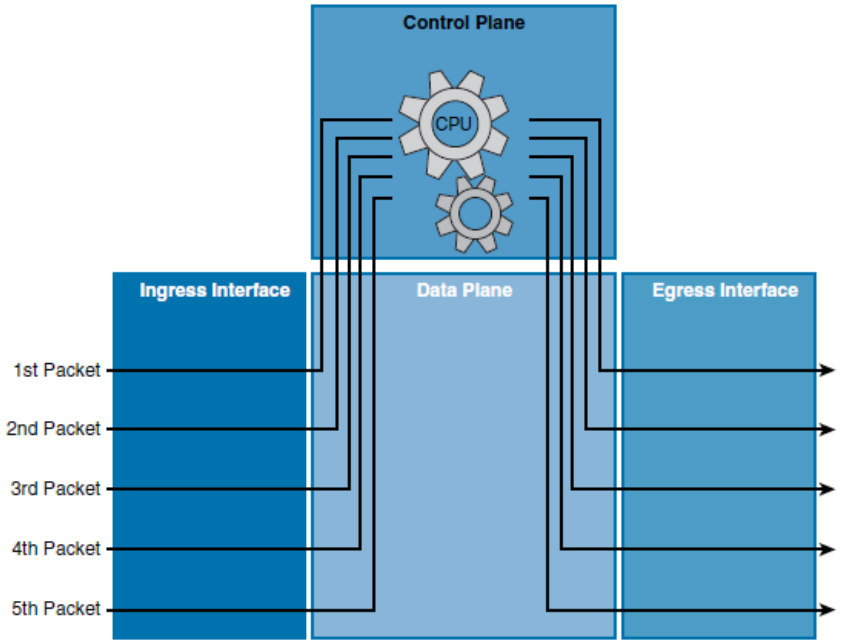
# Z akých krokov sa skladá unicastové smerovanie v IPv4 sieťach?

5. K IP adrese príjemcu vyhľadáme v smerovacej tabuľke vyhovujúci záznam
   - Smerovacia tabuľka je usporiadaná zostupne podľa masiek
   - Hľadáme prvý riadok tabuľky, v ktorom platí:
     - IP adr. príjemcu & Sieťová maska = Číslo siete
   - Ak nie, paket zahodíme a odosielateľa upozorníme ICMP správou
6. Ukazuje vyhľadaný riadok smerovacej tabuľky na východzie rozhranie, ktorým má paket odísť?
   - Ak áno, pokračujeme ďalším krokom
   - Ak nie, potom obsahuje IP adresu next hop routera. Tú si zapamätáme a s ňou sa vrátime na krok 5
7. K výstupnému rozhraniu a naposledy zapamätanej IP adrese next hop routera vyhľadáme Layer2 informáciu pre vytvorenie rámca
   - ARP tabuľka pre Ethernet, Mapovacia tabuľka pre Frame Relay

# Z akých krokov sa skladá unicastové smerovanie v IPv4 sieťach?

8. V IP pakete dekrementujeme pole TTL a prepočítame kontrolnú sumu
9. IP paket enkapsulujeme do príslušného rámca pomocou informácií vyhľadaných v kroku 7
   a odošleme rozhraním vyhľadaným v kroku 6

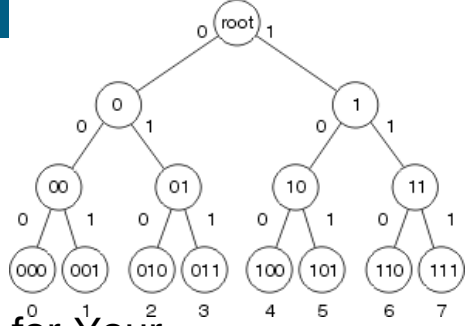
Pozn.
- Vykonávam pre každý paket zvlášť
- Tento spôsob riadenia smerovania sa nazýva „**Process routing or Process switching**“
- Najpomalší spôsob smerovania
- Rýchlosť spracovania závisí od architektúry, od zaťaženia CPU a ten závisí od počtu vstupujúcich paketov



# Efektívnosť smerovania

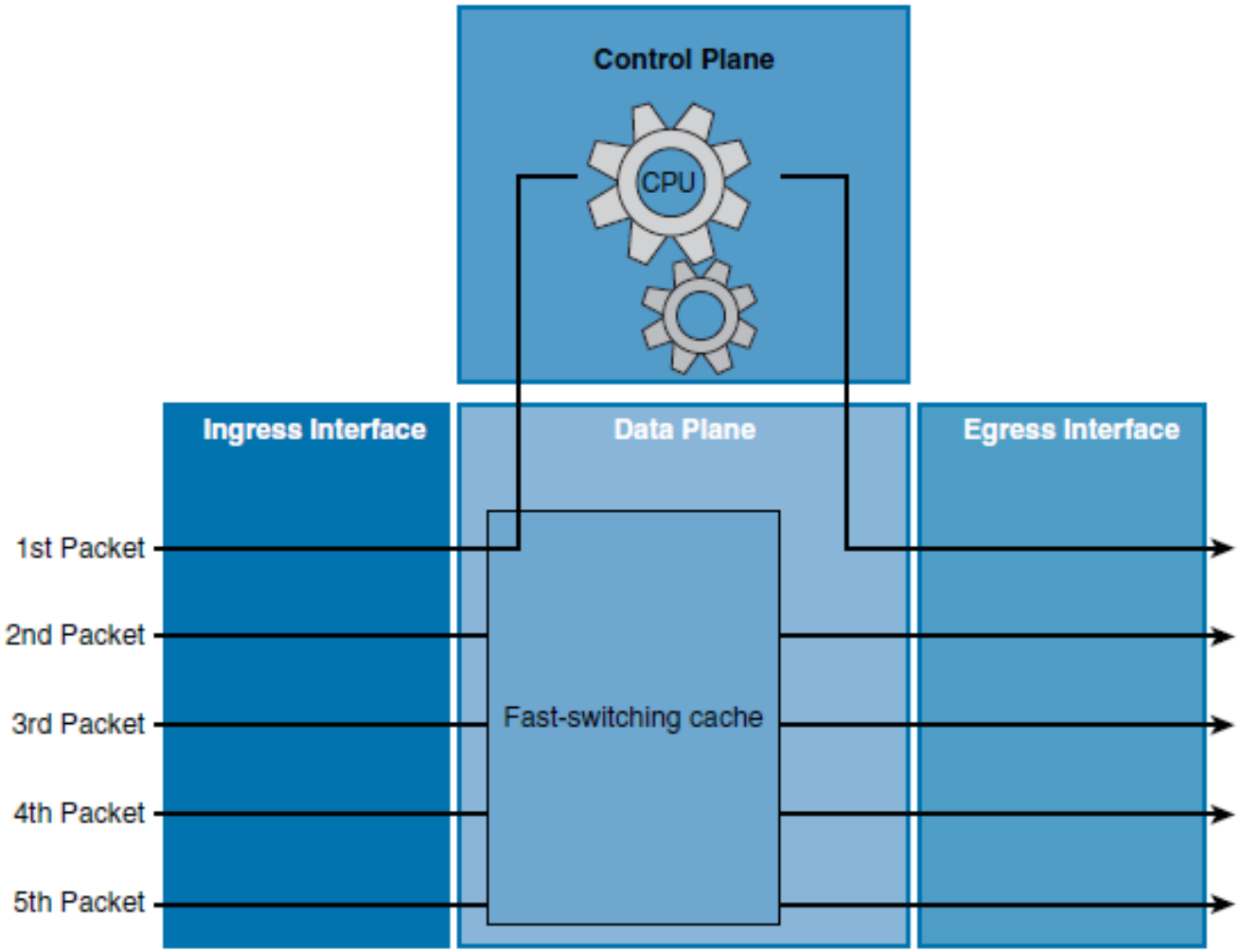
- Opísaný algoritmus pre jednoduchosť nerieši špecifické situácie
  - Fragmentácia IP paketov
  - ACL
  - Tunelovanie, šifrovanie/dešifrovanie
  - Preklad adries a mnohé ďalšie osobitné operácie
- Jeho najzdĺhavejšie kroky sú 5. – 7.
  - Vyhľadávanie rôznych informácií v rôznych databázach pre každý jeden paket, ktorý musíme smerovať
  - Na kontrolné sumy sa dá navrhnúť špecializovaný integrovaný obvod
- Veľké úsilie sa v posledných rokoch venovalo práve tomu, ako smerovaču uľahčiť život a ako tieto operácie zjednodušiť a tým **urýchliť**
  - Existujú viaceré vylepšenia
    - **Process switching**, **Fast switching**, Autonomous switching, Silicon switching engine (SSE) switching, Optimum switching, Distributed fast switching, **Cisco Express Forwarding (CEF)**, Distributed Cisco Express Forwarding (dCEF)

# Fast Switching




- **Document ID: 13706** , „How to Choose the Best Router Switching Path for Your Network“
- **Fast Switching** je takisto známy ako *route cache (NetFlow switching, Topology switching, apod. )*
- Idea:
  - Prvý paket idúci do istého cieľa prejde pôvodným algoritmom
    - Tak isto ako pri Process switching
  - Výsledok tohto algoritmu (výstupné rozhranie) sa však zapamätá v tzv. route cache (or Fast Cache)
    - Špeciálna dátová štruktúra s rýchlym prístupom a vyhľadávaním
    - Viacúrovňový binárny (32 úrovní) alebo znakový (256 vetiev a štyri úrovne) strom
  - Každý ďalší paket do toho istého cieľa môže využiť predpripravené informácie z route cache
    - ***„Route once, forward many times“***
- Nevýhody:
  - Route cache sa tvorí iba tokom paketov (príchodom prvého)
  - Jej položky pre potreby synchronizácie je potrebné priebežne nechávať exspirovať (5% náhodne každú minútu)
  - Zmena v ARP tabuľkách znamená prebudovanie časti route cache

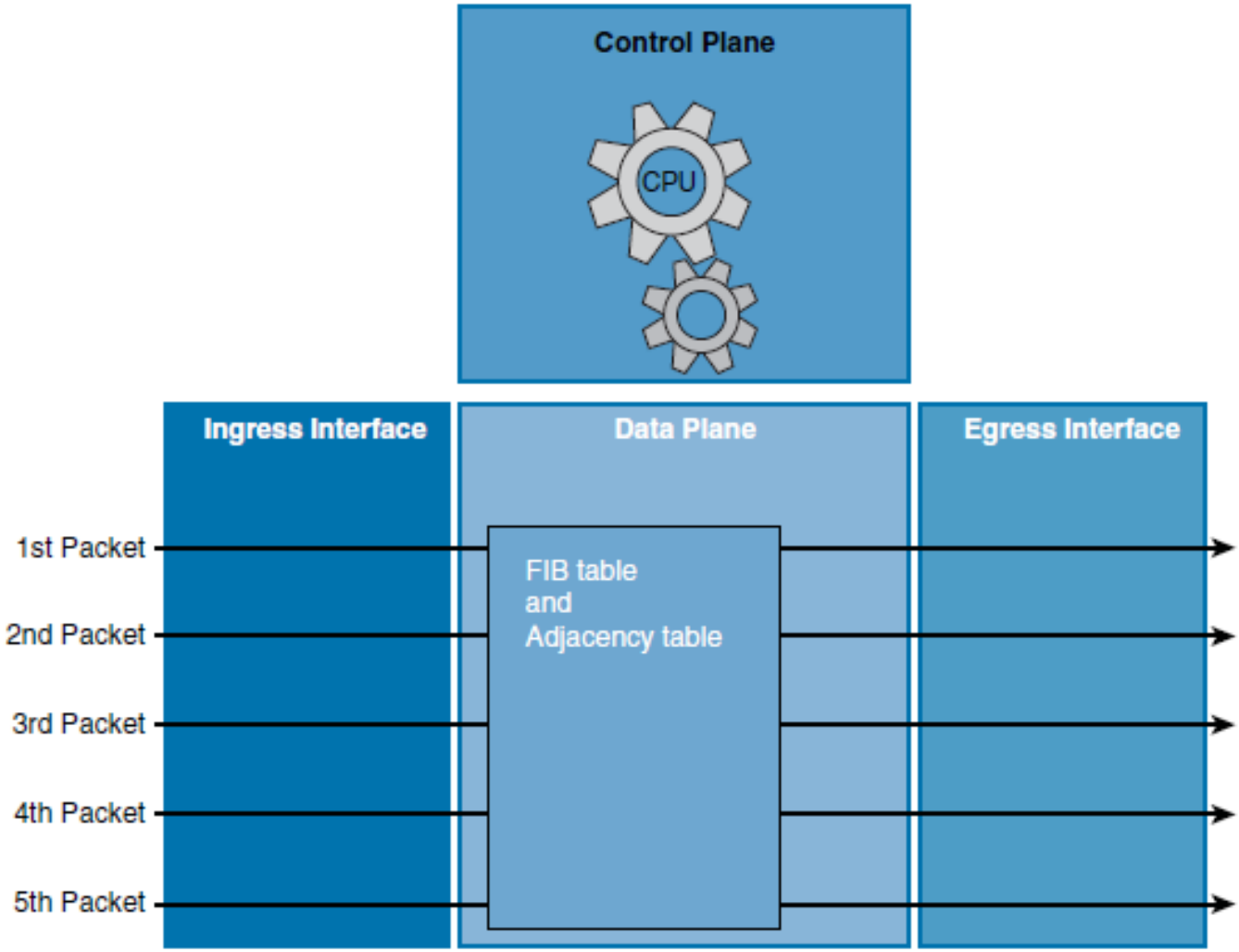
# Fast switching



# Cisco Express Forwarding (CEF)

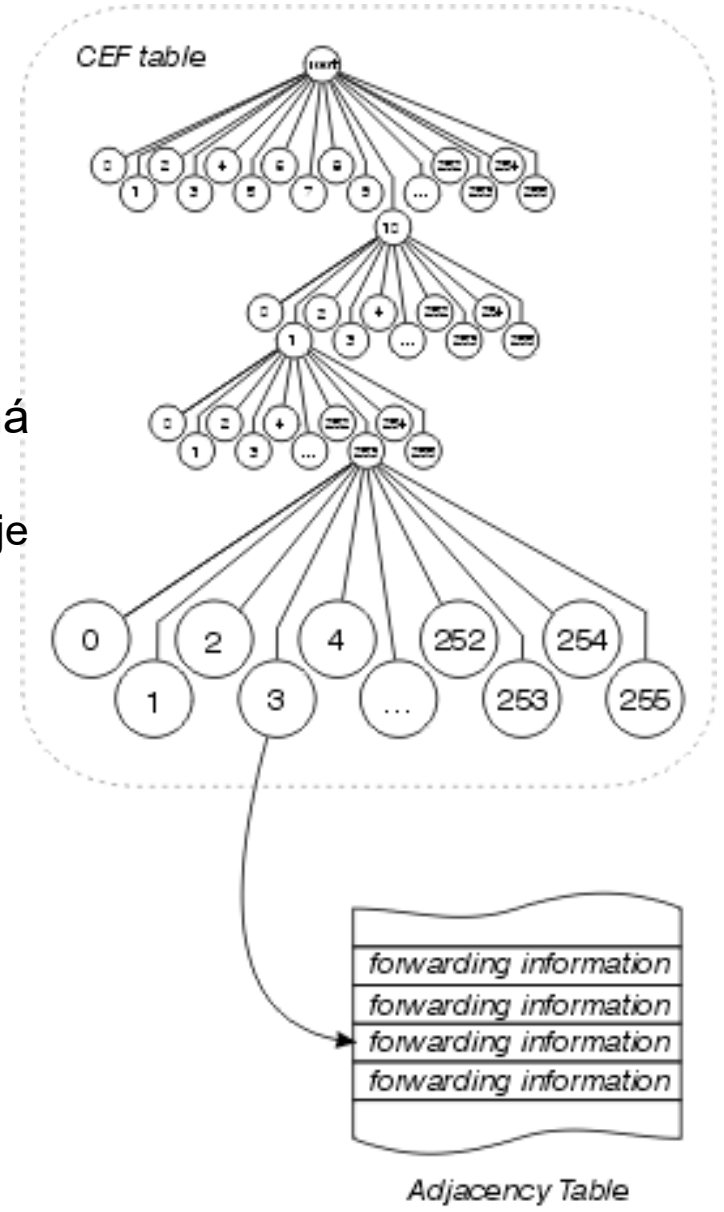
- CEF je ďalším vylepšením route cache (topology based switching)
- Idea:
  - Zorganizovať si položky zo smerovacej tabuľky do samostatnej dátovej štruktúry, tzv. **Forwarding Information Base (FIB)**, v ktorej sa dá veľmi rýchlo vyhľadávať (one to one mapping)
  - Informácie o prepise rámca si predpripraviť ihneď, ako je to možné, a organizovať si ich v tzv. **adjacency databáze**
    - DB L2 adries next hopov (priamo dosiahnuteľných susedov a ReWrite info) budovaná z napr. ARP, inARP a pod.
  - Jednotlivé položky vo FIB budú pomocou smerníkov ukazovať na položky v adjacency DB
- Vlastnosti:
  - Striktné oddelenie „Control plane HW“ od „Data plane HW“
    - Control Plane: Budovanie FIB a ADJ
    - Data Plane HW: rýchly switching
  - FIB a adjacency DB sa vytvárajú z existujúcich položiek (hotová smerovacia tabuľka, hotové L2 informácie)
  - **Zmena** v adjacency DB si **nevyžaduje zmeny** vo FIB

# Cisco Express Forwarding



# Cisco Express Forwarding

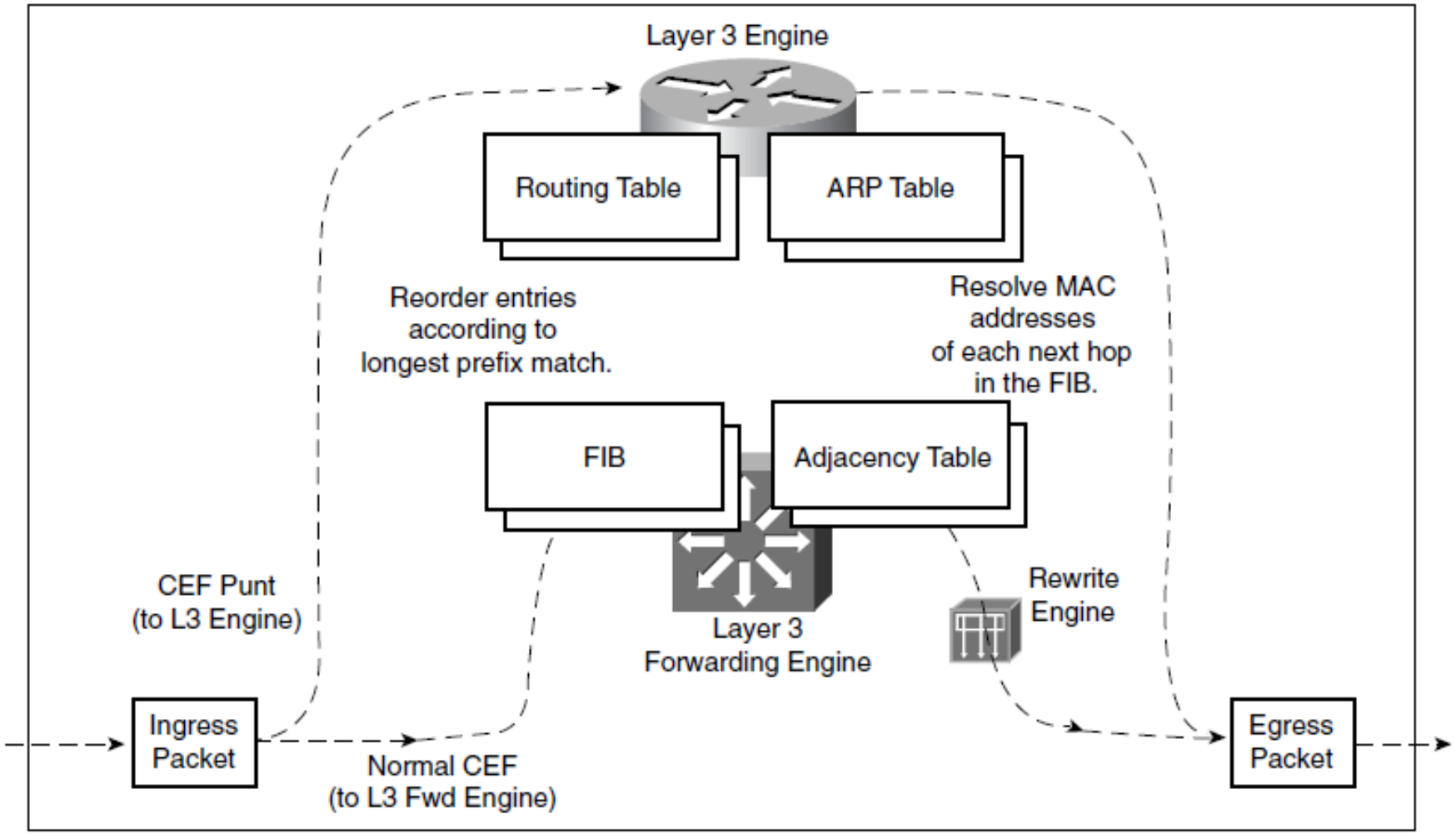
- Existujú dve realizácie CEF
  - Softvérová
  - Hardvérová
- **Softvérová** realizácia CEF
  - Stromová časť je FIB, principiálne sa jedná o tzv. znakový strom (trie resp. mtrie)
    - Trie or mtrie znamená, že strom neobsahuje hľadané dáta ale pointer na ne
    - Obsahuje aj next hop IP add. a VLAN info
  - Tabuľka obsahuje adjacency DB
  - Obe štruktúry sú v RAM
  - Informácia o prepise sa priamo vo FIB nenachádza
  - Softvérové CEF využívajú najmä **bežné Cisco smerovače**



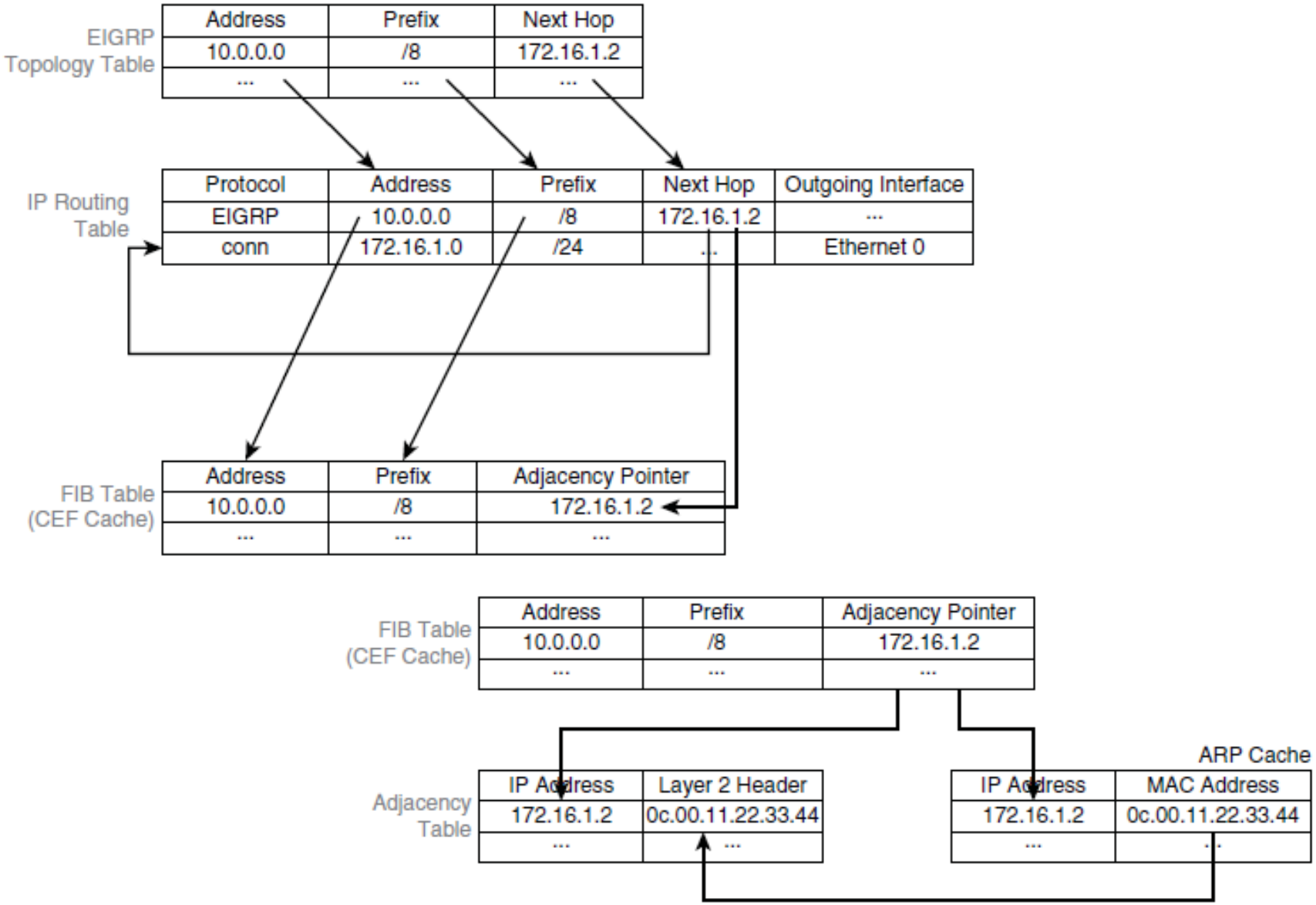
# Cisco Express Forwarding

- **Hardvérová** implementácia CEF využíva špecializované integrované obvody na uchovávanie FIB
  - Ternary Content Addressable Memory (TCAM)
- CAM a TCAM sú dva typy cache pamätí
  - CAM obvykle obsahuje informácie pre L2 switching
  - TCAM obsahuje informácie pre L3 switching, ACL, QoS
- CAM hľadá presne zadaný reťazec (úplná zhoda)
  - Používa bity 0 (true ) a 1 (false)
  - Vhodné pre „exact match“ vyhľadávanie
- TCAM hľadá buď najdlhší zhodný alebo prvý zhodný reťazec
  - Je možné povedať, ktoré bity reťazca nás nezaujímajú
    - Vstup 0, 1, a X (don’t care) bit values = *ternary* combination.
  - Môže byť rozdelená na regióny s rôznou politikou vyhľadávania
- Využitie TCAM na udržiavanie FIB je typickou doménou **multilayer prepínačov (MLS) a high-end smerovačov**
- Catalyst môže bez degradácie rýchlosti spraviť viaceré CAM a TCAM dotazy paralélne

# Tok paketov cez CEF MLS



# Cisco Express Forwarding



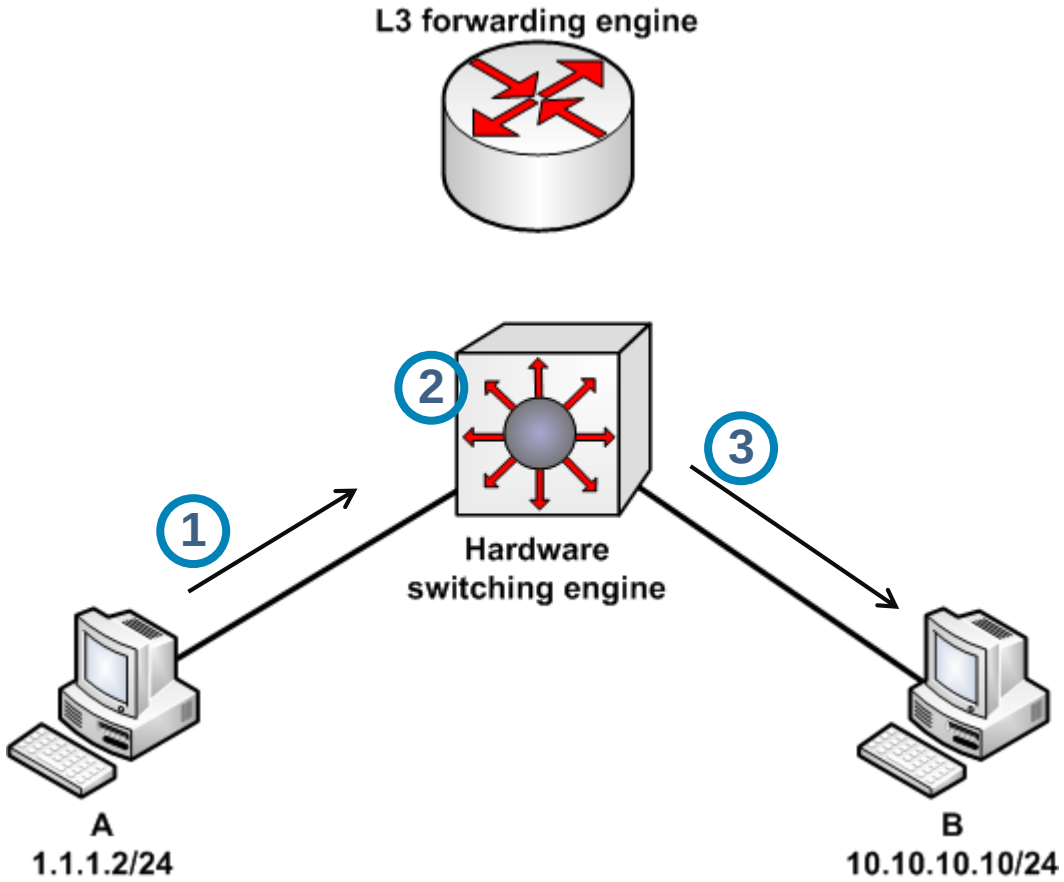
# Činnosť MLS

• Predpokladom je správna informácia vo *FIB* a správna rewrite informácia v *Adjacency Table*

3 Router prepíše požadované polia v pakete a rámci, a prepne paket na hosta B

2 Router na základe MAC cieľovej adresy zistí, že rámec je jeho a má byť smerovaný. Router vykoná CEF lookup pre IP hosta B. Z FIB vytiahne adj položku s rewrite údajmi.

1 Host A pošle unicast packet na host B



# Cisco Express Forwarding

- Zapnutie CEF:

```
Router(config)# ip cef
```

  - Na multilayer prepínačoch nie je možné CEF vypnúť

- Aktivácia/deaktivácia CEF na rozhraní

```
Router(config)# int fa0/1
Router(config-if)# [no] ip route-cache cef
```

  - CEF sa zásadne aktivuje alebo deaktivuje na vstupnom rozhraní
  - Route-cache sa aktivuje alebo deaktivuje na výstupnom rozhraní

- Zobrazenie informácií vo FIB a ADB

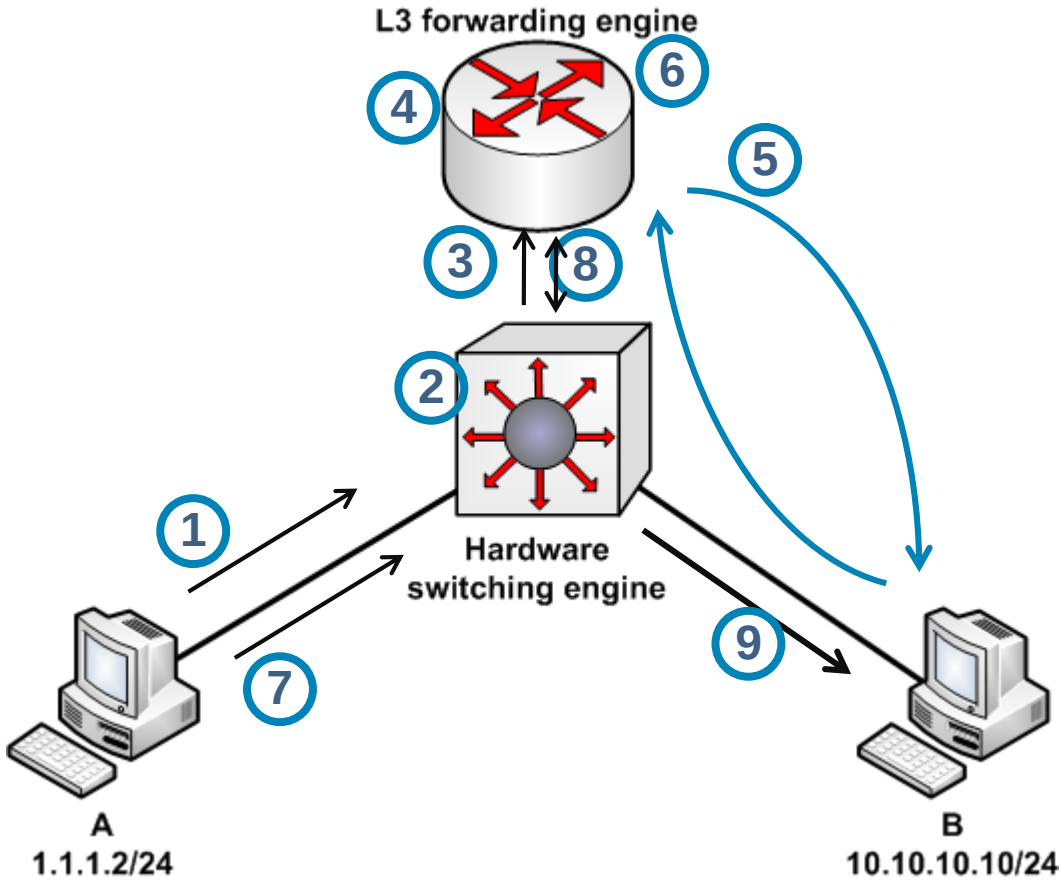
```
Router# show ip cef
Router# show adjacency
```

# Príklad CEF

```
sw-vd-FRI#show ip cef 194.160.136.5 detail
194.160.136.0/24, epoch 1
  nexthop 158.193.26.1 Vlan26

sw-vd-FRI#show adjacency 158.193.26.1 detail
Protocol  Interface                 Address
IP        Vlan26                    158.193.26.1(11)
                                    2 packets, 116 bytes
                                    epoch 0
                                    sourced in sev-epoch 88
                                    Encap length 14
                                    00E04C38C6D5001B8F8FDE410800
                                    ARP
sw-vd-FRI#show ip arp 158.193.26.1
Protocol  Address          Age (min)  Hardware Addr   Type   Iface
Internet  158.193.26.1            0   00e0.4c38.c6d5  ARPA   Vlan26
```

# ARP throttling

1. Host A pošle unicast packet na host B
2. MLS prepínač na základe MAC cieľovej adresy zistí, že rámec je jeho a má byť smerovaný. MLS vykoná CEF lookup pre IP hosta B. Z FIB vytiahne adj GLEAN – adj neexistuje
3. MLS pošle paket do L3 engine na L3 processing. HW engine nemôže poslať paket ďalej, chýba rewrite informácia
4. L3 engine nastaví v adj. DB ***ARP throttling*** stav pre IP adresu hosta B, pretože chýba ARP informácia potrebná na prepis. A a B ešte spolu nekomunikovali
5. L3 engine pošle ARP request na IP hosta B. Nastaví adj DROP. Host B odpovie ARP reply
6. L3 engine doplní chýbajúcu adjacency informáciu do ADJ DB, odstraní drop adjacency
7. Host A pošle nový unicast paket na host B
8. HW engine vykoná CEF lookup, nájde požadovanú adj položku
9. MLS prepíše odpovedajúce položky (zdrojová a cieľová MAC adresa), vypočíta CRC a prepne paket na hosta B



# Cisco Express Forwarding

- Nie všetky pakety môžu byť spracované v CEF
- Cisco Express Forwarding nepodporuje:
  - Pakety, pre ktoré v CEF neexistuje platný záznam
  - Pakety určené pre samotný router resp. switch
  - Broadcasty a multicasty
  - IP pakety, ktoré využívajú voliteľné časti hlavičky (options)
  - IP pakety, ktoré musia byť fragmentované
  - IP pakety, ktorým expiruje TTL
  - Network Address Translation
  - Šifrovanie počas prenosu

# Cisco Express Forwarding

- Podľa verzie Cisco zariadenia môže CEF pracovať v **centralizovanom** alebo **distribuovanom** režime (dCEF)
  - V **centralizovanom** režime sa všetky rozhodnutia o forwardovaní paketu realizujú nad spoločnou centralizovanou databázou a centrálnym ASIC pre všetky rozhrania
    - Príklad 4500 a 6500 bez Distributed Forwarding Cards (DFC)
    - Hardware switching je realizovaný cez centrálny forwarding engine a prepínaciu maticu (bus switching)
  - V **distribuovanom** režime sa vybrané časti CEF štruktúr nahrávajú do procesorov na zásuvných moduloch, ktoré potom vedia realizovať forwarding vo svojej vlastnej réžii
    - CEF v distribuovanom režime sa nazýva **dCEF**
    - 6500 so Supervisot Engine (720) a DFC modulmi

# Riadenie smerovacích ciest

# Riadenie smerovacích ciest

- Fokus tejto kapitoly je zameraný na spôsoby riadenia ciest, ktoré budú vyberané pre sieťovú prevádzku
  - Aj keď v niektorých prípadoch je len jedna cesta ☺
- Väčšina komplexnejších sietí má implementované redundantné pripojenia, ktoré môžeme manažovať
- Výber optimálnej cesty v takomto prostredí ovplyvňujú viaceré faktory
  - Nasadenie daného smerovacieho protokolu je jeden z faktorov, ktoré definujú ako sa bude diať výber ciest
    - Rozdielna AD, metriky, redistribúcia, filtrovanie apod.
  - Nasadenie viacerých však môže viesť k neoptimálnemu smerovaniu
- Ale v sieťach s redundanciou sú aj iné faktory ktoré treba brať do úvahy pre optimálny routing

# Charakteristiky sietí pri ich návrhu (1)

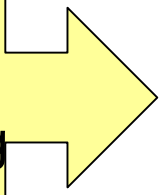
- Resiliency (odolnosť)
  - Schopnosť poskytovať službu na prijateľnej úrovni aj v prípade, keď v sieti nastanú výpadky
  - Redundancia automaticky neznamená resiliency
  - Prostriedky: fail-over, load balancing
- Availability (dostupnosť)
  - Schopnosť rýchlo ošetriť výpadok v sieti a nájsť záložnú trasu
  - Prostriedky: vyladené, rýchlo konvergujúce smerovacie protokoly
- Adaptability (prispôsobivosť)
  - Schopnosť dynamicky prispôsobiť činnosť siete podľa aktuálneho stavu, napríklad aktivovať záložnú linku aj v prípade vysokej záťaže
- Performance (výkonnosť)
  - Schopnosť využívať existujúce prostriedky siete pre poskytovanie čo najvyššieho výkonu
  - Prostriedky: load balancing, prípadne selektívna modifikácia metrík

# Charakteristiky sietí pri ich návrhu (2)

- Support for network and application services (Podpora pre sieťové a aplikačné služby)
  - Špecifické prispôsobenie smerovania pre vybrané služby
  - Prostriedy: QoS nástroje, WAAS (Wide Area Application Services), bezpečnostné nástroje
- Predictability (Predpovedateľnosť)
  - Ohľad na deterministické a predpovedateľné správanie, napríklad obojsmernosť tokov a rovnaká cesta, ktorou pôjdu
- Asymmetric traffic (Asymetrická prevádzka)
  - Upstream tok prechádajúci inou cestou ako downstream
  - Nie vždy neželané – existujú aplikácie, kedy je tento prístup potrebný (napr. satelitné prepoje: upstream je modem, downstream je satelit)

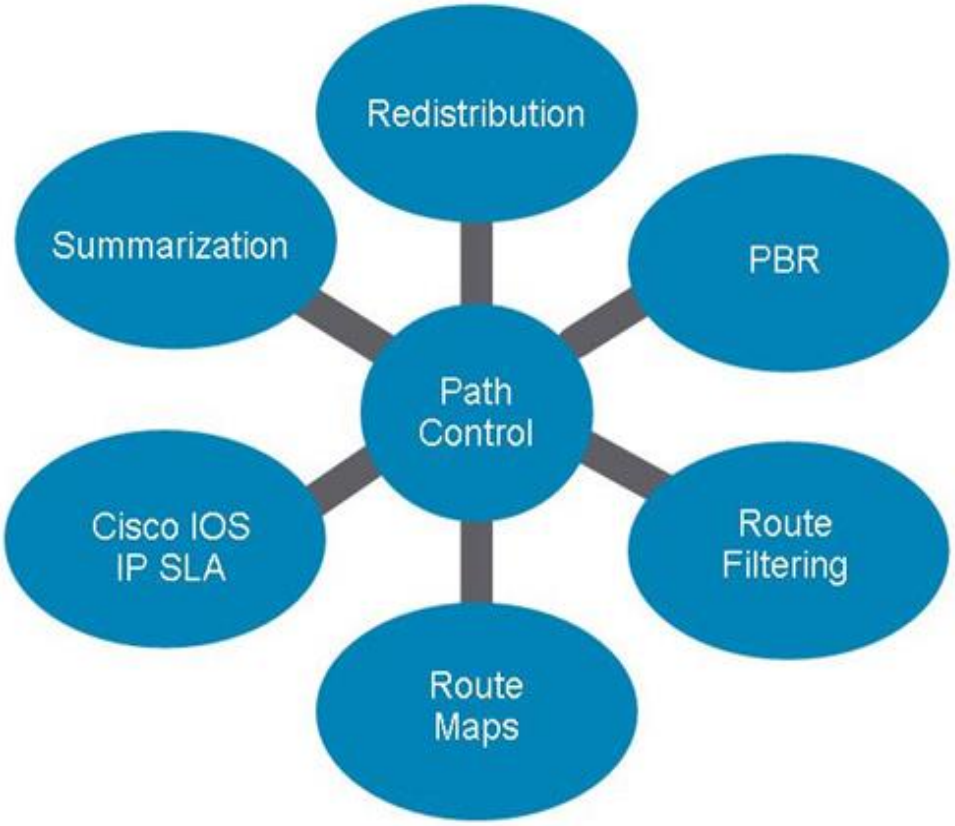
# Dostupné nástroje na riadenie ciest

- Už spomenuté:
  - Passive interfaces
  - Sumarizácia
  - Distribute lists
  - Prefix lists
  - Administrative distance
  - Route maps
  - Route tagging
- Pokročilé nástroje
  - Offset lists
  - Cisco IOS IP SLAs
  - Policy Based Routing



Fokus tejto kapitoly

# Stratégia




- Každý z týchto nástrojov môže byť súčasťou integrovanej stratégie na implementáciu riadenia výberu
- Preto je nevyhnutné pred začatím implementácie
  - Takúto stratégiu naplánovať a vypracovať

# Implementácia riadenia ciest pomocou Offset Lists

# Offset List

- Offset list je prostriedok na umelé zvýšenie metriky vybraných sietí len v protokole RIP alebo EIGRP
  - Môžeme zvýšiť metriku pre oznamované i prijímané siete
  - Offset list môže byť použitý pre celý smerovací protokol, alebo pre siete odosielané/prijímané konkrétnym rozhraním
- Offset-list sa vytvára príkazom **`offset-list`** a odkazom na ACL
  - Stanovená hodnota offsetu sa pripočíta k metrike (RIP) alebo k parametru *Delay* (EIGRP)

# Vytvorenie offset-listu

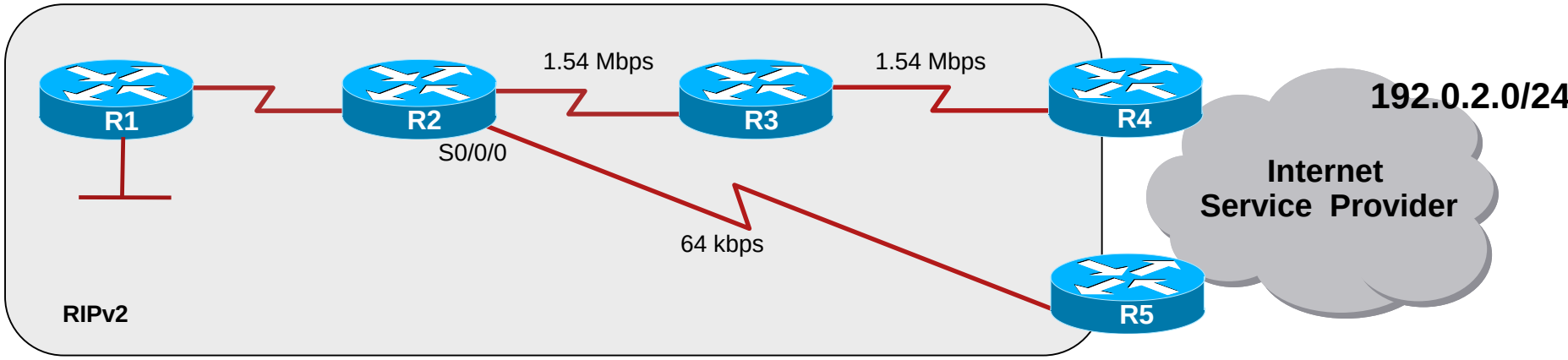
- Offset-list sa vytvára v smerovacom protokole

```
Router(config-router)#
offset-list {access-list-number | access-list-name} {in | out}
  offset [interface-type interface-number]
```

| Parameter | Description |
|---|---|
| *access-list-number* \| *access-list-name* | ACL, ktoré vyberá siete, ktorým sa má zvýšiť metrika. Číslo 0 znamená „všetky siete“. |
| **in** | Aplikuje offset list na siete v prichádzajúcich správach RIP/EIGRP |
| **out** | Aplikuje offset list na siete v odchádzajúcich správach RIP/EIGRP |
| *offset* | Hodnota, o ktorú sa metrika zvýši. Hodnota 0 znamená, že sa aktuálna hodnota metriky nezmení. |
| *interface-type interface-number* | (Nepovinné) Rozhranie, na ktoré je offset-list aplikovaný |

# Príklad - Použitie offset-listu

- Stanice na LAN za R1 môžu ísť na sieť 192.0.2.0/24 u ISP dvomi cestami
  - R5 je podľa metriky bližšie, ale je na podstatne pomalšej linke
- Pomocou offset-listu môžeme umelo na R2 zvýšiť metriku tejto siete, ako nám ju oznamuje R5, a zariadiť, aby preferovaná cesta bola cez R3
  - Offset-list v tomto príklade zvýši metriku 192.0.2.0/24 cez R5 o 2




```
R2(config)# access-list 21 permit 192.0.2.0 0.0.0.255
R2(config)# router rip
R2(config-router)# offset-list 21 in 2 serial 0/0/0
```

# Overenie Offset Lists

- **`traceroute`**
  - Overenie cesty, ktorou sú preposielané pakety.
- **`show ip route`**
  - Na overenie metriky postihnutých ciest
- Pre EIGRP **`show ip eigrp topology [all-links]`**
  - Na overenie topo tabuľky
- **`debug ip eigrp`** or **`debug ip rip`**.

# IP Policy Routing (PBR)

# Policy based routing (PBR)

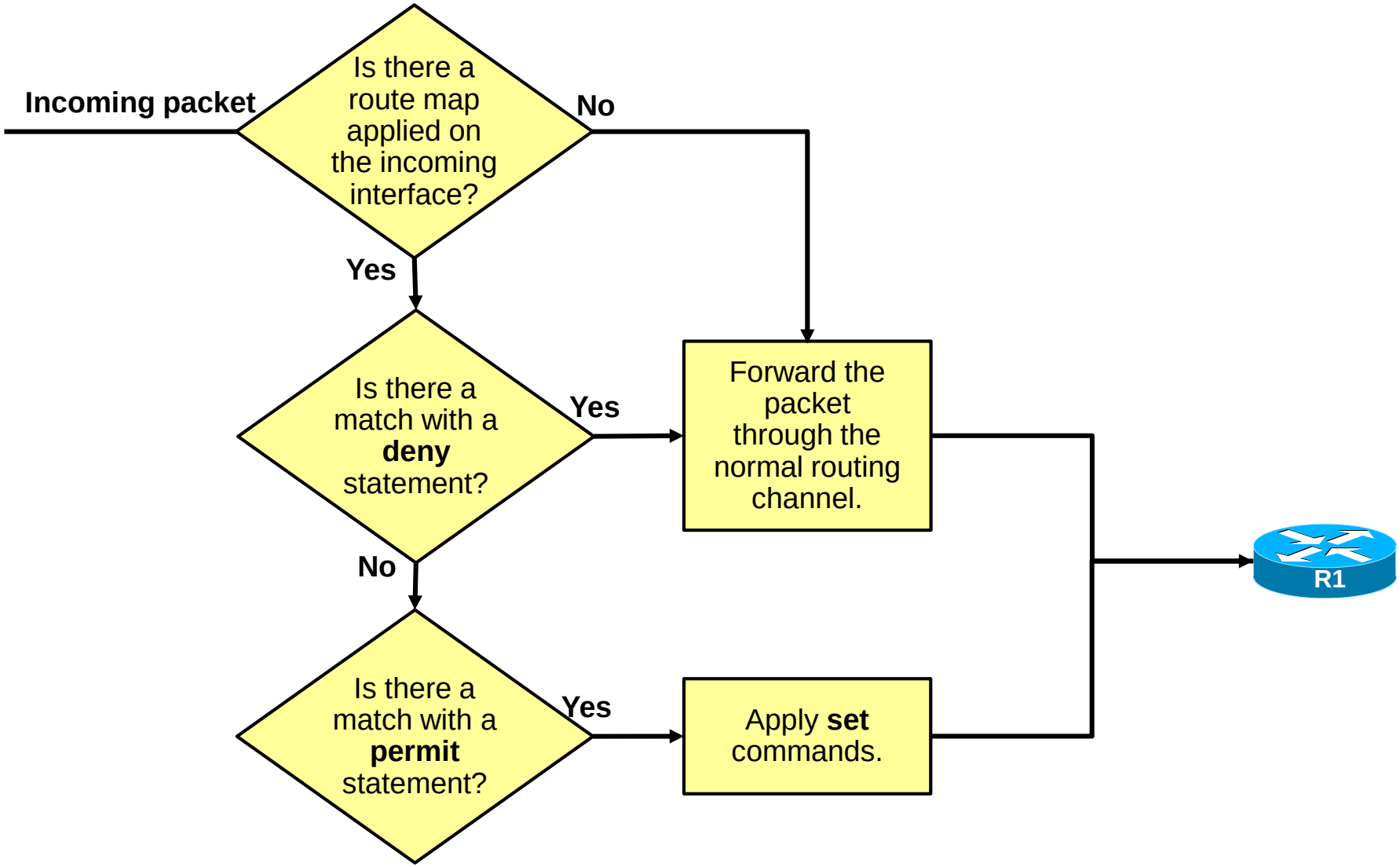
- PBR poskytuje
  - Smerovanie na základe politík alebo dodatočných kritérií (nielen cieľovej IP adresy)
    - Odosielateľ
    - Cieľ
    - Veľkosť paketu
    - Typ protokolu
  - Alebo techniky na značkovanie paketov (QoS)
- PBR prepisuje bežné smerovacie postupy
- Nasadenie PBR tam, kde existujú určité dôvody na smerovane prevádzky určitou cestou, inou ako smeruje smerovacia tabuľka
  - Znižovane nákladov za linku, QoS, výber tranzitného providera apod

# Policy based routing (PBR)

- Realizované pomocou route-map konštruktov
  - Vytvorenie route mapy
    - Akcia definuje, či budú pakety smerované pomocou policy routingu (permit) alebo podľa smerovacej tabuľky (deny)
    - Časť match vyberie pakety
  - Ak je akcia permit, časť set hovorí, ako sa paket prepošle ďalej
    - **`set ip next-hop, set interface, set ip default next-hop, set default interface`**
    - Set sa pri akcii deny nevyhodnocuje
    - Ak chceme dosiahnuť zahadzovanie paketov, je treba nastaviť set na rozhranie NULL
- Výsledný route-map sa aplikuje na vstupné rozhranie príkazom

```
Router(config-if)# ip policy route-map MENO
```

# Logické operácie pri PBR



# Route mapa – `match` kritéria využiteľné pre PBR

| Command | Description |
|---|---|
| `match community` | Matches a BGP community |
| `match interface` | Matches any routes that have the next hop out of one of the interfaces specified |
| **`match ip address`** | Matches any routes that have a source or destination network number address that is permitted by a standard or extended ACL |
| `match ip next-hop` | Matches any routes that have a next-hop router address that is passed by one of the ACLs specified |
| `match ip route-source` | Matches routes that have been advertised by routers and access servers at the address that is specified by the ACLs |
| **`match length`** | Matches based on the layer 3 length of a packet |
| `match metric` | Matches routes with the metric specified |
| `match route-type` | Matches routes of the specified type |
| `match tag` | Matches tag of a route |

# Príkaz `match ip-address`

- Špecifikuje porovnávacie kritéria, či už voči ACL alebo prefix listu

```
Router(config-route-map)#
```

**match ip address** *{ACCESS-LIST-NUMBER | NAME} [...ACCESS-LIST-NUMBER | NAME]* | **prefix-list** *PREFIX-LIST-NAME [..PREFIX-LIST-NAME]*

| Parameter | Description |
|---|---|
| *access-list-number \| name* | The number or name of a standard or extended access list to be used to test incoming packets.<br>If multiple access lists are specified, matching any one results in a match. |
| **prefix-list** *prefix-list-name* | Specifies the name of a prefix list to be used to test packets.<br>If multiple prefix lists are specified, matching any one results in a match. |

- Standard ACL definuje source = odkiaľ pakety tečú
- Exended ACL definuje source and destination = odkiaľ a kam pakety tečú

# Príkaz `match length`

- Porovnávanie na dĺžku paketu

```
Router(config-route-map)#
```

```
match length min max
```

| Parameter | Description |
|---|---|
| *min* | The packet's minimum Layer 3 length, inclusive, allowed for a match. |
| *max* | The packet's maximum Layer 3 length, inclusive, allowed for a match. |

# Route mapa – akcie `set` využiteľné pre PBR

| Command | Description |
|---|---|
| `set as-path` | Modifies an AS path for BGP routes |
| `set automatic-tag` | Computes automatically the tag value |
| `set community` | Sets the BGP communities attribute |
| **`set ip next-hop`** | Indicates where to output packets that pass a match clause of a route map for policy routing |
| **`set interface`** | Indicates where to output packets that pass a match clause of a route map for policy routing |
| **`set ip default next-hop`** | Indicates where to output packets that pass a match clause of a route map for policy routing and for which the Cisco IOS software has no explicit route to a destination |
| **`set default interface`** | Indicates where to output packets that pass a match clause of a route map for policy routing and have no explicit route to the destination |
| **`set ip tos`** | Used to set some of the bits in the IP ToS field in the IP packet. |
| **`set ip precedence`** | set the 3 IP precedence bits in the IP packet header. |
| `set tag` | Sets tag value for destination routing protocol |
| `set weight` | Specifies the BGP weight value |

** Partial list*

# Príkaz `set ip next-hop`

- Specify the next hop IP address for matching packets.

```
Router(config-route-map)#
```

```
set ip next-hop IP-ADDRESS [...IP-ADDRESS]
```

- The command provides a list of IP addresses used to specify the adjacent next-hop router in the path toward the destination to which the packets should be forwarded.
- If more than one IP address is specified, the first IP address associated with a currently up connected interface is used to route the packets.

# Príkaz `set interface`

- Specify interfaces through which packets can be routed out.

```
Router(config-route-map)#
```

```
set interface TYPE NUMBER [... TYPE NUMBER]
```

- If more than one interface is specified, the first interface that is found to be up is used to forward the packets.

# Príkaz `set ip default next-hop`

- Specify a list of default next-hop IP addresses.

```
Router(config-route-map)#
```

```
set ip default next-hop IP-ADDRESS [...IP-ADDRESS]
```

- A packet is routed to the next hop specified by the **set** command only if there is *no* explicit route for the packet's destination address in the routing table.
  - A default route in the routing table is not considered an explicit route for an unknown destination address.
- If more than one IP address is specified, the first next hop specified that appears to be adjacent to the router is used.
- The optional specified IP addresses are tried in turn.

# Príkaz `set default interface`

- Specify a list of default interfaces.

```
Router(config-route-map)#
```

```
set default interface TYPE NUMBER [...TYPE NUMBER]
```

- If no explicit route is available to the destination address of the packet being considered for policy routing, it is routed to the first up interface in the list of specified default interfaces.

# Význam slova *Default* - zhrnutie

- Bez slova default
  - Smeruje najprv pomocou PBR
  - A ak výstupné rozhranie alebo next hop nie je dostupný
    - použi bežné smerovanie (RIB routing)
- So slovom default
  - Najprv skontroluj či pre paket vybratý match nemáš explicitné cesty v RIB
  - Ak áno
    - Použi RIB
  - Ak nie
    - Použi PBR

# Príkaz `set ip tos`

- Mark packets using the IP ToS field.

```
Router(config-route-map)#
```

```
set ip tos [NUMBER | NAME]
```

- Used to set some of the bits in the IP ToS field in the IP packet.
  - The ToS field in the IP header is 8 bits long, with 5 bits for setting the class of service (CoS) and 3 bits for the IP precedence.
  - The CoS bits are used to set the delay, throughput, reliability, and cost.

| Parameter | Description |
|---|---|
| **0 \| normal** | Sets the normal ToS |
| **1 \| min-monetary-cost** | Sets the min-monetary-cost ToS |
| **2 \| max-reliability** | Sets the max reliable ToS |
| **4 \| max-throughput** | Sets the max throughput ToS |
| **8 \| min-delay** | Sets the min delay ToS |

# Príkaz `set ip precedence`

- Set the 3 IP precedence bits in the IP packet header.

```
Router(config-route-map)#
```

```
set ip precedence  [NUMBER | NAME]
```

- This command is used when implementing QoS and can be used by other QoS services, such as weighted fair queuing (WFQ) and weighted random early detection (WRED).
- With 3 bits, you have 8 possible values for the IP precedence; values 0 through 7 are defined.

# Parametre `set ip precedence`

| Parameter | Description |
|---|---|
| `0 \| routine` | Sets the routine precedence |
| `1 \| priority` | Sets the priority precedence |
| `2 \| immediate` | Sets the immediate precedence |
| `3 \| flash` | Sets the Flash precedence |
| `4 \| flash-override` | Sets the Flash override precedence |
| `5 \| critical` | Sets the critical precedence |
| `6 \| internet` | Sets the internetwork control precedence |
| `7 \| network` | Sets the network control precedence |

# Local PBR

- PBR sa vždy aplikuje na pakety v *incoming* smere
  - route map vyhodnocuje pakety vstupujúce cez dané rozhranie

```
Router(config-if)# ip policy route-map MAP-TAG
```

  - T.j. lokálne pakety zasielané priamo smerovačom nie sú smerované PBR
  - Riešenie aj pre lokálnym smerovačom odoslané pakety

```
# v GKR

Router(config)# ip local policy route-map MAP-TAG
```

- V IOS od 12.0 môže byť zapnuté IP PBR v spracovaní *fast switching*

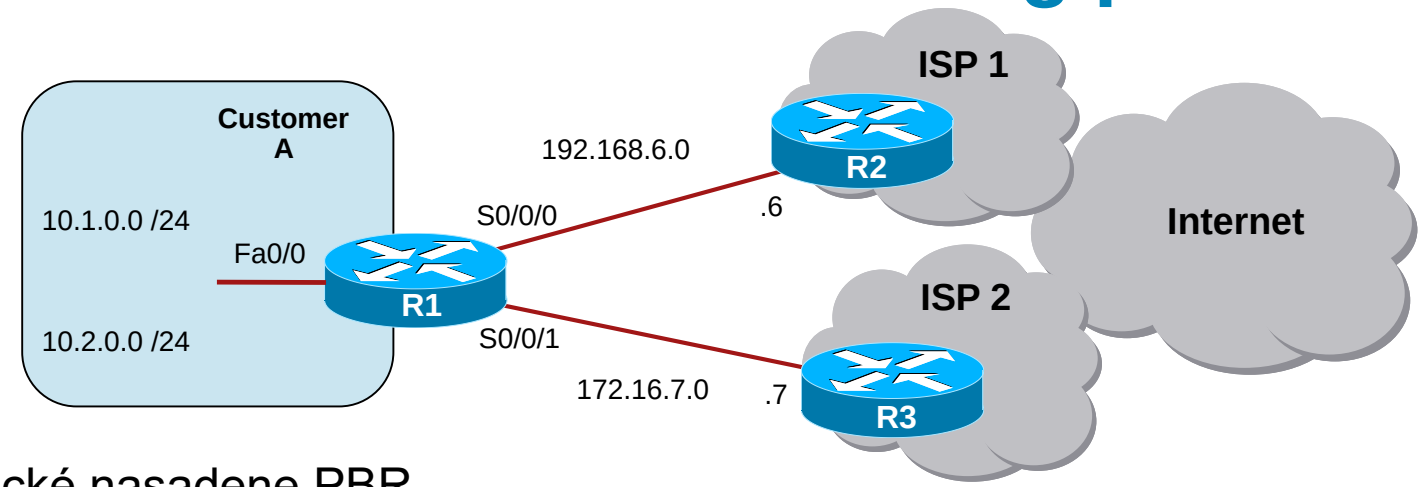
```
Router(config-if)# ip route-cache policy
```

- Výnimky:
  - Niektoré `set` príkazy nie sú podporované
    - Napr. `set ip default-network, set default interface`

# Overenie PBR

| Command | Description |
|---|---|
| **show ip policy** | Zobrazí route map použitú pre PBR |
| **show route-map** [*map-name*] | Zobrazí konfigurovanú route mapu |
| **debug ip policy** | Zobrazí PBR detaily o tom či pakety odpovedajú porovnávacím kritériám |

# Nasadenie PBR v Multihoming prostredí

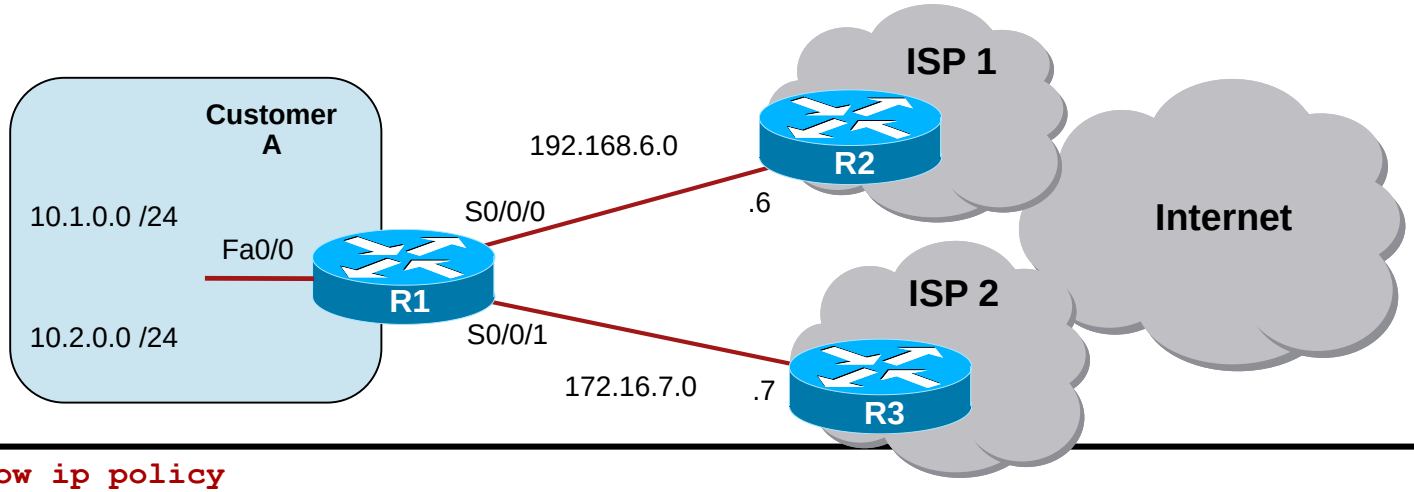



- Typické nasadene PBR.
- R1 smeruje prevádzku na základe odkiaľ paket tečie
  - Prevádzka z 10.1.0.0 pôjde na ISP1 cez 192.168.6.6
  - Prevádzka z 10.2.0.0 pôjde na ISP2 cez 172.16.7.7

```
R1(config)# access-list 1 permit 10.1.0.0 0.0.0.255
R1(config)# access-list 2 permit 10.2.0.0 0.0.0.255
R1(config)# route-map EQUAL-ACCESS permit 10
R1(config-route-map)# match ip address 1
R1(config-route-map)# set ip next-hop 192.168.6.6
R1(config-route-map)# route-map EQUAL-ACCESS permit 20
R1(config-route-map)# match ip address 2
R1(config-route-map)# set ip next-hop 172.16.7.7
R1(config-route-map)# route-map EQUAL-ACCESS permit 30
R1(config-route-map)# set interface null0
R1(config-route-map)# exit
R1(config)# interface FastEthernet 0/0
R1(config-if)# ip address 10.1.1.1 255.255.255.0
R1(config-if)# ip policy route-map EQUAL-ACCESS
R1(config-if)# exit
```

Pakety z iného zdroja sú dropnuté

# Overenie PBR

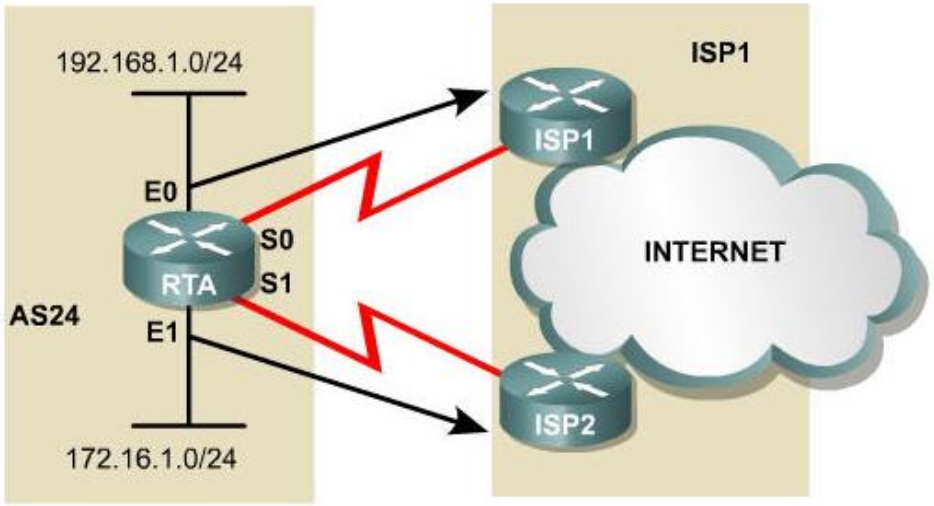



```
R1# show ip policy
Interface          Route map
FastEthernet0/0    EQUAL-ACCESS
```

```
R1# show route-map
route-map EQUAL-ACCESS, permit, sequence 10
  Match clauses:
    ip address (access-lists): 1
  Set clauses:
    ip next-hop 192.168.6.6
Policy routing matches: 3 packets, 168 bytes
route-map EQUAL-ACCESS, permit, sequence 20
  Match clauses:
    ip address (access-lists): 2
  Set clauses:
    ip next-hop 172.16.7.7
route-map EQUAL-ACCESS, permit, sequence 30
Set clauses:
    default interface null0
```

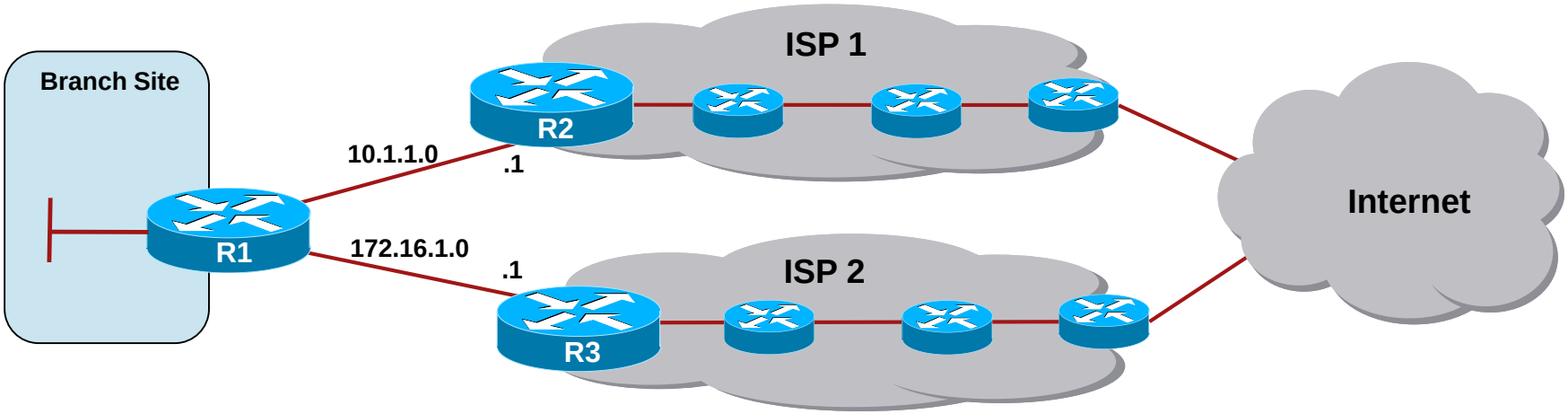
# Príklad 2

```
hostname RTA
!
interface Ethernet 0
  ip address 192.168.1.1 255.255.255.0
  ip policy route-map ISP1

interface Ethernet 1
  ip address 172.16.1.1 255.255.255.0
  ip policy route-map ISP2

route-map ISP1 permit 10
  match ip address 1
  set interface Serial0

route-map ISP2 permit 10
  match ip address 2
  set interface Serial1

access-list 1 permit 192.168.1.0 0.0.0.255
access-list 2 permit 172.16.1.0 0.0.0.255
```



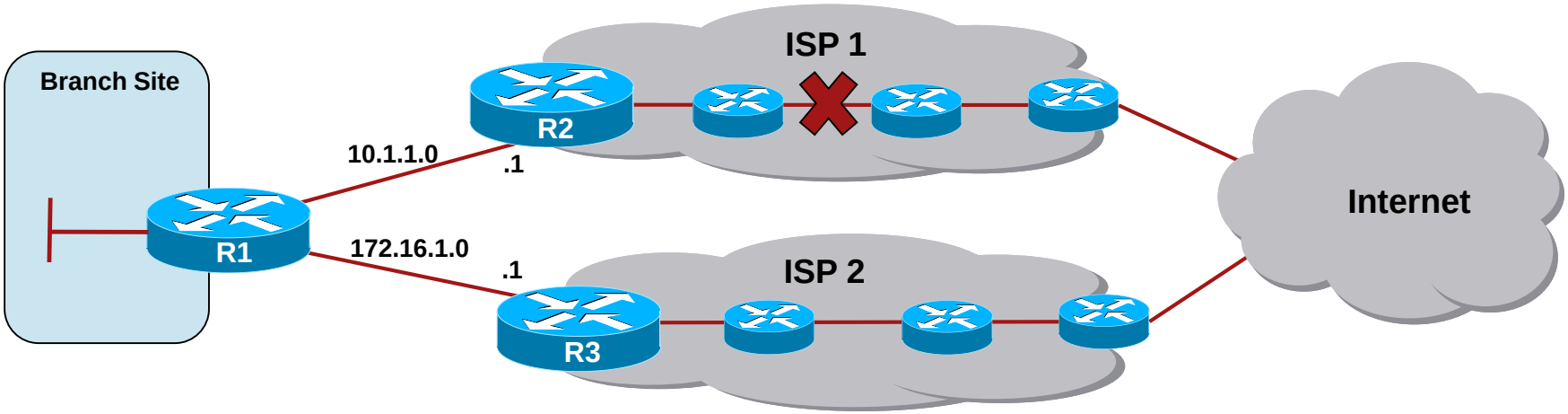
# IP Service Level Agreement (IP SLA)

# Multihomed pripojenie



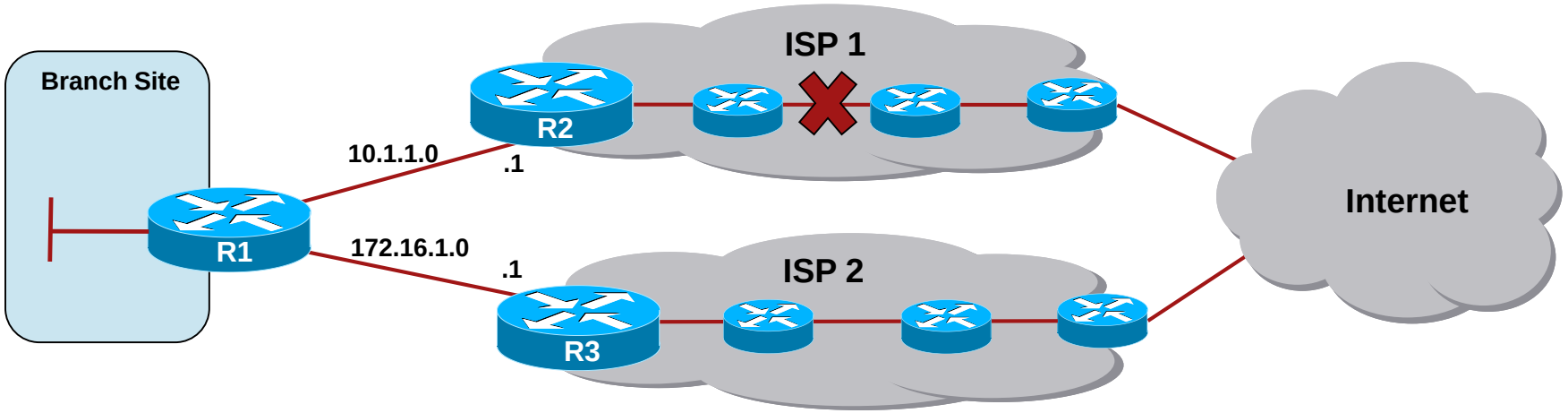

- Predpokladajme, že R1 má dual-homed pripojenie k internetu cez dvoch providerov
- Na R1 stačia dve statické default routes
  - Router bude využívať obe pre load balancing
  - Ak jedna z priamo pripojených liniek vypadne, R1 bude používať zostávajúcu

# Multihomed pripojenie




- Čo sa však stane, ak problém nastane vo vnútri ISP1?
  - Z pohľadu routera R1 je linka do ISP1 stále OK a bude ju trvale používať
  - Dáta odoslané do internetu cez ISP1 sa však stratia
- Ako túto situáciu riešiť?
  - Jedným z riešení je zaviesť dynamický smerovací protokol medzi ISP a zákazníka
  - Závisí na dohode s providerom, mnohokrát provider nie je nadšený, že musí spustiť smerovací protokol voči zákazníkovi

# Multihomed pripojenie




- Iným riešením je na R1 použiť statické cesty alebo PBR, ale podmieniť ich platnosť dodatočným aktívnym testom
  - Napríklad periodický ping na DNS alebo mail server u providera
  - Ak ciele testované na dostupnosť nebudú odpovedať, nimi podmienená statická cesta sa odstráni zo smerovacej tabuľky
- Tieto aktívne testy dosiahnuteľnosti, prípadne i ďalších parametrov sa nazývajú IP Service Level Agreements (SLA)
  - Ak test IP SLA zlyhá, router odstráni zo smerovacej tabuľky položky podmienené týmto testom

# Riadenie smerovania pomocou IOS IP SLAs

- Cisco IOS IP Service Level Agreements (SLAs) slúžia na aktívny monitoring činnosti siete
  - Cisco IP SLA je vlastnosť IOS, ktorá umožňuje vykonávať merania
    - Vykonávané zasielaním umelej prevádzky na hosta alebo smerovač, ktorí sú nakonfigurovaný naň odpovedať
- Cisco IOS IP SLA testy prenášajú sieťou simulované dáta a merajú parametre ich prenosu
  - Je možné stanoviť, aké hodnoty meraných parametrov musia byť splnené, aby bol test považovaný za úspešný
    - IP SLA podporuje veľké množstvo testov
      - Protokoly
        - UDP, TCP, ICMP, HTTP, DNS, DHCP, FTP,…
      - Testovane konektivity (ICMP or UDP)
      - Testovanie chvenia (Jitter)

# Cisco IOS IP SLAs

- Medzi merané parametre patria:
  - Dostupnosť sieťovej služby
  - Čas odpovede (response time)
  - Jednosmerné oneskorenie (One-way latency)
  - Jitter (kolísanie oneskorenia)
  - Stratovosť paketov
  - Hodnotenie kvality hlasu
  - Aplikačný výkon

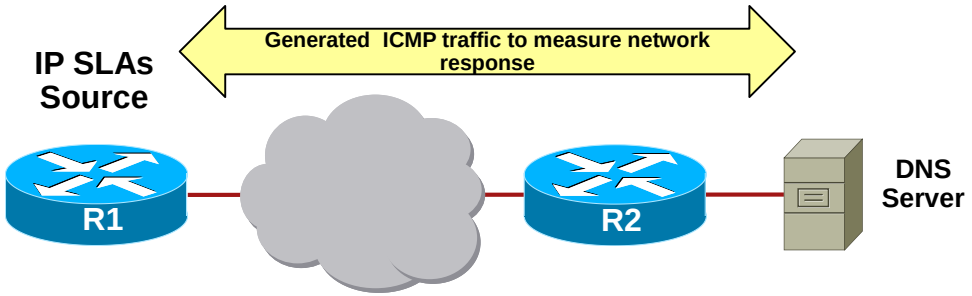
| | *DATA TRAFFIC | *VoIP | *SERVICE LEVEL AGREEMENT | *AVAILABILITY | **STREAMING VIDEO |
|---|---|---|---|---|---|
| REQUIREMENT | • Minimize Delay, Packet Loss<br>• Verify QoS | • Minimize Delay, Packet Loss, Jitter | • Measure Delay, Packet Loss, Jitter<br>• One-way | Connectivity testing | • Minimize Delay, Packet Loss |
| IP SLA MEASURMENT | • Jitter<br>• Packet loss<br>• Latency<br>• per QoS | • Jitter<br>• Packet loss<br>• Latency<br>• MOS Voice Quality Score | • Jitter<br>• Packet loss<br>• Latency<br>• One-way<br>• Enhanced accuracy<br>• NTP | • Connectivity tests to IP devices | • Jitter<br>• Packet loss<br>• Latency |

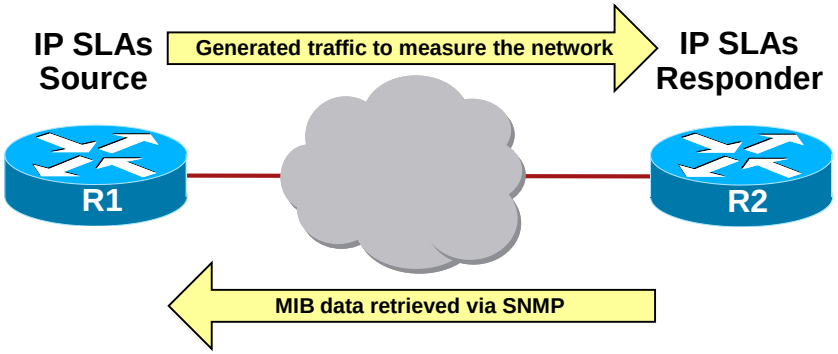
# IP SLA zdroj a respondent

- **IP SLA zdroj (source)**
  - Posiela testovaciu prevádzku na stanovený cieľ
    - Všetky testy sú konfigurované na SLA zdroji (CLI or GUI)
    - SLA zdroj využíva samostatný riadiaci protokol pre komunikáciu s responderom ešte pred začiatkom testu
    - Najmä pre časové charakteristiky je nutné, aby zdroj a respondent boli časovo synchronizovaní (NTP)
- **IP SLA respondent (responder)**
  - je súčasťou IOSu,
  - je komponent na cieli testovacej prevádzky, ktorý slúži na koordináciu prebiehajúceho testu s IP SLA zdrojom
- **IP SLA operácia (operation)**
  - je meranie, ktorého súčasťou je protokol, frekvencia a prahové hodnoty parametrov

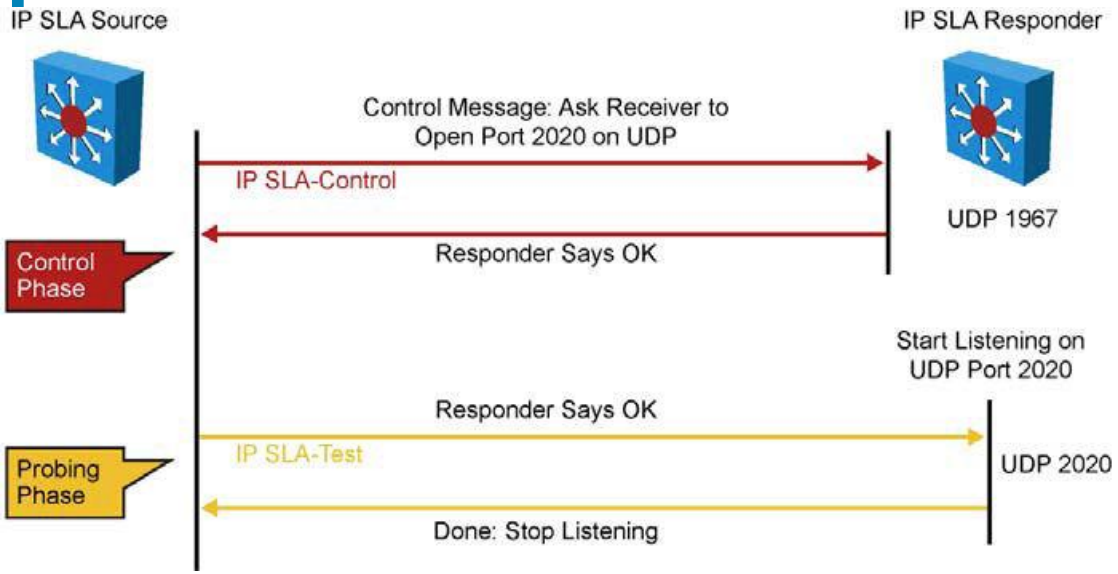
# IP SLAs operácie

- IP SLA testy je možné realizovať:
  - IP SLA voči zariadeniu, na ktorom nebeží SLA respondent (web server alebo IP stanica)
    - Obvykle sú to testy bežného aplikačného protokolu alebo ping
  - IP SLA voči zariadeniu, na ktorom beží SLA respondent (napr. Cisco router)
    - Je možné realizovať dodatočné testy, prípadne získavať presnejšie výsledky



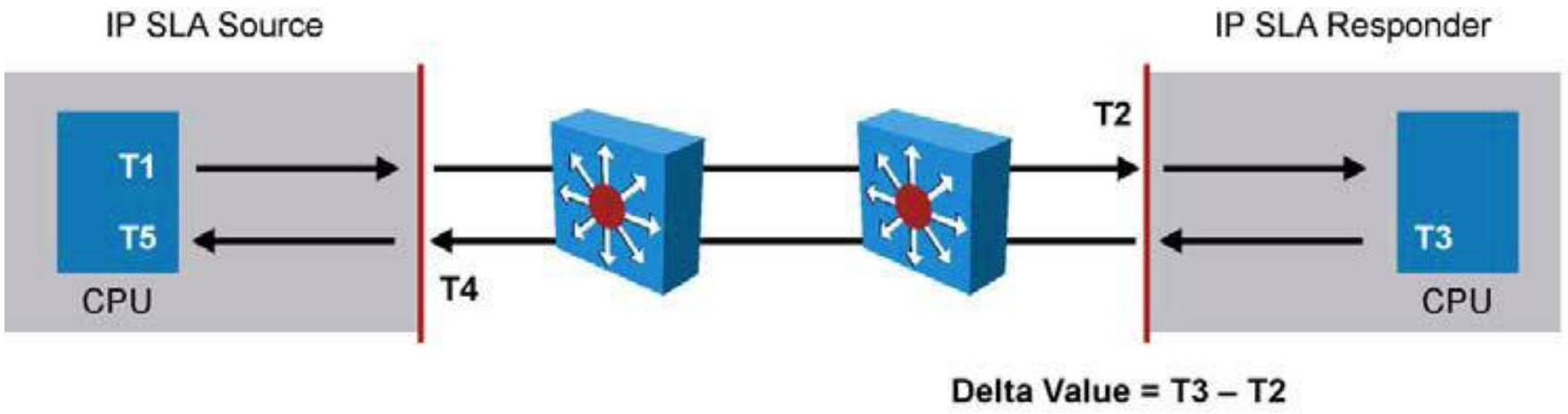



# IP SLA Operácie




- Na IP SLA source sa zadefinuje pre každú IP SLA operáciu
  - Cieľové zariadenie (probe), protokol a UDP or TCP port číslo
- Pred začatím testu prebehne kontrolná fáza
  - IP SLA zdroj následne pred poslaním test paketu použije na komunikáciu s responderom riadiaci protokol (port 1967), ak všetko v poriadku Responder odpovie OK
    - Info o trvaní testu, protokole a porte
    - Riadiace správy môžu byť overované MD5 hashom
- Test začne posielaním sady testovacích paketov medzi IP SLA zdrojom a responderom za periódu času
- Keď test skončí, výsledok je uložený  na zdroji
  - Napr. v SNMP IP SLA MIB

# IP SLA Responder Timestamps




- IP SLA responder timestamps je využité pri kalkulácii round-trip time (RTT)
- IP SLA source posiela paket v čase T1.
- IP SLA responder zahrnie v odpovedi čas príjmu (T2) a čas odoslania (T3).
- Je vhodné mať zdroj aj cieľ synchronizovaný cez NTP

# Konfigurácia IP SLA s object tracking

1. Definovať aspoň jednu SLA operáciu (test, tzv. probe)
2. Definovať dobu trvania operácie

3. Definovať aspoň jeden tzv. tracking object, ktorý bude reprezentovať úspech alebo neúspech SLA operácie
4. Definovať akciu asociovanú s tracking object-om

- Pozor:
  - Počnúc verziou IOSu 12.4(4)T, 12.2(33)SB a 12.2(33)SXI sa príkaz **ip sla monitor** nahrádza príkazom **ip sla**

# Vytvorenie IP SLA operácie

- Vytvorenie IP SLA operácie

```
Router(config)#
ip sla operation-number
```

- Parameter *operation-number* je ID operácie (ľubovoľné)

```
R1(config)# ip sla 1
R1(config-ip-sla)# ?
IP SLAs entry configuration commands:
  dhcp          DHCP Operation
  dns           DNS Query Operation
  exit          Exit Operation Configuration
  icmp-echo     ICMP Echo Operation
  icmp-jitter   ICMP Jitter Operation
  ! Skrátené kvôli stručnosti
R1(config-ip-sla)#
```

# Definovanie IP SLAs ICMP Echo Operácie

- Definovanie ping operácie voči non-responder cieľu

```
Router(config-ip-sla)#
```

```
icmp-echo {destination-ip-address | destination-hostname} [source-
  ip {ip-address | hostname} | source-interface interface-name]
```

| Parameter | Description |
|---|---|
| *destination-ip-address* \| *destination-hostname* | Cieľová IPv4/IPv6 adresa |
| **source-ip** *{ip-address \| hostname}* | (Nepovinné) Stanovuje zdrojovú IPv4/IPv6 adresu |
| **source-interface** *interface-name* | (Nepovinné) Stanovuje rozhranie, z ktorého sa požičia zdrojová IPv4/IPv6 adresa |

**Pozor**:

- Počnúc verziou IOSu 12.4(4)T, 12.2(33)SB a 12.2(33)SXI sa príkaz **`type echo protocol ipIcmpEcho`** nahrádza príkazom **`icmp-echo`**

# icmp-echo – nastavenie detailov

```
R1(config-ip-sla)# icmp-echo 209.165.201.30
R1(config-ip-sla-echo)# ?

IP SLAs echo Configuration Commands:
  default              Set a command to its defaults
  exit                 Exit operation configuration
  frequency            Frequency of an operation
  history              History and Distribution Data
  no                   Negate a command or set its defaults
  owner                Owner of Entry
  request-data-size    Request data size
  tag                  User defined tag
  threshold            Operation threshold in milliseconds
  timeout              Timeout of an operation
  tos                  Type Of Service
  verify-data          Verify data
  vrf                  Configure IP SLAs for a VPN Routing/Forwarding in-stance

R1(config-ip-sla-echo)#
```

- Existuje množstvo parametrov, avšak pre nás sú teraz podstatné len parametre **`frequency`** a **`timeout`**

# icmp-echo – nastavenie detailov

```
Router(config-ip-sla-echo)#
```

```
frequency seconds
```

- Stanovuje, ako často sa operácia bude opakovať
  - Parameter *seconds* udáva počet sekúnd medzi dvomi behmi tejto operácie. Štandardná hodnota je 60 sekúnd.

```
Router(config-ip-sla-echo)#
```

```
timeout milliseconds
```

- Stanovuje čas, do ktorého SLA operácia očakáva odpoveď na odoslanú žiadosť

# Naplánovanie SLA operácie – doba trvania

- IP SLA operáciu je potrebné naplánovať

```
Router(config)#
ip sla schedule operation-number [life {forever | seconds}]
  [start-time {hh:mm[:ss] [month day | day month] | pending |
  now | after hh:mm:ss}] [ageout seconds] [recurring]]
```

**Pozor**:

- Počnúc verziou IOSu 12.4(4)T, 12.2(33)SB a 12.2(33)SXI sa príkaz **`ip sla monitor schedule`** nahrádza príkazom **`ip sla schedule`**

# Voľby príkazy `ip sla schedule`

| Parameter | Description |
|---|---|
| *operation-number* | Number of the IP SLAs operation to schedule. |
| **life forever** | (Optional) Schedules the operation to run indefinitely. |
| **life** *seconds* | (Optional) Number of seconds the operation actively collects information.<br>The default is 3600 seconds (one hour). |
| **start-time** | (Optional) Time when the operation starts. |
| *hh:mm*[*:ss*] | Specifies an absolute start time using hour, minute, and (optionally) second.<br>Use the 24-hour clock notation. |
| *month* | (Optional) Name of the month to start the operation in.<br>If month is not specified, the current month is used. |
| *day* | (Optional) Number of the day (in the range 1 to 31) to start the operation on.<br>If a day is not specified, the current day is used. |
| **pending** | (Optional) No information is collected. This is the default value. |
| **now** | (Optional) Indicates that the operation should start immediately. |
| **after** *hh:mm:ss* | (Optional) Indicates that the operation should start this amount of time after this command was entered. |
| **ageout** *seconds* | (Optional) Number of seconds to keep the operation in memory when it is not actively collecting information (default is 0 seconds which means it never ages out). |
| **recurring** | (Optional) Indicates that the operation will start automatically at the specified time and for the specified duration every day. |

# Vytvorenie tracking object-u

- Vytvoriť tracking object, ktorý bude vyhodnocovať výsledok IP SLA operácie

```
Router(config)#
track OBJECT-NUMBER ip sla OPERATION-NUMBER {state |
 reachability}
```

| Parameter | Description |
|---|---|
| *object-number* | Číslo tracking object-u od 1 do 500 (ľubovoľné) |
| *operation-number* | Číslo SLA operácie, ktorej stav bude tento tracking object uchovávať. |
| **state** | Uchováva návratový kód (OK, OverThreshold, ...) |
| **reachability** | Uchováva všeobecnú úspešnosť |

**Pozor**:

- Počnúc verziou IOSu 12.4(20)T, 12.2(33)SXI1 a 12.2(33)SRE je príkaz **`track rtr`** nahradený príkazom **`track ip sla`**

# Konfigurácia oneskorenia reakcie na Tracking Delay

- Špecifikuje časové oneskorenie ktoré musí uplynúť kým sa zareaguje na zmenu trackovaného objektu

```
Router(config-track)#
```

**delay {up** *seconds* **[down** *seconds*] **| [up** *seconds*] **down** *seconds*}

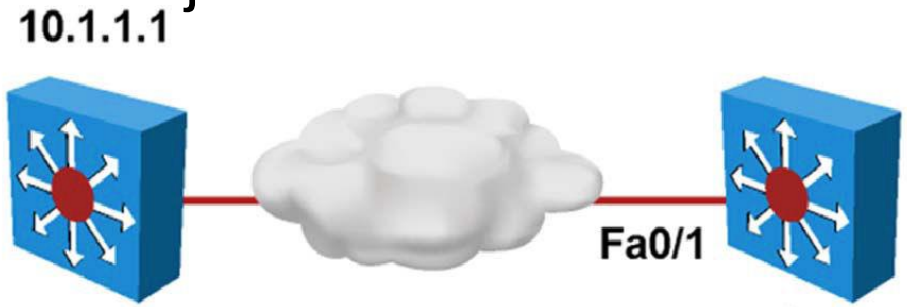
| Parameter | Description |
|---|---|
| **up** | Time to delay the notification of an up event. |
| **down** | Time to delay the notification of a down event. |
| *seconds* | Delay value, in seconds.<br>The range is from 0 to 180 with the default being 0. |

# Overenie IP SLA

| Command | Description |
| --- | --- |
| **show ip sla configuration** [*operation*] | Display configuration values including all defaults for all Cisco IOS IP SLAs operations, or for a specified operation.<br>The *operation* parameter is the number of the IP SLAs operation for which the details will be displayed. |
| **show ip sla statistics** [*operation-number* \| **details**] | Display the current operational status and statistics of all Cisco IOS IP SLAs operations, or of a specified operation. |

# Príklad IP SLA (monitor – po starom)

- IP SLA test vykoná odoslanie ipIcmpEcho správy každých 10sekúnd na cieľovú IP adresu 10.1.1.1 cez lokálne rozhranie IP SLA zdroja Fa0/1




```
SwitchB(config)# ip sla monitor 11
SwitchB(config-sla)# type echo protocol ipIcmpEcho 10.1.1.1 source-
int fa0/1
SwitchB(config-sla)# frequency 10
SwitchB(config-sla)# exit
SwitchB(config)# ip sla monitor schedule 11 life forever start-time
now
SwitchB(config)# track 1 ip sla 11 reachability
```

# Overenie IP SLA konfigurácie

- Výpis informácii o IP SLA testovacej konfigurácii

```
Switch# show ip sla configuration
IP SLAs, Infrastructure Engine-II
Entry number: 1
Owner:
Tag:
Type of operation to perform: echo
Target address/Source address: 10.1.3.10/10.1.253.1
Type Of Service parameter: 0x0
Request size (ARR data portion): 28
Operation timeout (milliseconds): 5000
Verify data: No
Vrf Name:
Schedule:
     Operation frequency (seconds): 5
     Next Scheduled Start Time: Start Time already passed
     Group Scheduled : FALSE
     Randomly Scheduled : FALSE
     Life (seconds): Forever
     Entry Ageout (seconds): never
     Recurring (Starting Everyday): FALSE
     Status of entry (SNMP RowStatus): Active
Threshold (milliseconds): 5000
<output omitted>
```

# Overenie IP SLA štatistík

- Po spustení testu sú zozbierané výsledky
  - Test môže skončiť úspechom alebo neúspechom

```
Switch# show ip sla statistics
Round Trip Time (RTT) for Index 1
Latest RTT: NoConnection/Busy/Timeout
Latest operation start time: 11:11:22.533 eastern Thu Jul 9 2010
Latest operation return code: Timeout
Over thresholds occurred: FALSE
Number of successes: 177
Number of failures: 6
Operation time to live: Forever
Operational state of entry: Active
Last time this entry was reset: Never
```

# Reštart IP SLA štatistík

- Resetovanie nazbieraných výsledkov

```
Router(config)# ip sla restart IP_SLA_OPER_NUMB
```

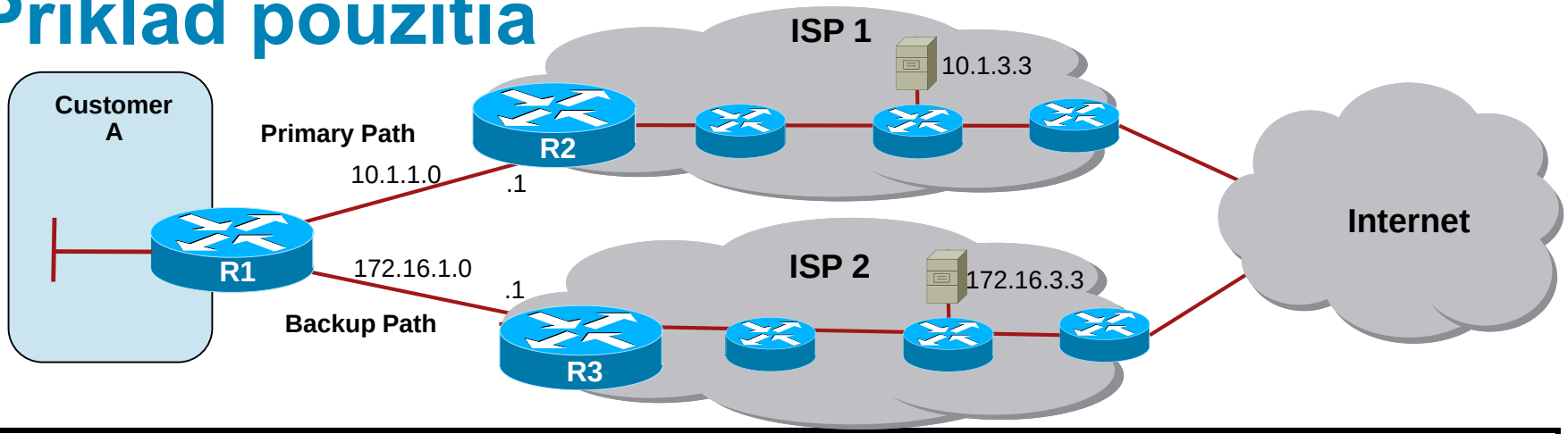
# Konfigurácia statických ciest a IP SLAs

- Konfigurácia statickej cesty aby reagovala na IP SLA tracking.

```
Router(config)#
ip route PREFIX MASK ADDRESS INTERFACE dhcp DISTANCE name
  NEXT-HOP-NAME permanent track NUMBER tag TAG
```

| Parameter | Description |
|---|---|
| *prefix mask* | The IP network and subnet mask for the remote network to be entered into the IP routing table. |
| *address* | The IP address of the next hop that can be used to reach the destination network. |
| *interface* | The local router outbound interface to be used to reach the destination network. |
| **dhcp** | (Optional) Enables a DHCP server to assign a static route to a default gateway. |
| *distance* | (Optional) The administrative distance to be assigned to this route. |
| **name** *next-hop-name* | (Optional) Applies a name to the specified route. |
| **permanent** | (Optional) Specifies that the route will not be removed from the routing table even if the interface associated with the route goes down. |
| **track** ***number*** | (Optional) Associates a track object with this route.<br>Valid values for the number argument range from 1 to 500. |
| **tag** *tag* | (Optional) A value that can be used as a match value in route maps. |

# Príklad použitia




```
R1(config)# ip sla 11
R1(config-ip-sla)# icmp-echo 10.1.3.3
R1(config-ip-sla-echo)# frequency 10
R1(config-ip-sla-echo)# exit ! 2x
R1(config)# ip sla 22
R1(config-ip-sla)# icmp-echo 172.16.3.3
R1(config-ip-sla-echo)# frequency 10
R1(config-ip-sla-echo)# exit ! 2x
R1(config)# track 1 ip sla 11 reachability
R1(config-track)# delay down 10 up 1
R1(config-track)# exit
R1(config)# track 2 ip sla 22 reachability
R1(config-track)# delay down 10 up 1
R1(config-track)# exit
R1(config)# ip sla schedule 11 life forever start-time now
R1(config)# ip sla schedule 22 life forever start-time now
R1(config)# ip route 0.0.0.0 0.0.0.0 10.1.1.1 2 track 1
R1(config)# ip route 0.0.0.0 0.0.0.0 172.16.1.1 3 track 2
```


CISCO