INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Tato publikace je spolufinancována Evropským sociálním fondem a státním rozpočtem České republiky

Registrační číslo projektu: CZ.1.07/1.1.12/01.0004

Studijní materiál

CCNA Exploration – Směrování, koncepce a protokoly

(Semestr 2)

VOŠ a SPŠE Plzeň

2010

Tato publikace je předmětem průběžné aktualizace bez předchozího upozornění.

Verze: 3.01

Tato publikace je spolufinancována Evropským sociálním fondem a státním rozpočtem České republiky v rámci projektu „Výuka počítačových sítí v mezinárodním programu Síťová akademie Cisco na střední průmyslové škole elektrotechnické“.

Registrační číslo projektu: CZ.1.07/1.1.12/01.0004.

Vydala VOŠ a SPŠE Plzeň, Koterovská 85, 326 00 Plzeň v roce 2010.

Kolektiv autorů (řešitelé projektu):

- ***Koncepce a text:*** ***Ing. Miroslav Páv***
- ***Vektorová grafika:*** ***Mgr. Jan Syřínek***
- ***Konzultace angličtiny:*** ***Mgr. Jana Hošková***

- *Tato publikace je určena jako doplňkový studijní materiál ke kurzu CCNA Exploration – Routing Protocols and Concepts. Nejedná se o doslovný překlad celého kurikula ale o nově vytvořené vlastní výklady podporující představivost žáků a komentované upravené překlady vybraných částí jednotlivých kapitol anglického kurikula určené pro usnadnění výuky i studia originálního kurzu v prostředí české odborné střední školy.*
- *Obsah kurzu je integrován v rámci ŠVP naší školy.*
- *Tento dokument je zpracovaný v kancelářském balíku OpenOffice.org a jeho vektorová grafika v grafickém editoru Dia.*
- *Protože se jedná o materiál podléhající v rámci projektu průběžné aktualizaci, používejte vždy poslední dostupnou verzi.*
- *Pro zachování vazby na původní učební materiály (kurikula) jsou u českých termínů uváděny i jejich anglické originály.*
- *Aktuální verze originálních materiálů v angličtině (pro registrované účastníky programu NetAcad): http://www.cisco.com/web/learning/netacad/index.html (http://cisco.netacad.net, www.cisco.com/go/netacad , http://www.cisco.com/edu)*

NEPRODEJNÉ

Prosím, dodržujte licenci pro použití této publikace: ***Určeno výhradně pro LCNA a RCNA programu Cisco NetAcad (CNA, CNAP) v ČR i v SR s licencí Creative Commons*** *(Uveďte autora-Neužívejte dílo komerčně-Nezasahujte do díla 3.0 Česko):*

Dílo smíte šířit za těchto podmínek: Uveďte autora, neužívejte dílo komerčně, nezasahujte do díla (viz plný text licence).

To znamená, že ve své vlastní síťové akademii můžete tuto publikaci šířit volně a nekomerčně tak jak je.

Pokud tuto publikaci používáte při své výuce, prosím Vás o informaci o této skutečnosti. Věcné a konstruktivní připomínky, náměty i popřípadě nalezené chyby mi zasílejte, prosím, na adresu: pav@spse.pilsedu.cz, věc: ***CCNA_Exploration_2.PDF (verze: 3.01)****.*

Vaší spolupráce si vážím a děkuji Vám za ni!

Za kolektiv autorů

Miroslav Páv

CCNA Exploration – Směrování, koncepce a protokoly

Upozornění: Tento materiál nenahrazuje samotné kurikulum ani Vaše vlastní školní poznámky.

- *Pro procvičování jednotlivých příkazů a celých konfigurací sítí používejte **simulátor Packet Tracer** (v poslední dostupné verzi).*
- *Pro analýzu síťového provozu na stanici používejte **analyzátor síťových protokolů WireShark** v režimu **s právy lokálního administrátora na stanici**.*
- *Samostatně si odpovídejte na kontrolní otázky v souhrnu a kvízu pro každou kapitolu v kurikulu.*
- *Postupujte podle pravidla: **pochopit – naučit se – procvičit – otestovat znalosti i dovednosti**.*
- *Při nastavování na reálných zařízeních v učebně i pro Packet Tracer používejte stále stejná hesla:*
  - *pro privilegovaný režim enable: **cisco***
  - *pro linku vty - telnet a také pro linku konzole: **class***
- *Protože se jedná o pracovní verzi (stále se upravuje), používejte vždy poslední dostupnou verzi dle data exportu do PDF (a zbytečně netiskněte).*
- *Originální materiály v angličtině: http://www.cisco.com/web/learning/netacad/index.html (http://cisco.netacad.net, http://www.cisco.com/edu).*

Předpokládané znalosti

Kurz navazuje na *CCNA Exploration - Network Fundamentals (CCNA1 Exploration)* (Informace o e-learningové iniciativě Cisco CNAP a obsah celého kurzu CCNA viz soubor **CCNA_Exploration_1.PDF**.).

Směrování, koncepce a protokoly

Základní dovednosti a kompetence absolventa kurzu ***CCNA Exploration - Routing Protocols and Concepts*:**

- Konfiguruje a ověřuje činnost rozhraní směrovače
- Demonstruje obsáhlé dovednosti nastavení RIPv1
- Navrhuje a implementuje beztřídní IP adresní schéma sítě
- Aplikuje základní konfigurační příkazy RIPv2 a vyhodnocuje směrovací aktualizace RIPv2 u beztřídního směrování
- Používá pokročilých konfiguračních příkazů na směrovačích s protokolem EIGRP
- Identifikuje charakteristiky směrovacích protokolů s vektorem vzdálenosti.
- Implementuje základní nastavení směrovacího protokolu OSPF.

Obsah kursu **CCNA Exploration - Routing Protocols and Concepts:**

10 Směrovací protokoly typu stav linky (Link-State)
- 10.1 principy
- 10.2 implementace protokolů tohoto typu

11 Protokol OSPF
- 11.1 propagace směrovacích informací
- 11.2 základní konfigurace
- 11.3 výpočet metriky
- 11.4 sítě s vícenásobnými přístupy (*multi-acces network*), více bran do sítě

Úvod

Zopakujte si úvodní kapitolu v kurikulu pro první semestr.

Příkazy pro nastavení směrovačů jsou uváděny kromě povinných (mandatorních) též jako nepovinné (volitelné), tyto nepovinné příkazy/parametry sice nejsou přímo obsahem kurikula, ale je poměrně vhodné je alespoň rámcově znát.

Pro každou kapitolu si v rámci originálního kurikula vždy zpracujte pro každý směrovací protokol následující tři aktivity *Configuration Labs* – konfigurační laboratorní cvičení - v simulátoru sítě Packet Tracer:

- *Basic Configuration* – základní konfigurace s detailním návodem,
- *Challenge Configuration* – pokročilejší konfigurace, bez detailního návodu,
- *Troubleshooting* – hledání neznámých chyb a jejich odstraňování v demonstrační konfiguraci (toto cvičení znalé studenty baví nejvíce).

Kapitola 1 – Úvod do směrování a přeposílání paketů na směrovači

V této kapitole se naučíme:

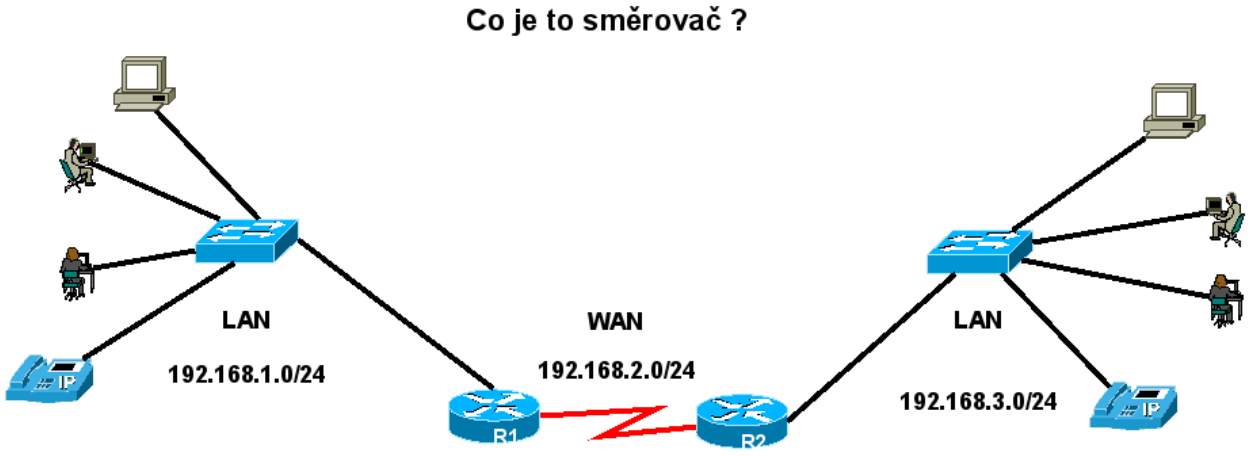
- Směrovač je počítač s operačním systémem (OS) a HW, který je speciálně navržený pro směrování.
- Demonstrovat schopnost konfigurování zařízení a nastavení adres rozhraní.
- Popsat strukturu směrovací tabulky.
- Popsat jak směrovač určuje cestu a přepíná pakety.

Směrovač

Směrovač (router) je centrem datové sítě. Vyjádřeno jednoduše: **směrovač propojuje jednu síť s jinou sítí**. Proto je směrovač zodpovědný za doručení dat mezi různými sítěmi včasným způsobem. Efektivita komunikace mezi vzájemně propojenými sítěmi je ve velké míře závislá na schopnosti směrovače přeposílat pakety co nejvíce efektivním způsobem. Aby vyhověly těmto požadavkům, používají se směrovače také k:

- zajištění 24x7 dostupnosti (24 hodin denně a 7 dnů v týdnu). V případě, že jedna cesta selhala, směrovač použije jinou.
- Poskytování integrovaných služeb pro data, video a hlasové služby prostřednictvím drátových či bezdrátových sítí. Směrovače používají při kvalitě služeb (Quality of service (QoS)) nastavení priorit IP paketů tak, aby se zaručil v reálném čase provoz, který nesmí přerušen nebo zpožděn (jako jsou hlasová služba, video nebo kritická data).
- Zmírnění vlivu červů, virů a jiných útoků v síti pomocí povolení nebo zakázání přeposlání paketů.

Primární odpovědností routeru ale stále zůstává **přeposílání paketů z jedné sítě do druhé**.



Struktura směrovače

Směrovač je počítač, právě tak jako jakýkoliv jiný počítač včetně PC. Úplně první směrovač, použitý pro síť ARPANET (*Advanced Research Projects Agency Network*), byl *Interface Message Processor* (IMP). IMP byl minipočítač Honeywell 316 a byl uveden do provozu 30.8.1969.

Poznámka: Síť ARPANET byla vytvořena agenturou ministerstva obrany USA ARPA (Advanced Research Projects Agency = Agentura pro pokročilé výzkumné projekty). ARPANET byla první funkční sítí na bázi přepínání paketů na světě a byla předchůdcem dnešního Internetu.

Směrovače mají mnoho stejných HW a SW komponent, jaké jsou v jiných počítačích včetně:

- CPU
- RAM
- ROM
- (síťový) operační systém (OS)
- síťové rozhraní – každé leží v jiné (pod)síti (LAN, WAN) – LAN je obvykle Ethernet, používá různá média, respektive sloty na zasunutí modulů s rozhraními a WAN technologie[1] zahrnující sériové linky, T1 připojení s protokolem PPP (*Point to Point Protocol*), Frame Relay a nebo ATM (*Asynchronous Transfer Mode*).
- konzolový port – pro počáteční konfiguraci – není to síťové rozhraní – síť nemusí být v okamžiku připojení nakonfigurována
- pomocný port (AUX) – pro vzdálené připojení modemem
- jsou obvykle bezdiskové a místo pevných disků používají paměť typu flash.

Směrovače vybírají nejlepší cestu

Primární odpovědností směrovače je směrovat (přeposlat) pakety mířící (směřující, mající cílovou adresu) do lokální nebo vzdálené sítě pomocí:

- **určení nejlepší cesty** pro poslání paketu (na L3)[2]
- **posílání (přepínání) paketů** směrem k jejich cíli (na L2) – včetně zapouzdření na linkové vrstvě.

Výběr nejlepší cesty probíhá na základě obsahu směrovací tabulky. **Obsah směrovací tabulky** se vytváří:

- staticky (administrátor ručně)
- dynamicky (dynamický směrovací protokol).

CPU a paměti

Směrovač obvykle nepotřebujete otvírat, pokud zrovna nechcete upgradovat paměť.

Podobně jako PC směrovač obsahuje:

- procesor CPU (*Central Processing Unit* (CPU)
- operační paměť RAM (*Random-Access Memory*)
- paměť typu ROM (*Read-Only Memory*)

Použití:

1 Rozhraní řešené obvykle jako zásuvné výměnné moduly WIC (WAN Interface Card).
2 Směrovač propojuje sítě. Pokud ho nějaké do sítě přidáme (a změníme adresaci), zmenšuje broadcastovou doménu.

- CPU - vykonává příkazy operačního systému, jako je inicializace OS, funkce směrování a funkce přepínání.
- RAM – 128MB rozšířitelná na 384MB, při restartu směrovače ztrácí svůj obsah a ukládá následující komponenty:
  - OS se do RAM zkopíruje během zavádění systému (bootup) (je to rychlejší než pracovat přímo s pamětí Flash, jak tomu bylo ve starých routerech),
  - aktuální běžící konfigurační soubor (*running-config*),
  - směrovací tabulka,
  - ARP cache – mapování IP adres na MAC adresy,
  - vyrovnávací paměť paketů (*packet buffer*) – když je přijat na rozhraní nebo dokud není odeslán z rozhraní
- ROM – permanentní paměť, která obsahuje firmware, jež obvykle není třeba upgradovat:
  - instrukce pro zavádění systému (*bootstrap, loader*) - zavaděč,
  - základní diagnostický SW pro HW směrovače (= *POST = Power-On Self-Test*),
  - odlehčená verze IOS (*scaled-down version*) (= tzv. ROM monitor).
- Flash paměť – permanentní paměť na SIMM nebo PCMCIA kartě – 32, 64, 128 MB – implicitně 32MB, která lze elektricky vymazat a nahrát
  - obrazy (*images*) operačního systému (různě zvolená vybraná funkcionalita OS)
- NVRAM (*Nonvolatile RAM*) – 2-4MB, energeticky nezávislá permanentní paměť, po vypnutí napájení či při restartu neztrácí svůj obsah:
  - startovací konfigurace směrovače (*startup-config*) – při změnách v aktuální konfiguraci je třeba potom aktuální konfiguraci nahrát do startovací konfigurace.

Síťový operační systém IOS

IOS (*Cisco Internetwork Operating System*) – spravuje HW a SW zdroje směrovače (protože může být použit i na L3 přepínači, tak také L3 přepínače) jako alokace paměti, zabezpečení a souborový systém. IOS je víceúlohový (multitasking) OS, který integruje úlohy vztahující se ke směrování, přepínání, propojování sítí a telekomunikaci.

Ačkoliv se IOS může jevit jako stejný na mnoha směrovačích, je zde mnoho různých obrazů (*images*) IOS. Obraz systému obsahuje kompletní IOS pro určitý směrovač. Obrazy jsou závislé na modelu (typu) směrovače a funkcích obsažených v systému. Typicky, čím více funkcí, tím je větší obraz a tím větší je potřeba flash i operační paměť pro systém. Například některé funkce obsahují schopnost spustit IPv6 nebo NAT (*Network Address Translation*).

Jako jiné OS má i IOS svoje vlastní uživatelské rozhraní. Ačkoliv některé směrovače mají grafické uživatelské rozhraní (*GUI, graphical user interface*), je nejběžnějším rozhraním příkazová řádka (*CLI, command line interface*). V tomto kurikulu je použita výhradně příkazová řádka.

Během zavádění systému je startovací konfigurační soubor (startup-config) z NVRAM zkopírován do RAM a uložen jako běžící konfigurace (running-config). Jakékoliv změny vložené administrátorem sítě jsou uloženy do běžící konfigurace a **bezprostředně** uvedeny v činnost v IOS.

Postup zavedení OS

Jsou čtyři hlavní fáze postupu zavedení operačního systému (*bootup process*):

1. Provedení testu POST (*Power-On Self Test*) - automatický test po zapnutí (v ROM) testuje HW směrovače – diagnostika procesoru, RAM, NVRAM.
2. Natažení zaváděcího program (*bootstrap program*, *loader*) - zavaděč je natažen do operační paměti, jednotlivé instrukce provádí procesor z RAM, od této chvíle je funkční konzolové připojení a na monitoru konzole je možné vidět průběžné výpisy stavu. V této chvíli verze bootstrap.
3. Nalezení a zavedení IOS – IOS je typicky uložen v paměti flash, ale může být také uložen na TFTP serveru. Jak se začne natahovat IOS na konzoli se vypisuje znak dvojitý kříž (*hash mark*) (#) jak postupuje dekomprese systému.
4. Nalezení a zavedení souboru startovací konfigurace nebo spuštění režimu nastavování – setup. Startup-config je natažen z NVRAM a obsahuje uloženou předchozí běžící konfiguraci Obsahem jsou konfigurační příkazy a parametry jako:
   - adresy rozhraní,
   - směrovací informace,
   - hesla,
   - všechna ostatní nastavení uložené administrátorem.

- Jestliže v NVRAM není startovací konfigurace, může jí směrovač hledat na TFTP serveru pomocí všesměrového vysílání.
- Pokud se konfigurační soubor nalezne, jsou jeho jednotlivé příkazy vykonány.
- Pokud se konfigurační soubor nenalezne, směrovač uživateli nabídne vstup do interaktivního nastavovacího režimu – *setup mode*. (Lze spustit přímo příkazem #setup. Ale tomu se v tomto kursu vyhneme.)

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

Pokud se do něho (setup mode) náhodou dostanete, ukončíte ho stiskem **Ctrl+C**.

Rozhraní příkazové řádky z konzole (CLI): Před vstupem do něj směrovač nabídne ukončení automatické instalace (to přijmeme):

```
Would you like to terminate autoinstall? [yes]: <Enter>

Press the Enter key to accept the default answer.

Router>
```

POZOR pokud na předchozí dotaz směrovače na ukončení **AutoInstall** odpovíte NO, nebo pokud má směrovač smazanou konfiguraci nebo je úplně nový, směrovač se bude pokoušet získat konfiguraci z TFTP serveru, nastavit Ethernetová rozhraní pomocí protokolu DHCP a nastavit sériová rozhraní pomocí protokolu SLARP (Serial Line Address Resolution Protocol) **a to zabere několik minut**.
PROTO: **Před zapnutím takovéhoto nenastaveného směrovače odpojte všechna síťová rozhraní**.

ROM	---------->	Bootstrap – loader = zavaděč	ROM monitor - nouzový režim
Flash paměť	---------->	Cisco IOS	Nalezne a zavede operační systém
TFTP server	---------->		
ROM (ROM monitor, omezená verze IOS)	---------->		
NVRAM	---------->	Configuration File – konfigurační soubor	Nalezne a zavede konfigurační soubor nebo vstup do interaktivního režimu „Setup“.
TFTP Server	---------->		
Console	---------->		

Ověření zavedeného systému

Příkazem ***#show version*** zjistíte:

1. zavedená verze IOS
2. použitý program bootstrap z ROM
3. umístění obrazu IOS (odkud byl zaveden) a jméno obrazu
4. procesor a velikost paměti RAM směrovače
5. rozhraní směrovače (názvy rozhraní na výměnných modulech ve slotech: FastEthernet0/0, Serial0/1/1, ...)
6. velikost paměti NVRAM
7. velikost paměti Flash
8. hodnotu konfiguračního registru (Nastavená hodnota konfiguračního registru různými způsoby mění chování směrovače například: odkud zavádí OS, chování během jeho zavedení, například přeskočení konfigurace, rychlost konzolového připojení apod.
    - Implicitní tovární nastavení je 0x2102: pokusí se zavést IOS z Flash a konfigurační soubor z paměti NVRAM.
    - Hodnota 0x2142 přeskočí konfiguraci v NVRAM.
    - Má to více použití, například obnova zapomenutého hesla. Viz *Password Recovery Procedure*, později.

```
R3#show version
Cisco IOS Software, 1841 Software (C1841-ADVIPSERVICESK9-M), Version
12.4(15)T1, RELEASE SOFTWARE (fc2)
<vynecháno>
ROM: System Bootstrap, Version 12.3(8r)T8, RELEASE SOFTWARE (fc1)
<vynecháno>
System image file is "flash:c1841-advipservicesk9-mz.124-15.T1.bin"
<vynecháno>
```

```
Cisco 1841 (revision 5.0) with 114688K/16384K bytes of memory.
Processor board ID FTX0947Z18E
M860 processor: part number 0, mask 49
2 FastEthernet/IEEE 802.3 interface(s)
2 Low-speed serial(sync/async) network interface(s)
191K bytes of NVRAM.
63488K bytes of ATA CompactFlash (Read/Write)

Configuration register is 0x2102

R3#
```

Rozhraní směrovače

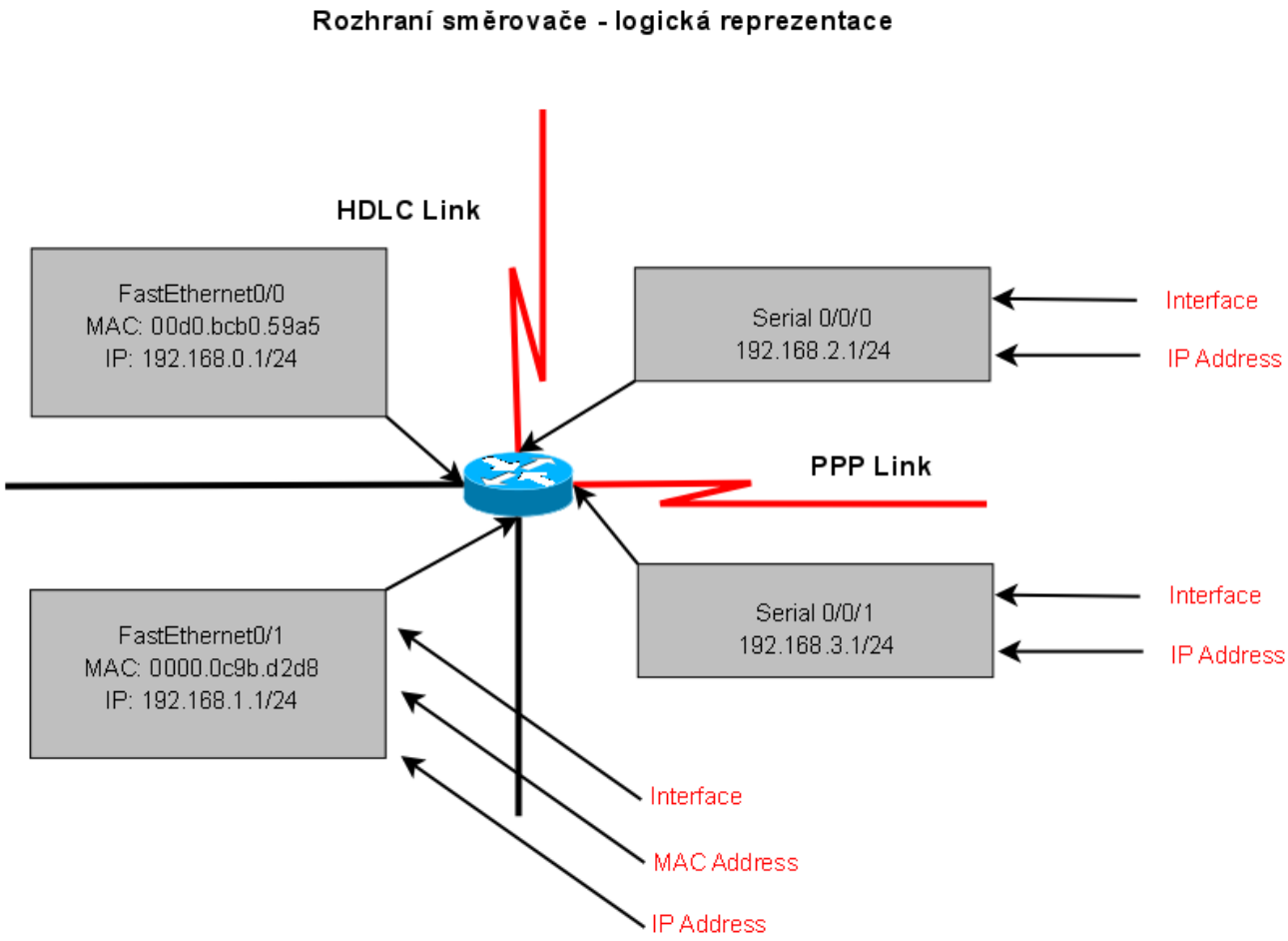
Administrativní porty – fyzické konektory pro správu směrovače. Jsou dvojího druhu:

- **Konzolový (*console*) port** – pro (počáteční) konfiguraci. Protože to není síťové rozhraní není třeba mít konfigurované síťová rozhraní a síťové služby. Není určený pro přeposílání paketů. Připojené PC (přes RS232 a DB9) musí mít nainstalovaný SW pro emulaci terminálu (HyperTerminal, TerraTerm).
- **Pomocný (AUX, *auxiliary*) port** – pro konfiguraci po připojení modemu. Také nesíťový port. V tomto kurzu nebudeme používat.

(Síťová) rozhraní směrovače (interface) – pojem rozhraní směrovače odkazuje na fyzický port na směrovači jehož hlavní funkcí je přijímat a posílat pakety. Směrovač má více rozhraní, které jsou připojeny do **různých** sítí[3]. (Cisco IOS nepovolí na jednom směrovači, aby bylo v jedné síti více rozhraní.) Typicky se rozhraní zapojují do různých typů sítí, což znamená, že potřebují různé druhy konektorů a médií.

- Obvykle směrovač používá pro **připojení do sítí LAN** rozhraní typu FastEthernet (kabeláž UTP (podle druhu propojovaných zařízení přímý nebo překřížený) a konektory RJ-45). Používají fyzickou MAC adresu i IP adresu a protokol ARP pro jejich vzájemné spárování.
- Pro **připojení do sítí WAN** směrovač používá různé typy sériových linek jako T1, DSL, IDSN nebo technologii Frame Relay (*nyní už často ale i Gigabit Ethernet*). Ve WAN se MAC adresy nepoužívají (jde obvykle o dvoubodové připojení), ale některé technologie WAN všesměrovu MAC adresu mají použitou v záhlaví protokolu např. PPP a HDLC. Rozhraní mají vždy IP adresu.

3 Každé rozhraní na jednom směrovači je v jedné jiné síti. Poznámka: U virtuálních sítí VLAN má potom jedno rozhraní několik virtuálních podrozhraní (virtual subinterface). A každé podrozhraní je v jiné virtuální síti. (Bude v CCNA3.)



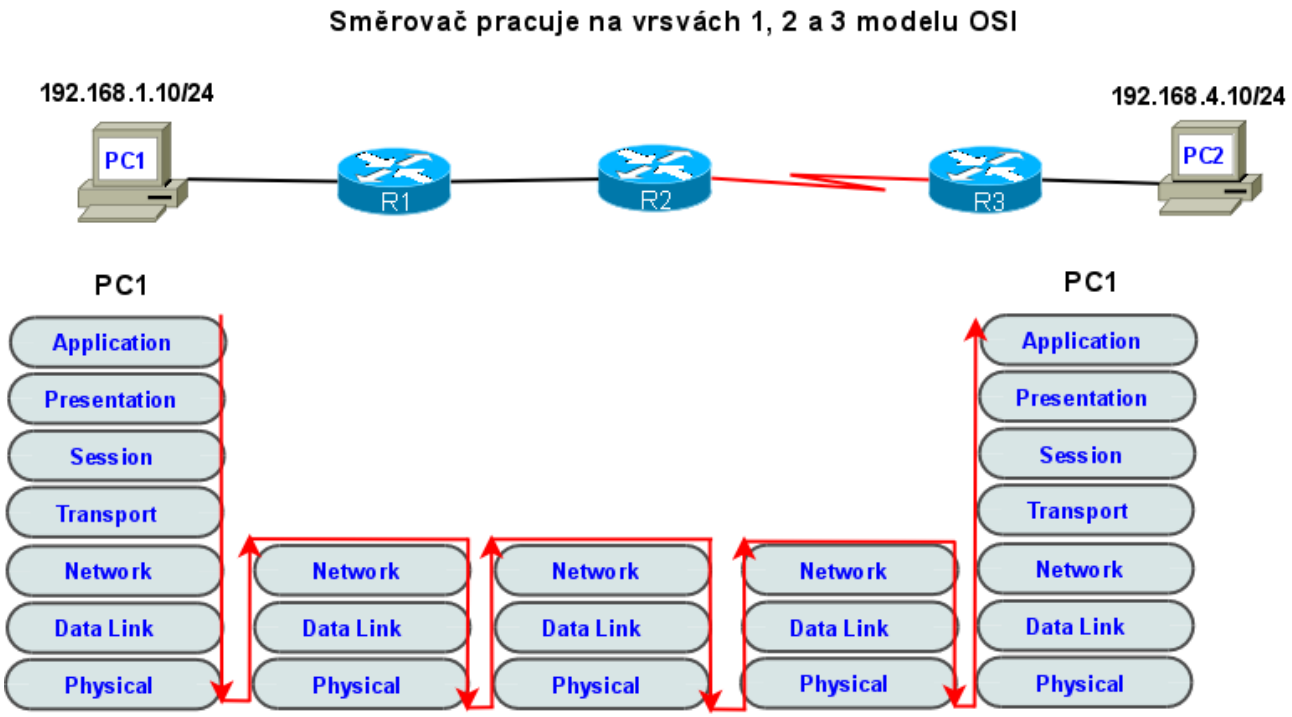
Jako většina síťových zařízení, směrovače používají k indikaci stavu rozhraní elektroluminiscenční diody LED. Konkrétní význam světelného signálu je závislý na konkrétním směrovači. Například nepřerušované světlo znamená stav obsazená linka.

Rozhraní se mohou do slotů směrovače přidávat jako zásuvné moduly například High-Speed WAN Interface Card (HWIC) nebo WAN Interface Card (WIC) s například sériovými konektory Smart Serial nebo většími DB60.

Směrovač na 3. vrstvě OSI modelu.

Směrovač pracuje na vrstvách 1., 2. a 3 modelu OSI.

- Na 3. vrstvě směruje pakety z jedné sítě do druhé (nejlepším směrem k cílové síti).
- Na 2. vrstvě odpouzdřuje a zapouzdřuje pakety do rámců a přepíná.
- Na 1. vrstvě zpracovává, tj. přijímá a vysílá, signály.



Implementace základního adresního schéma

Když navrhujete novou síť nebo mapujete a dokumentujete existující síť, tak minimální dokumentace by měla zahrnovat schéma topologie sítě (topologický diagram) a tabulku IP adres s následujícími informacemi:

- jméno zařízení,
- použité rozhraní,
- IP adresa a maska podsítě,
- pro koncová zařízení jako jsou PC také adresu implicitní brány (*default gateway*).

Viz 1.2.1 obr. 1. (Oba „konce drátu“ leží vždy v jedné IP síti.)

Základní konfigurace směrovače

Základní úkoly (kroky) při konfigurace směrovače:

- pojmenování směrovače (*hostname*)
- nastavení hesel (*password*)
- konfigurace rozhraní (*interface*)
- konfigurace denní uvítací zprávy (*banner Message of the Day, MOTD*)
- uložení změn na směrovači
- ověření konfigurace a správné funkce směrovače

Globální konfigurační režim

```
Router>enable
Router#
Router#configure terminal
Router(config)#
```

Pojmenování směrovače

```
Router(config)#hostname <jméno>
```

Nastavení hesel

```
Router(config)#enable secret <heslo>
Router(config)#line console 0
Router(config-line)#password <heslo>
Router(config-line)#login
Router(config)#line vty 0 4
Router(config-line)#password <heslo>
Router(config-line)#login
```

Uvítací zpráva

```
Router(config)#banner motd # <zpráva> #
```

Znak # (*hash mark*, dvojitý kříž) vložíte pomocí pravý_ALT+X.

Konfigurace rozhraní

```
Router(config)#interface <typ> <číslo>
Router(config-if)#ip address <ip adresa> <maska>
Router(config-if)#description <popis, a dále například číslo na helpdesk posky-
tovatele služby>
Router(config-if)#no shutdown
```

Každé síťové rozhraní směrovače je (musí být) v jiné síti (podsíti). => jinak vznikne chyba překrývání (*overlap*) sítí (IOS tuto chybu detekuje a nepovolí ji provézt, smaže nově vkládanou překrývající se adresu).

Uložení konfigurace

Router#copy running-config startup-config

Kontrola výpisů příkazu SHOW

```
Router#show running-config
Router#show startup-config
```

```
Router#show ip route
Router#show ip interface brief
Router#show interfaces
```

Obsah a tvorba obsahu směrovací tabulky

Obsah směrovací tabulky

Výpis na směrovači: *Router#show ip route*

a na hostitelské počítači PC (ve Windows:**C:\>route print** a v Linuxu **$route**).

Obsahuje:

- u přímo připojené sítě (sousední sítě):
  C 192.168.1.0/24, is directly connected, FastEthernet0/0
- u vzdálené sítě (dostupné přes alespoň jeden další směrovač):
  - kód protokolu (statická nebo dynamická cesta),
  - cílovou síť/masku,
  - next hop (*gateway*) – IP adresa vstupního portu následujícího směrovače,
  - administrativní vzdálenost/metrika (např.: [120/1]).

Statické směrování (statická cesta)

Kód = S

Příklad výpisu:

S 192.168.5.0/24 [1/0] via 192.196.2.2, 00:00:20, Serial0/0/0

Použije se v následujících případech:

- Síť se skládá z pouze několika mála směrovačů. Použití dynamického směrovacího protokolu v tomto případě nemá žádný významný přínos. Naopak, dynamické směrování může přidat více režie na administrátora.
- Síť je do Internetu připojena pouze přes jednoho ISP. Není zde třeba dynamické směrování, protože ISP je jediným výstupním bodem ze sítě do Internetu.
- Velká síť konfigurovaná v topologii s jedním jediným centrálním zařízením (*hub-and-spoke topology*). Použití dynamického směrovacího protokolu je zbytečné, protože z každé větve sítě je do cíle pouze jedna cesta přes toto centrální zařízení.

Obvykle se používá kombinace statického a dynamického směrování. Než se nastavují cesty do vzdálených sítí, musí být nastaveny rozhraní a linky do přímo připojených sítí.

Dynamické směrování

Příklad výpisu:

R 192.168.4.0/24 [120/1] via 192.196.2.2, 00:00:20, Serial0/0/0

Sloupce: protokol, cílová síť/maska, brána (next-hop), stáří řádky, odchozí rozhraní (outgoing interface).

Dynamické směrovací protokoly jsou určeny pro sdílení informací o směrování mezi jednotlivými směrovači.

Základní činnosti směrovacího protokolu:

- automatické prozkoumávání sítě
- aktualizace a správa směrovacích tabulek

Směrovací protokoly pro IP

Pro IP existuje několik směrovacích protokolů. Zde je několik nejběžnějších dynamických směrovacích protokolů pro **směrování IP paketů**:

- RIP (Routing Information Protocol)
- IGRP (Interior Gateway Routing Protocol)
- EIGRP (Enhanced Interior Gateway Routing Protocol)
- OSPF (Open Shortest Path First)
- IS-IS (Intermediate System-to-Intermediate System)
- BGP (Border Gateway Protocol)

Principy směrovací tabulky.

Občas se v tomto kurzu odkazujeme na **tři principy vztahující se ke směrovací tabulce**, které vám pomohou porozumět, nastavit a odstranit chyby směrování. (Tyto principy jsou převzaty z knihy *Alex Zinin: Cisco IP Routing*.)

1. **Každý směrovač činí svá rozhodnutí samostatně a založené pouze na svojí vlastní směrovací tabulce.**
2. **Skutečnost, že jeden směrovač má ve své směrovací tabulce určité informace, neznamená, že ostatní směrovače mají tytéž informace.**
3. **Směrovací informace o cestě z jedné sítě do druhé neposkytují směrovací informace o opačné neboli zpětné cestě.** (Při asymetrickém směrování může být zpětná cesta jiná.)

Jaké jsou **důsledky těchto principů**?
Představte si, že máme za sebou propojené tři směrovače R1, R2 a R3. K R1 a R3 jsou připojeni klienti PC1 a PC2.

1. Po směrovacím rozhodnutí, směrovač R1 pošle paket adresovaný do PC2 na směrovač R2. R1 zná pouze informace ze své směrovací tabulky, které říkají, že R2 je další skok na cestě. R1 neví jestli R2 skutečně má či nemá cestu do cílové sítě.
2. Je v zodpovědnosti administrátora sítě zajistit, aby každý směrovač, který spravuje, měl úplné a správné směrovací informace, pomocí kterých mohou být pakety poslány mezi libovolnými dvěma sítěmi. To lze zajistit statickými cestami, dynamickými směrovacími protokoly nebo kombinací obojího.
3. Směrovač R2 je schopen odeslat paket směrem k cílové síti PC2. Přesto byl paket z PC2 do

PC1 zahozen směrovačem R2. Ačkoliv má R2 ve své směrovací tabulce informace o cílové síti paketu z PC1, nevíme zda má informace pro zpětnou cestu do sítě s PC1.

Asymetrické směrování

Protože směrovač nemusí mít nutně ve svých směrovacích tabulkách ty samé informace, pakety mohou cestovat sítí v jednom směru jednou cestou a zpátečním směrem jinou cestou. To se nazývá asymetrické směrování. Asymetrické směrování je běžnější v Internetu, který používá směrovací protokol BGP, než v interních sítích.

V tomto případě, když síť navrhuje a odstraňuje chyby, měl by administrátor ověřit následující směrovací informace:

Je cesta od zdroje k cíli dostupná v obou směrech?

Je cesta braná v obou směrech ta samá cesta? (Asymetrické směrování není neběžné, ale někdy může přinášet vznik dalších problémů.)

Určení cesty a přeposlání

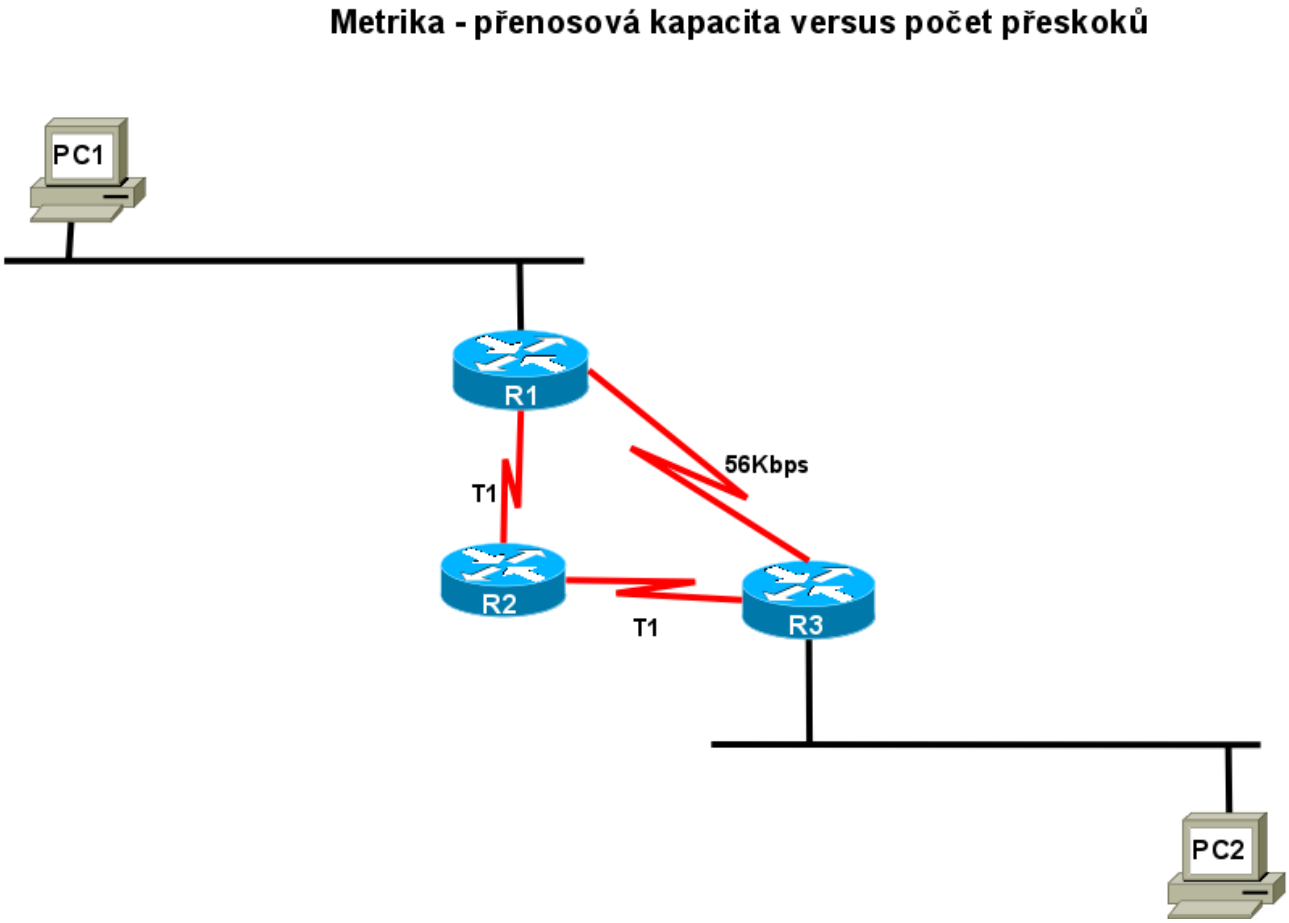
Zopakujte si nejprve strukturu IP paketu a rámce Ethernet. Význam a formát nejdůležitějších polí příslušné PDU.

- IPv4: Verze, životnost - TTL, zdrojová IP adresa, cílová IP adresa
- Ethernet: zdrojová MAC adresa, cílová MAC adresa, FCS

Nejlepší cesta a metrika

Směrování je nalezení v nějakém smyslu nejlepší cesty. Nejlepší cesta se potom vybírá podle nejmenší hodnoty Administrativní vzdálenosti (*Administrative Distance*, *AD*) a při stejné AD nejmenší metriky. Administrativní vzdálenost je číselné vyjádření kvality („ceny“, *cost*) či důvěryhodnosti (*trustfulness*) směrovacího protokolu, kterým byla vytvořena příslušná řádka ve směrovací tabulce[4]. Metrika je potom pro jeden konkrétní směrovací protokol vyjádření kvality (=ceny) linky (směru, cesty). Nejkvalitnější je ta cesta s číselně nejmenší metrikou.

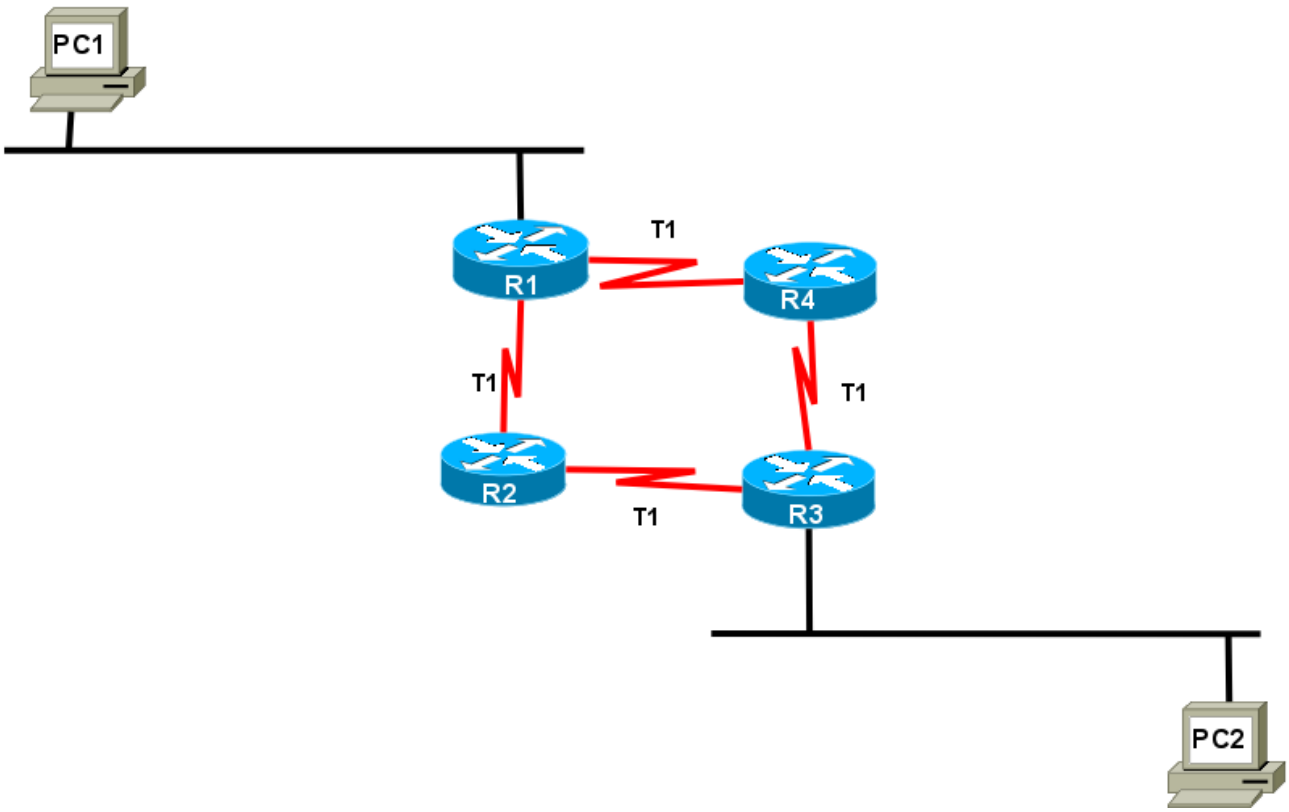
4 Na jednom směrovači může najednou běžet více různých směrovacích protokolů. Směrovací protokol může také pomocí tzv. Redistribuce přebírat směry z jiných směrovacích protokolů.



Vyvažování zátěže u cest se stejnou cenou

Pokud je více linek se stejnou cenou (administrativní vzdáleností a metrikou), byla by vždy používána pouze první linka (směr) ve směrovací tabulce. To není vždy vhodné, je zatížena jedna linka a ostatní nejsou zatížené vůbec, proto je možné na směrovači zapnout vyvažování zátěže cest se stejnou cenou (*equal cost load balancing*). Jednotlivé směry jsou potom cyklicky přepínány (*round robin aproach*).

Vyrovnávání zátěže při stejné ceně trasy (Equal Cost Load Balancing)



(Směrovací protokol EIGRP podporuje i vyvažování zátěže pro cesty s různými cenami.)

Proces zapouzdřování a odpouzdřování, tok dat z uzlu na uzel

Proces zapouzdřování (*encapsulation*) paketu do rámce přidáním L2 záhlaví a zápatí a proces odpouzdřování (*decapsulation*) rámce na paket odstraněním L2 záhlaví a zápatí je jedna ze základních činností směrovače.

Zjednodušený postup zpracování dat na směrovači, když přijme paket z jedné sítě, který je adresovaný do jiné sítě.

1. Odpouzdří L3 paket oddělením hlavičky L2 rámce.
2. Použije cílovou IP adresu v hlavičce IP paketu k nalezení adresy dalšího skoku ve směrovací tabulce (=směrování = nalezení nejlepší cesty do cílové sítě).
3. Zapouzdří L3 paket do nového L2 rámce a pošle ho ven z odchozího rozhraní (směrem na next-hop) (= přepínání paketu).

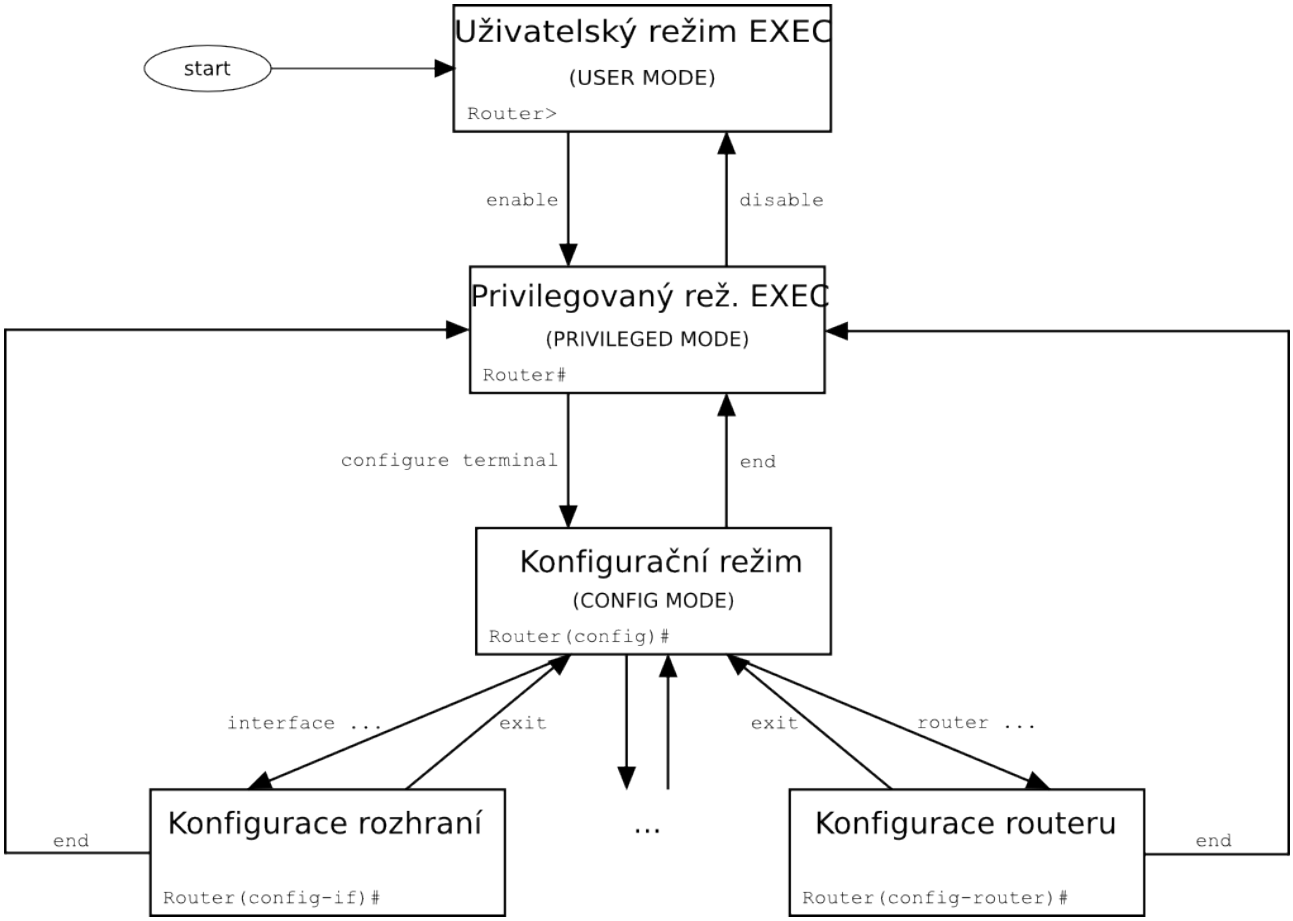
Průvodce základní konfigurací (nastavením) směrovače

V této podkapitole jsou uvedeny informace a příkazy týkající se následujících oblastí:

- Konfigurace routeru a to zvláště:

- názvy (*Names*)
- hesla (*Passwords*)
- rozhraní (*Interfaces*)
- uvítací zprávy (*MOTD banners*)
- tabulky IP hostitelů (*IP host tables*)
- ukládání a rušení vašich nastavení (*Saving and erasing your configurations*)

- příkazy show, ke kontrole nastavení (konfigurací) směrovače.



(*Zdroj obrázku: modifikace http://CS.Wikipedia.org/wiki/Cisco_IOS*)

Režimy směrovače

Router>	Uživatelský (User EXEC)
Router#	Privilegovaný (Enable EXEC)
Router(config)#	Globální konfigurační
Router(config-if)#	Konfigurace rozhraní
Router(config-subif)#	Konfigurace pod-rozhraní (Subinterface)
Router(config-line)#	Konfigurace linky (vty, console)
Router(config-router)#	Konfigurace (nastavení) směrovacího protokolu

TIP: Existují ještě další režimy (módy) než zde uvedené. Ne všechny příkazy pracují ve všech režimech. Proto si dávejte pozor. Jestliže napíšete příkaz (například show run) a vrátí se chyba, ujistěte se, že jste ve správném režimu. (**I když umíte používat nápovědu „?“, musíte vědět v jakém režimu lze ten který příkaz použít.**)

Globální konfigurační mód

Router>	Vidíte nastavení, ale nlze měnit
Router#	Vidíte nastavení a můžete se přesunout so režimu, kde lze změnit.
Router#config t Router(config)#	Přechod do globálního konfiguračního režimu Tento prompt indikuje, že můžete začít dělat změny.

Nastavení názvu směrovače

Tento příkaz pracuje jak na směrovačích tak i na přepínačích.

Router(config)#hostname Cisco **Cisco**(config)#	Název může být jakékoliv slovo si zvolíte. **Změní prompt (systémovou výzvu).**

Nastavení hesel

Pracuje jak na směrovačích tak i přepínačích.

~~Router(config)#enable password cisco~~	Nastaví nešifrované heslo příkazu enable (na slovo cisco). NEPOUŽÍVEJTE
Router(config)#enable secret class	Nastaví šifrované heslo příkazu enable (na slovo class).
Router(config)#line con 0 Router(config-line)#password console Router(config-line)#login	Vstup do režimu konfigurace konzole Nastaví heslo konzolové linky na **console** Vynutí ověření hesla při přihlášení (login).
Router(config)#line vty 0 4 Router(config-line)#password telnet Router(config-line)#login	Vstup do režimu linky virtuálního terminálu (vty) – Telnet pro všech 5 linek vty Nastaví heslo vty na slovo telnet. Vynutí ověření hesla při přihlášení (login).
Router(config)#line aux 0 Router(config-line)#password backdoor Router(config-line)#login	Vstup do režimu konfigurace pomocné linky (auxiliary line mode) Nastaví heslo pomocné linky na slovo **backdoor.** Vynutí ověření hesla při přihlášení (login).

Varování: Heslo *Enable secret* je implicitně šifrované a *Enable password* není. Z tohoto důvodu je doporučena praxe nikdy nepoužívat heslo *enable password*. Při konfiguraci směrovače používejte pouze heslo *enable secret.* (Pokud jsou z historických důvodů nastavena obě, musí být z bezpečnostních důvodů různá a *enable secret* má přednost.)

Varování: Nelze nastavit stejné heslo v obou příkazech *enable secret* a *enable password*. Pokud by to šlo, anulovalo by to šifrování.

Šifrování hesel

Router(config)#service password-encryption	Na všechna hesla je aplikováno **slabé šifrování**.
Router(config)#enable password *cisco*	Nastavuje heslo *enable password* na slovo **cisco**
Router(config)#line con 0	...
Router(config-line)#password *cisco*	Pokračuje nastavování hesel stejně jako nahoře.
	...
Router(config)#no service password-encryption	Vypne šifrování hesel.

Varování: Jestliže zapnete službu šifrování hesel, použije se, a jestliže ji potom opět vypnete, všechna zašifrovaná hesla zůstanou zašifrovaná. Nová hesla budou již ovšem nezašifrovaná.

Příkazy show

Router#show ?	Vypíše všechny dostupné příkazy show.
Router#show interfaces	Zobrazí statistiky pro všechna rozhraní.
Router#show interface serial 0	Zobrazí statistiky pro určité rozhraní, v tomto případě Serial 0.
Router#show ip interface brief	Zobrazí přehled všech rozhraní, včetně stavu (status) a přiřazené IP adresy.
Router#show controllers serial 0	Zobrazí statistiky pro HW rozhraní. Statistiky zobrazují zda jsou nastaveny hodiny (*clock rate*) a zda je kabel DCE, DTE, nebo není připojen.
Router#show clock	Zobrazí čas nastavený na zařízení.
Router#show hosts	Zobrazí lokální lokální „kešovanou“ tabulku hosts (lokální hostitel-IP adresa). Zde jsou názvy a IP adresy hostitelů v síti, ke kterým se můžete připojit.
Router#show users	Zobrazí všechny uživatele připojené k zařízení.
Router#show history	Zobrazí historii použitých příkazů.
Router#show flash	Zobrazí informace o paměti Flash.
Router#show version	Zobrazí informace o zavedené verzi SW.
Router#show arp	Zobrazí tabulku ARP.
Router#show protocols	Zobrazí stav nastavených protokolů L3.
Router#show startup-config	Zobrazí konfiguraci uloženou v paměti NVRAM.

Router#show running-config	Zobrazí konfiguraci aktuálně běžící v RAM.

Názvy rozhraní

Jedním z největších problémů, kterým čelí noví administrátoři jsou názvy rozhraní na různých modelech směrovačů. V následující tabulce jsou vypsány názvy rozhraní pro Ethernet, Fast Ethernet, a Sériových rozhraní na směrovačích řady 2500, 1700 a 2600.

Pevná rozhraní (řada 2500)	**Modulární (výměnná) rozhraní (řada 1700)**	**Modulární (výměnná) rozhraní (řada 2600)**
Router(config)#interface *type port*	Router(config)#interface *type port*	Router(config)#interface *type slot/port*
Router(config)#int serial 0 (s0)	Router(config)#intterface serial 0	Router(config)#interface serial 0/0 (s0/0)
Router(config)#int ethernet 0 (e0)	Router(config)#interf ace fastethernet 0	Router(config)#int fastethernet 0/0 (fa0/0)

Přechod mezi rozhraními

To co se děje ve sloupci 1 je totéž co ve sloupci 2.

Router(config)#int s0	Router(config)#int s0	Přechod do režimu konfigurace rozhraní S0.
Router(config-if)#exit	Router(config-if)#int e0	Z int S0, přechod do E0.
Router(config)#int e0	Router(config-if)#	Nyní v režimu E0
Router(config-if)#		Prompt se nezměnil. Dávejte si pozor.

Konfigurace sériového rozhraní

Router(config)#int s0/0	Přechod do režimu konfigurace rozhraní Serial 0/0.
Router(config-if)#description Link to ISP	Volitelný popis linky může být důležitý – něco významného.
Router(config-if)#ip address 192.168.10.1 255.255.255.0	Přiřadí adresu a masku podsítě k rozhraní.
Router(config-if)#clock rate 56000	Přiřadí takt hodin (*clock rate*) pro rozhraní **DCE**. (v Kb)
Router(config-if)#no shutdown	Zapne rozhraní. (**Implicitně je rozhraní směrovače vždy administrativně vypnuté!**)

TIP: Příkaz nastavení taktu hodin (*clock rate*) se použije jen na sériovém rozhraní, do kterého je zastrčen DCE kabel (na druhé straně kabelu resp. v jeho polovině, pokud jsou spojené, je konektor **V.35 Female**). Takt hodin musí být nastaven na každé sériové lince mezi směrovači. Nezáleží na tom, do kterého směrovače je zastrčen DCE kabel, nebo do kterého rozhraní je kabel zastrčen. Se-

rial 0 na jednom směrovači může být zastrčen do Serial 1 na druhém směrovači.

Konfigurace rozhraní Ethernet/FastEthernet

Router(config)#int fa0/0	Přechod do režimu konfigurace rozhraní Fast Ethernet 0/0.
Router(config-if)#description Accounting LAN	Volitelný popis linky může být důležitý – něco lokálně významného.
Router(config-if)#ip address 192.168.20.1 255.255.255.0	Přiřadí adresu a masku podsítě k rozhraní.
Router(config-if)#no shut	Administrativně zapne rozhraní.

Vytvoření uvítacího hlášení (MOTD Banner)

Router(config)#banner motd # This is a secure system. Authorized Personnel Only! # Router(config)#	# (dvojitý kříž, hash mark) je známý jako oddělovací znak. Oddělovací znak musí obklopovat text uvítacího hlášení z obou stran a nesmí být použit uvnitř samotného textu hlášení. Znak # vložíte pomocí pravý_ALT+X.

Nastavení časového pásma (Clock Time Zone)

Router(config)#clock timezone EST –5	Nastaví časovou zónu pro zobrazení. Založeno na světovém času UTC (coordinated universal time). (EST - Eastern Standard Time (na východě USA) je 5 hodin za UTC, SEČ je UTC + 1 hodina = 1 hodinu před.)

Přiřazení lokálního jména hostitele k IP adrese

Router(config)#ip host london 172.16.1.3	Přiřadí jméno/název hostitele k IP adrese. Po tomto přiřazení můžete použít jméno místi IP adresy v příkazu *telnet* nebo *ping*.
Router#ping london = Router#ping 172.16.1.3	

TIP: Implicitní číslo portu v příkazu *ip host* je 23, nebo Telnet. Když se chcete připojit Telnetem k nějakému zařízení, pouze vložte samotné jméno zařízení:
Router#london = Router#telnet london = Router#telnet 172.16.1.3

Příkaz *no ip domain-lookup* (vypnutí překladu jména na IP adresu)

Router(config)#no ip domain-lookup Router(config)#	Vypíná automatický vyhodnocení neznámého příkazu jako lokální jméno hostitele.

TIP: Musíte čekat minutu nebo dvě, vždy když napíšete příkaz nesprávně a když směrovač zkouší přeložit váš příkaz na doménový server 255.255.255.255? Směrovač je implicitně nastaven na

pokus vyhodnotit jakékoliv slovo, které není příkaz, na DNS serveru s adresou 255.255.255.255. Jestliže se nechystáte nastavit DNS server, vypněte tuto vlastnost, abyste šetřili čas - zvlášť pokud jste špatný písař.

Příkaz *logging synchronous*

Router(config)#line con 0 Router(config-line)#logging synchronous	Zapne synchronní logování (synchronous logging). Informace odesílané na konzoli nebudou přerušeny příkazem, který píšete na klávesnici. Příkaz se přesune na novou řádku.

TIP: Objeví se příkaz, který píšete, vždy uprostřed řádky? Ztratíte pozici v řádku? Nevíte, kde jste v příkazu, a tak stisknete ENTER a začnete vše od začátku? Příkaz *logging synchronous* řekne směrovači, že jestliže se na display zobrazují nějaké informace, měl by se prompt a nový příkaz přesunout na novou řádku pokud se snažíte něco psát na klávesnici, aby vás to nemátlo. Jestliže jste pokračovali v psaní, měl by se příkaz vykonat správně, i když na obrazovce vypadá zobrazený špatně.

Příkaz *exec-timeout*

Router(config)#line con 0 Router(config-line)#exec-timeout 0 0 Router(config-line)#	Nastavuje časový limit pro automatické odhlášení konzole. Nastavení na 0 0 (minuty sekundy) znamená, že konzole nebude nikdy automaticky odhlášena.

TIP: Příkaz *exec-timeout 0 0* je skvělý v laboratoři, protože se konzole nikdy neodpojí. Ale v reálném světě je to velmi nebezpečné (špatná bezpečnost).

Uložení konfigurace

Router#copy run start	Uloží aktuální běžící konfiguraci (running-config) do lokální paměti NVRAM.
Router#copy run tftp	Uloží aktuální běžící konfiguraci na vzdálený TFTP server.

Smazání počáteční konfigurace

Router#erase start	Smaže soubor *startup-config* z paměti NVRAM.
Router#reload	Znovu zavedení operačního systému po smazání počáteční konfigurace. => prázdná aktuální běžící konfigurace

TIP: Running-config je ale i po smazání startup-config z NVRAM stále v dynamické operační paměti RAM. Znovu natažení (*Reload*) OS směrovače smaže aktuální konfiguraci running-config.

Smazání předchozí konfigurace (erase start) proveďte, pokud Vám nebude řečeno jinak, na začátku a na konci každého praktického cvičení se směrovači v laboratoři datových sítí. Je to slušnost vůči ostatním studentům.

Příklad konfigurace: základní nastavení směrovače

Síťová topologie pro příklad: Zleva doprava. **PC1**, překřížený kabel UTP, rozhraní FastEthernet 0/0 **směrovače Plzeň** –rozhraní Serial 0/0 (172.16.20.1, maska 255.255.255.0), **strana DCE** sériového kabelu - sériová linka, rozhraní Serial 0/0 (172.16.20.2, maska 255.255.255.0) - **směrovač Praha**, přímý UTP kabel, **switch**, přímý UTP kabel, **PC2**. Jednotlivé **IP sítě zleva doprava**: 172.16.10.0/24, 172.16.20.0/24 a 172.16.30.0/24.

1. **Nakreslete si nejprve detailní schéma zapojení (fyzickou topologii sítě) s IP adresami a názvy rozhraní. Zapište si též i adresy a masky jednotlivých sítí.**
2. **Případně i vyplňte tabulku adres rozhraní.**
3. **Ověřte návrh (výpočet) adresace (že to, co má být v jedné síti je skutečně v jedné síti a co má být v různých sítích je v různých sítích).**

Nastavení pro směrovač Plzen (jsou zadávané zkrácené příkazy)

Prompt (výzva uživatele) a příkaz	***Popis***
Router>en	Vstup do privilegovaného režimu.
Router#clock set 18:30:00 15 Mar 2008	Nastaví lokální čas na směrovači.
Router#config t	Vstup do globálního konfiguračního režimu.
Router(config)#hostname Plzen	Nastaví jméno směrovač na Plzen.
Plzen(config)#no ip domain-lookup	Vypne překlad jmen pro neznámé příkazy (chyby pravopisu).
Plzen(config)#banner motd # This is the Plzen Router. Authorized Access Only #	Vytvoří uvítací hlášení (MOTD banner).
Plzen(config)#clock timezone SEC 1	Nastaví časovou zónu na SEČ (+1 od UTC)
Plzen(config)#enable secret cisco	Nastaví šifrované heslo privilegovaného režimu *enable secret* na slovo *cisco*.
Plzen(config)#service password-encryption	Hesla budou šifrována slabou šifrou.
Plzen(config)#line con 0	Vstup do režimu nastavení konzole.
Plzen(config-line)#logging synchronous	Příkazy nebudou přerušeny nevyžádanými hlášeními nebo zprávami.
Plzen(config-line)#password class	Nastaví heslo na slovo *class*.
Plzen(config-line)#login	Umožní kontrolu hesla při přihlášení.
Plzen(config-line)#line vty 0 4	Přesun do režimu konfigurace virtuálních linek Telnet od 0 do 4 (implicitně je jich 5).
Plzen(config-line)#password class	Nastaví heslo na slovo *class*.
Plzen(config-line)#login	Umožní kontrolu hesla při přihlášení.
Plzen(config-line)#line aux 0	Přesun do režimu konfigurace pomocné linky (auxiliary line mode).

Plzen(config-line)#password class	Nastaví heslo na slovo *class*.
Plzen(config-line)#login	Umožní kontrolu hesla při přihlášení.
Plzen(config-line)#exit	Přesun zpět do režimu globální konfigurace.
Plzen(config)#no service password-encryption	Vypne šifrování hesel.
Plzen(config)#int fa 0/0	Přesun do režimu konfigurace Fast Ethernet 0/0.
Plzen(config-if)#desc Engineering LAN	Nastaví lokálně významný popis rozhraní.
Plzen(config-if)#ip address 172.16.10.1 255.255.255.0	Přiřadí IP adresu a masku podsítě k rozhraní.
Plzen(config-if)#no shut	Administrativně zapne rozhraní.
Plzen(config-if)#int s0/0	Přesun přímo do režimu konfigurace Serial 0/0.
Plzen(config-if)#desc Linka na smerovac Praha	Nastaví lokálně významný popis rozhraní.
Plzen(config-if)#ip address 172.16.20.1 255.255.255.0	Přiřadí IP adresu a masku podsítě k rozhraní.
Plzen(config-if)#clock rate 56000	Nastaví takt hodin pro synchronní sériový přenos. (V tomto rozhraní musí být zastrčen kabel **DCE.**)
Plzen(config-if)#no shut	Administrativně zapne rozhraní.
Plzen(config-if)#exit	Přesun zpět do režimu globální konfigurace.
Plzen(config)#ip host Praha 172.16.20.2	Nastaví překlad lokálního jména hostitele Praha na IP adresu 172.16.20.2.
Plzen(config)#exit	Přesun zpět do privilegovaného režimu.
Plzen#copy run start	Uložení aktuální běžící konfigurace do paměti NVRAM
Plzen#reload	Znovu natažení operačního systému a konfigurace. Pokud nebyla konfigurace uložena, je nenávratně ztracena.

Zdroj: CCNA2 Companion Guide, Cisco Press, kapitola 3 + úpravy

Obnova zapomenutého hesla pro směrovače Cisco

(*Password Recovery Procedure*[5])

Krok	***Příkazy pro řadu 2500***	***Příkazy pro řady 1700/2600/ISR***
Krok 1: Při zavádění OS přerušte zavádění.	Stiskněte CTRL+BREAK >	Stiskněte CTRL+BREAK[6] rommon 1>

5 Postup pro konkrétní síťové zařízení naleznete v jeho dokumentaci pod tímto anglickým označením.

6 Ukončovací sekvence závisí na použitém operačním systému, ve které běží emulátor terminálu a na nastavení samotného emulátoru (typicky pro Windows Ctrl+Break nebo Ctrl+C.)

Krok 2: Změna konfiguračního registru, aby byl ignorován obsah konfigurace v paměti NVRAM.	>o/r 0x2142 >	rommon 1>confreg 0x2142 rommon 2>
Krok 3: Znovuzavedení OS.	>i	rommon 2>reset
Krok 4: Vstup do privilegovaného režimu. (Nevstupujte do interaktivního konfiguračního režimu setup.)	Router>enable Router#	Router>enable Router#
Krok 5: Kopie startovací konfigurace do běžící konfigurace.	Router#copy startup-config running-config ...<vynechaný výstup>... Denver#	Router#copy startup-config running-config ...<vynechaný výstup>... Denver#
Krok 6: Změna hesla.	Denver#configure terminal Denver(config)#enable secret *new* Denver(config)#	Denver#configure terminal Denver(config)#enable secret *new* Denver(config)#
Krok 7: Znovunastavení konfiguračního registru do na jeho implicitní hodnotu.	Denver(config)#config-register 0x2102	Denver(config)#config-register 0x2102
Krok 8: Uložení konfigurace.	Denver(config)#exit Denver#copy running-config startup-config Denver#	Denver(config)#exit Denver#copy running-config startup-config Denver#

IOS Escape Sequence

Kdykoliv potřebujete ukončit běh určitého příkazu v operačním systému IOS, použijte jako **ukončovací sekvenci** (*escape sequence*) trojhmat **Shift+Ctrl+6**.

Popřípadě **uspat relaci Telnet pomocí Shift+Ctrl+6 X** . A opět obnovit stiskem ENTER na prázdné řádce.

Kontrolní opakovací otázky a odpovědi (kvíz):

1) Kroky při startu směrovače:
   a) provedení testu HW POST z ROM,
   b) spuštění zavaděče systému (bootstrap) z ROM,
   c) nalezení a zavedení operačního systému z paměti flash,
   d) zavedení konfiguračního souboru z NVRAM.
2) Dva příkazy, které by technik mohl použít pro zjištění IP adres na sériovém rozhraní směrovače:

a) show interfaces

b) show ip interface brief

3) Pravdivé tvrzení o směrování:

a) Každý směrovač se rozhoduje sám a pouze na základě svojí směrovací tabulky.

4) Dvě základní činnosti směrovacího protokolu:

a) objevuje nové sítě,

b) aktualizuje a udržuje obsah směrovací tabulky.

5) Administrátor konfiguruje nový směrovač. Jsou nastaveny IP adresy a masky ale nikoliv směrovací protokol nebo statické cesty. Které směry jsou v této chvíli ve směrovací tabulce?

a) Přímo připojené sítě.

6) Jak směrovač přeposílá pakety?

a) Pokud je cílová IP adresa v přímo připojené síti, směrovač odešle paket z odchozího rozhraní uvedeného ve směrovací tabulce pro cílovou síť.

b) Pokud je cílová IP adresa ve vzdálené síti, směrovač odešle paket na další skok (next hop) uvedený ve směrovací tabulce pro cílovou síť.

7) Definice metriky:

a) Metrika je kvantitativní hodnota, kterou směrovací protokol hodnotí cenu, náklady pro určitou cestu/směr.

8) Administrátor nastavil na směrovači příkaz „ip route 0.0.0.0 0.0.0.0 serial0/0“. Jak se tento příkaz projeví ve směrovací tabulce za předpokladu, že je rozhraní serial0/0 zapnuté?

a) S* 0.0.0.0/0 is directly connected, Serial0/0

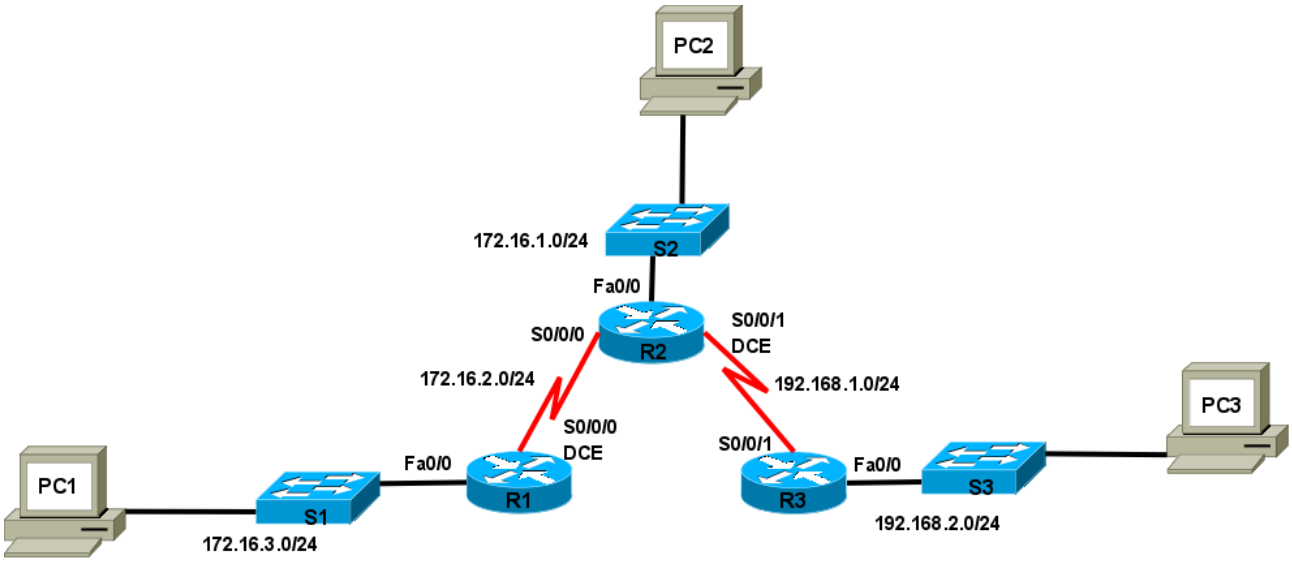
Kapitola 2 – Statické směrování

V této kapitole se naučíme:

- Definovat všeobecnou roli, kterou směrovač hraje v síti
- Popsat přímo připojené sítě a různá rozhraní směrovače
- Prozkoumat přímo připojené sítě ve směrovací tabulce a použít protokol CDP
- Popsat statické cesty s jejich odchozími rozhraními
- Popsat sumarizovanou a implicitní cestu
- Prozkoumat jak jsou pakety přeposílány když je použita statická cesta
- Spravovat statické cesty a odstraňovat jejich závady

Role směrovače v síti

Propojovací zařízení mezi jednotlivými sítěmi, které směruje a přepíná pakety a dále slouží k segmentaci (rozdělení a zmenšení) domén všesměrového vysílání.



Topologie a tabulka adres

Před zahájením práce je nanejvýš vhodné si připravit:

1. **Topologické schéma sítě (z názvy, IP adresami a maskami rozhraní i jednotlivých sítí)**
2. **Tabulka adres (například):**

Zařízení	*Rozhraní*	*IP adresa*	*Maska podsítě*	*Brána*
R1	Fa0/0	172.16.3.1	255.255.255.0	N/A
	Serial0/0/0	172.16.2.1	255.255.255.0	N/A
...				

Použití kabeláže

Router je do LAN zapojen obvykle přes rozhraní Ethernet nebo Fast Ethernet a s LAN komunikuje přes hub či switch. K propojení routeru a switche/hubu je třeba přímý (*straight-through*) kabel.
Rozhraní 10BASE-TX či 100BASE-TX vyžadují UTP kabel kategorie 5 a lepší. Při propojení routeru přímo k počítači či jinému routeru je třeba křížený (*crossover*) kabel.
Je třeba použít správné rozhraní; mnohé konektory pro různá rozhraní vypadají stejně. Např. rozhraní pro Ethernet, ISDN BRI, konzolový, integrovaný CSU/DSU[7] a Token Ring používají stejný osmi-pinový konektor – RJ-45, RJ-48 nebo RJ-49.
Pro služby WAN potřebuje zákazník vybavení, což je často router jako DTE. Je připojen k poskytovateli služeb skrze DCE jednotku, tou bývá modem nebo CSU/DSU; ta slouží k převodu dat z DTE do podoby vhodné pro poskytovatele služeb WAN.
Nejčastějšími rozhraními routerů pro WAN jsou sériová rozhraní (konektor SmartSerial nebo DB60) (za ním je ale ještě připojen modem nebo CSU/DSU, telekomunikační vedení a na druhé straně opět modem nebo CSU/DSU a sériová linka vedoucí do dalšího směrovače). Je třeba dávat pozor na typ kabelu, typ konektoru a zda jde o DTE či DCE jednotku **(DTE nebo DCE stranu V.35 kabelu, DCE je tam, kde je konektor V.35 Female na opačné straně ke SmartSerial rozhraní)**.
Z příkazové řádky (CLI) směrovače stranu sériového kabelu na portu zjistíte příkazem **show controllers ...** .

Poznámka k zapojení dvoudílného sériového kabelu se zástrčkou *SmartSerial* do výměnného modulu rozhraní (WIC-2A/S) směrovačů C1720:

- **delší zástrčku** *SmartSerial* (strana **DCE**) zapojit **do spodní zásuvky Serial0** (na tomto rozhraní typu DCE nezapomeňte nastavit takt hodin „***clock rate***“)
- **kratší zástrčku** *SmartSerial* (strana **DTE**) zapojit **do horní zásuvky Serial1**

Připojení LAN (propojení mezi zařízeními)

Port nebo připojení	*Typ portu*	*Připojené do*	*Barva (originální kabelu Cisco)*	*Kabel*
Ethernet	**RJ-45**	**Ethernet switch**	**Světle žlutá**	**RJ-45**
T1/E1 WAN	RJ-48C/CA81A	T1 nebo E1	Světle zelená	RJ-48 T1
Konzole	**8 pin**	**COM port počítače**	**Světle modrá**	**Rollover RJ-45 (redukce DB-9)**
AUX	**8 pin**	**Modem**	Černá **(světle modrá)**	**Rollover RJ-45 (redukce DB-25)**

7 Digitální modem (*Channel Service Unit/Data Service Unit*).

BRI S/T	RJ-48C/CA81A	zařízení NT1 nebo Private Integrated Network Exchange (PINX)	Oranžová	RJ-45
BRI U WAN	RJ-49C/CA11A	ISDN	Oranžová	RJ-45

Určení typu kabelu propojujícího zařízení

Jestliže zařízení A má/je:	***Jestliže zařízení B má/je:***	***Potom použijte tento kabel:***
Sériový (COM) port počítače	Konzolový port směrovače/přepínače	Konzolový (Rollover)
Síťová karta (NIC) počítače	Switch	Přímý (Straight-through)
Síťová karta (NIC) počítače	Síťová karta (NIC) počítače	Překřížený (Crossover)
Switch port	Ethernetový port směrovače	Přímý (Straight-through)
Switch port	Switch port	Překřížený (Crossover) (vypněte uplink)
Ethernetový port směrovače	Ethernetový port směrovače	Překřížený (Crossover)
Síťová karta (NIC) počítače	Ethernetový port směrovače	Překřížený (Crossover)
Sériový port směrovače	Sériový port směrovače	Sériový DCE/DTE Cisco

Propojení konektorů pro různé typy kabelů

Straight-Through Cable 568A/568A	***Crossover Cable 568A/568B***	***Rollover Cable***
Pin 1 – Pin 1	Pin 1 – Pin 3	Pin 1 – Pin 8
Pin 2 – Pin 2	Pin 2 – Pin 6	Pin 2 – Pin 7
Pin 3 – Pin 3	Pin 3 – Pin 1	Pin 3 – Pin 6
Pin 4 – Pin 4	Pin 4 – Pin 4	Pin 4 – Pin 5
Pin 5 – Pin 5	Pin 5 – Pin 5	Pin 5 – Pin 4
Pin 6 – Pin 6	Pin 6 – Pin 2	Pin 6 – Pin 3
Pin 7 – Pin 7	Pin 7 – Pin 7	Pin 7 – Pin 2
Pin 8 – Pin 8	Pin 8 – Pin 8	Pin 8 – Pin 1

Standardy 568A a568B

568A Standard				***568B Standard***			
Pin	***Barva***	***Pár***	***Popis***	***Pin***	***Barva***	***Pár***	***Popis***
1	Bílá/zelený	3	RecvData +	1	Bílá/oranžový	2	TxData +
2	Zelená	3	RecvData -	2	Oranžová	2	TxData -

3	Bílá/oranžový	2	Txdata +	3	Bílá/zelený	3	RecvData +
4	Modrá	1	Nepoužitý	4	Modrá	1	Nepoužitý
5	Bílá/modrý	1	Nepoužitý	5	Bílá/modrý	1	Nepoužitý
6	Oranžová	2	TxData -	6	Zelená	3	RecvData -
7	Bílá/hnědý	4	Nepoužitý	7	Bílá/hnědý	4	Nepoužitý
8	Hnědá	4	Nepoužitý	8	Hnědá	4	Nepoužitý

Tx – vysílání
Recv – příjem

Typy linek WAN

- Pronajatá linka – synchronní sériová,
- Přepínaný okruh – asynchronní sériová, ISDN L1,
- Přepínané pakety – synchronní sériová – poskytovatel.

Zkoumání obsahu směrovací tabulky a stavu rozhraní

```
Router1>show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.1.0/24 is directly connected, FastEthernet0/0
S    192.168.2.0/24 [1/0] via 192.168.3.1
     192.168.3.0/30 is subnetted, 1 subnets
C       192.168.3.0 is directly connected, FastEthernet0/1
```

```
Router2#show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol

FastEthernet0/0        192.168.2.254   YES manual up                    up

FastEthernet0/1        192.168.3.1     YES manual up                    up
```

```
Vlan1                  unassigned      YES manual administratively down down
Router2#
```

Pole **status** (stav) ve výpisu **show ip interfaces brief** (i v jiných výpisech) může nabývat následujících hodnot a významů:

- **up -** rozhraní je zapnuto/pracuje
- **down** – rozhraní je vypnuto/nepracuje
- **administratively down** – rozhraní je vypnuto administrátorem, respektive nebylo povoleno při nastavení.

Stav rozhraní a linkového protokolu:

1. Pokud je rozhraní down, pak je down také linkový protokol, protože neexistuje funkční médium. Pokud je rozhraní administratively down, znamená to že je rozhraní administrátorem vypnuté.
2. Stav linkového protokolu, spolu s protokolem sítě LAN, který nad ním pracuje. Pokud je linkový protokol down, není na druhé straně zapnuto a nastaveno rozhraní nebo není kabel v rozhraní.

Stav rozhraní/protokolu a typy možné chyby

Interface	***Protokol linky***	***Typ chyby***
UP	UP	L1 a L2 OSI modelu pracují v pořádku a případné chyby jsou výsledkem činnosti vyšších vrstev.
UP	DOWN	Chyba na L2 OSI modelu. - Chyba protokolu na L2 nebo chyba zapouzdření L2 (například značkování rámců ve VLAN protokolem IEEE 802.1Q).
DOWN	DOWN	Závada na L1 OSI modelu. - Kabely, fyzické rozhraní, další routery musí být prověřeny na přítomnost napájení a správnou instalaci a konfiguraci.
DOWN	UP	Duplikace MAC adresy v lokální síti připojené k rozhraní Ethernet, nebo chyba servisního modulu na interní rozšiřující kartě. **Na routerech lze administrativně měnit (klonovat) MAC adresu rozhraní, na rozdíl od běžné (starší) síťové karty.**

Příkaz show interfaces

Jeden z nejdůležitějších příkazů show je **show interfaces**, který vypíše status a statistiky na všech portech směrovače. **Show interfaces <jméno rozhraní>** vypíše stav a statistiky požadovaného rozhraní (např. show interfaces serial 0/0). Pomocí show interfaces se mohou zjistit problémy na fyzické vrstvě, hardware a logické vrstvě nebo software.

Další informace vypsané pomocí **show interfaces** o rozhraní

- IP adresa ,
- MAC adresa ,

- maska podsítě ,
- statistické údaje o síti ,
- poslední vynulování čítače
- výskyt chyb

```
Router1#sh interfaces fa0/1
FastEthernet0/1 is up, line protocol is up (connected)
  Hardware is Lance, address is 0005.5ec2.7b02 (bia 0005.5ec2.7b02)
  Internet address is 192.168.3.2/30
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec, rely 255/255, load 1/255
  Encapsulation ARPA, loopback not set
  ARP type: ARPA, ARP Timeout 04:00:00,
  Last input 00:00:08, output 00:00:05, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue :0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 input packets with dribble condition detected
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
Router1#
```

Průzkum přímo připojených sítí a protokol CDP

CDP (Cisco Discovery Protocol) je proprietární protokol vytvořený firmou Cisco běžící na 2. vrstvě (L2). Tento protokol běží na drtivé většině síťových zařízeních (Cisco ale i jiných firem) a je používán na sdílení informací o jiných **přímo připojených sousedních (*neighbor*[8]) zařízeních (na L2)**[9].

CDP se používá k získání HW platformy, IP adresy a názvu rozhraní sousedních zařízení. CDP je

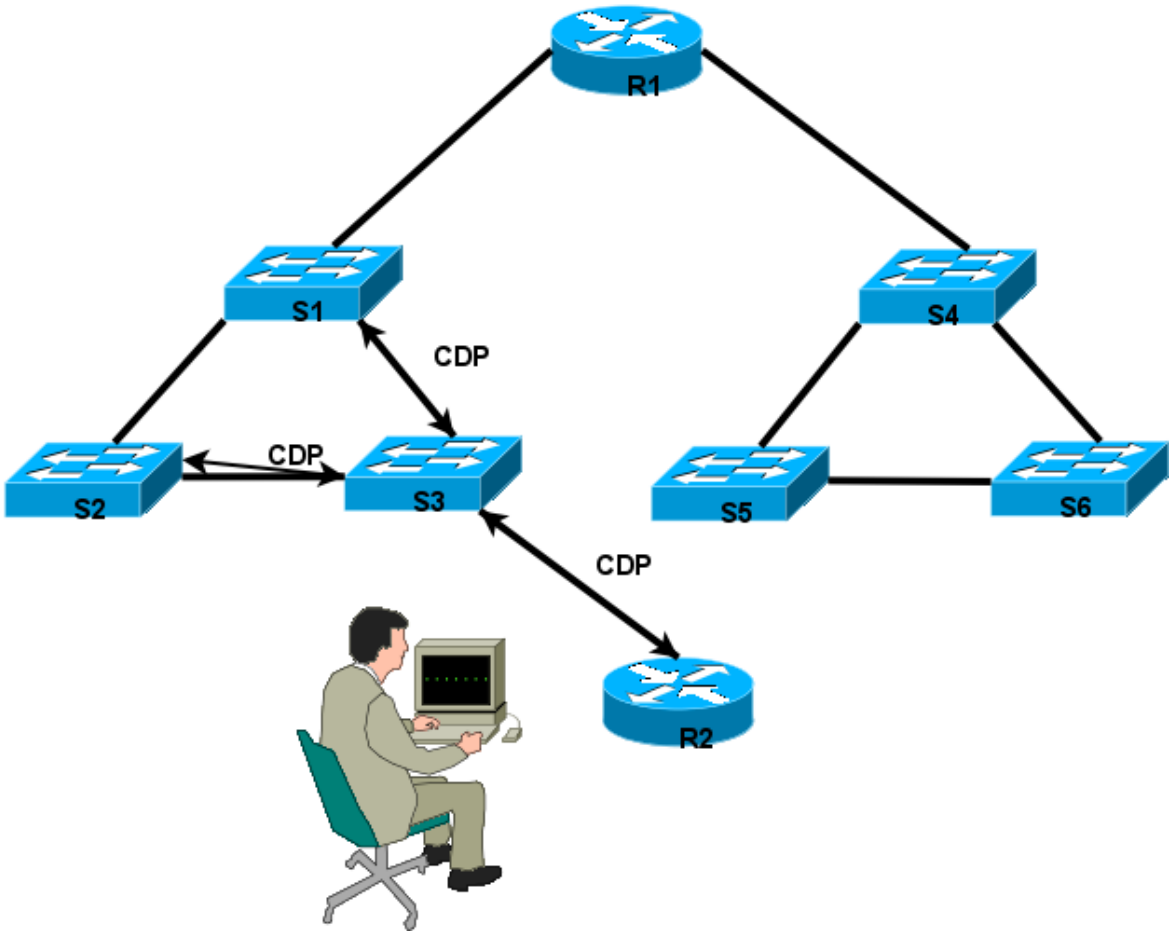
8 *Neighbor* = soused v americké angličtině. V britské angličtině častěji *neighbour*.

9 Pojem soused: na L2 jde skutečně o fyzicky sousední zařízení, (sousední na L3 = je ve stejné broadcastové doméně, ve stejné síti – sousední směrovač).

nezávislý na přenosovém médiu a linkovém (i síťovém) protokolu a běží na Cisco routerech, můstcích, přístupových serverech i přepínačích[10].

Každé zařízení, které je konfigurované pro CDP posílá periodické zprávy, tzv. oznamovače (*advertisements*). Oznamovače obsahují různé informace jako je: HW platforma, IP adresa a název rozhraní, doba uchování (*holdtime*) což je čas, po kterou si přijímací zařízení ponechá konkrétní CDP informaci, než ji smaže. Každé zařízení také zjišťuje obsah CDP zpráv odeslaných z jiných zařízení, aby zjistilo informace o jednotlivých sousedech.

CDP - Cisco Discovery Protocol



Příkazy pro protokol CDP

Příkaz	***Účel***
cdp enable	Povolí CDP na konkrétním rozhraní – v (config-if)# (je implicitně zapnuté)
cdp advertise-v2	Povolí CDPv2 na rozhraní
clear cdp counters	Resetuje čítače provozu na nulu – v enable režimu#
show cdp	V privilegovaném (enable) režimu: Zobrazuje interval mezi vysíláními CDP oznamovačů (CDP advertisement),

10 Existují i klienti a agenti (= servery) CDP pro hostitelské počítače.

	počet sekund, po které je oznamovač platný pro daný port a verzi oznamovače.
show cdp entry entry-name [protocol | version]	Zobrazuje informace o určitém sousedovi, které mohou být omezeny protokolem nebo verzí.
show cdp interface [type number]	Zobrazuje informace o rozhraních, na kterých je CDP umožněn
show cdp neighbors [type number] [detail]	Zobrazuje typy zařízení, které byly objeveny, jméno zařízení, číslo a typ lokálního rozhraní nebo portu, počet sekund, po které je oznamovač platný pro port, typ zařízení, číslo typu, duplexní mód, VTP doménu asociovanou k sousednímu zařízení, pokud bylo použito klíčové slovo detail.
cdp run	(Je implicitně zapnuté.) CDP lze v globálním konfiguračním režimu vypnou příkazem (config)#**no cdp run**. V tomto případě je třeba ho znovu zapnout příkazem (config)#cdp run.

Co zobrazují výpisy z CDP:

- **Device ID** (identifiers) – například nastavení host name u přepínače
- Address list – alespoň jednu adresu na síťové vrstvě pro každý podporovaný protokol
- **Port ID** (identifiers) – jméno lokálního a vzdáleného portu v podobě řetězce ASCII znaků například ethernet0
- **Capability** – například zda je zařízení switch nebo router
- **Platform** – HW platforma zařízení, například „Cisco **7200** series router“

```
Switch>show cdp
Global CDP information:
    Sending CDP packets every 60 seconds
    Sending a holdtime value of 180 seconds
    Sending CDPv2 advertisements is enabled
```

```
Switch>show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone
Device ID     Local Intrfce   Holdtme    Capability   Platform     Port ID
S1            Fas 0/1          146           S       2950         Fas 0/1
R3            Fas 0/0          146           R       C2600        Fas 0/0
```

```
Switch>show cdp neighbors detail

Device ID: R3
Entry address(es):
  IP address : 192.168.3.254
Platform: cisco C2600, Capabilities: Router
Interface: FastEthernet0/0, Port ID (outgoing port): FastEthernet0/0
Holdtime: 146

Version :
Cisco Internetwork Operating System Software
IOS (tm) C2600 Software (C2600-I-M), Version 12.2(28), RELEASE SOFTWARE (fc5)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2005 by cisco Systems, Inc.
Compiled Wed 27-Apr-04 19:01 by miwang

advertisement version: 2
Duplex: full
```

Zjišťování změn ve směrovací tabulce

Pokud vypadne přímo připojená síť, ze směrovací tabulky okamžitě zmizí všechny směry (cesty), které se na tuto přímo připojenou síť odkazují. (Sítě, jejichž „next-hop“ ve vypadlé přímo připojení síti leží, nebo sítě, které do ní přepínají na odchozím rozhraní.)

Uvedeme příklad: ladění (*debugging*, *debug*): Zapnuto. Vytažen kabel ze směrovače (příslušné cesty jsou smazány ze směrovací tabulky). Ladění vypnuto.

POZOR: ladění (*debugging*) velice zatěžuje CPU směrovače, zapínejte ho proto pouze na nezbytnou dobu pro dohledávání chyb (a nikoliv při běžném provozu).

Zapnutí ladění:

```
Router1#debug ip routing
IP routing debugging is on
Router1#
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state
to down
RT: interface FastEthernet0/0 removed from routing tableRT: del 192.168.1.0 via
0.0.0.0, connected metric [0/4294967295]
RT: delete network route to 192.168.1.0
RT: NET-RED 192.168.1.0/24
```

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state
to up

RT: interface FastEthernet0/0 added to routing tableRT: SET_LAST_RDB for
192.168.1.0/24

    NEW rdb: is directly connected


RT: add 192.168.1.0/24 via 0.0.0.0, connected metric [0/0]

RT: NET-RED 192.168.1.0/24

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state
to down

RT: interface FastEthernet0/1 removed from routing tableRT: del 192.168.3.0 via
0.0.0.0, connected metric [0/4294967295]

RT: delete network route to 192.168.3.0

RT: NET-RED 192.168.3.0/30

RT: del 192.168.2.0 via 192.168.3.1, static metric [1/0]

RT: delete network route to 192.168.2.0

RT: NET-RED 192.168.2.0/24


Vypnutí ladění:

Router1#undebug all

All possible debugging has been turned off

Router1#
```

Statická cesta s adresou dalšího skoku

Nastavení:

```
Router1(config)#ip route 192.168.2.0 255.255.255.0 192.168.3.1
```

Obsah směrovací tabulky:

```
Router1#show ip route

<vynecháno>

C    192.168.1.0/24 is directly connected, FastEthernet0/0

S    192.168.2.0/24 [1/0] via 192.168.3.1

     192.168.3.0/30 is subnetted, 1 subnets

C       192.168.3.0 is directly connected, FastEthernet0/1

Router1#
```

V tomto případě se při směrování do vzdálené sítě (192.168.2.0) musí nejprve nalézt adresa dalšího skoku (next-hop) a po jejím nalezení **rekurzivně vyhledat** (*recursive lookup*) odchozí rozhraní do přímo připojené sítě, ve které je následující skok.

Statická cesta s odchozím rozhraním

Nastavení:

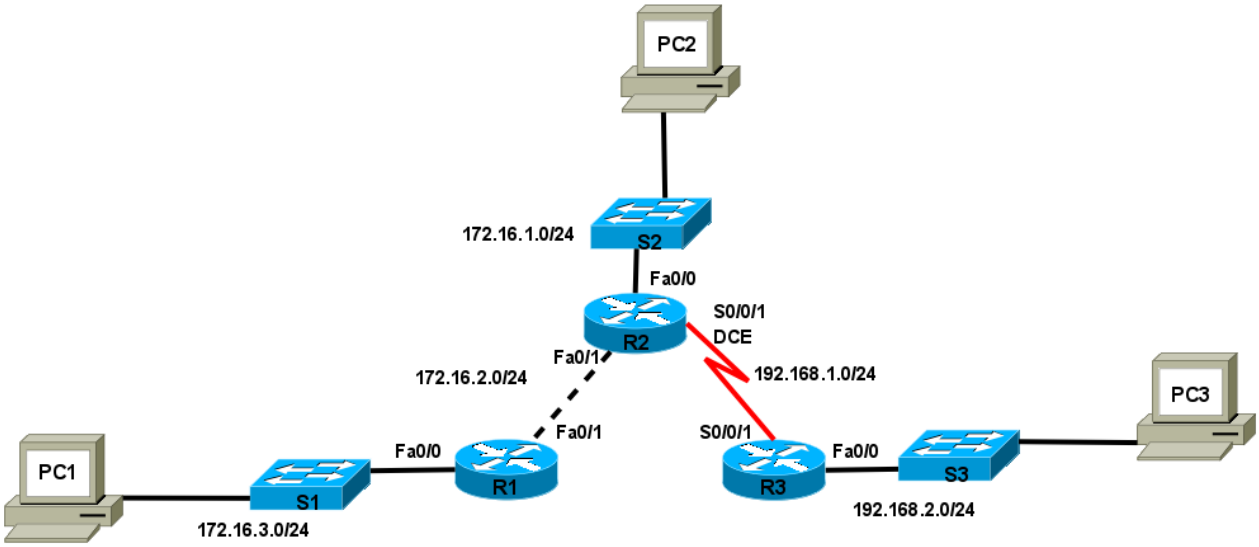
```
Router2(config)#ip route 192.168.1.0 255.255.255.0 Fa0/1
```

Obsah směrovací tabulky:

```
Router2#sh ip route
<vynecháno>
S    192.168.1.0/24 is directly connected, FastEthernet0/1
C    192.168.2.0/24 is directly connected, FastEthernet0/0
     192.168.3.0/30 is subnetted, 1 subnets
C       192.168.3.0 is directly connected, FastEthernet0/1
Router2#
```

V tomto případě se staticky definovaná vzdálená síť jeví jako přímo připojená a po nalezení cesty (směru) se může paket rovnou odeslat příslušným směrem (ber rekurzivního vyhledávání adresy dalšího skoku). Tato varianta je všeobecně doporučována. Její nevýhoda se ale projeví při Ethernetové síti s vícenásobným přístupem (*multi-acces network*) tj.v síti s více branami, kdy se nedá nalezenout MAC adresa následujícího skoku pro zapouzdření paketu do rámce.

Odchozí rozhraní a nebo adresa dalšího přeskoku (Exit interface and next-hop address)



R1(config)#ip route 192.168.2.0 255.255.255.0 FastEthernet 0/1
R1(config)#ip route 192.168.2.0 255.255.255.0 172.16.2.2

Sumarizace (agregace) cest

Zatím jsme se vždy zabírali stavem, kdy každý směrovač má ve své směrovací tabulce informace o všech sítích v určité skupině sítí, kde nastavujeme statické směrování. Pokud máme hierarchickou stromovou strukturu podsítí, je vhodné zavést **sumarizaci cest** (*route summarization*), také známé pod pojmem **agregace cest** (*route aggregation*). Informace o všech sítích ve skupině mají pak pou-

ze směrovače v této skupině (podsítí) a ostatní směrovače (mimo skupinu) mají pak pouze cestu na hraniční směrovač celé skupiny, tato cesta pak ukazuje na „**nadsíť**“ (*supernet*) celé skupiny = sumarizace (sumarizovaná síť = síť s kratší maskou než podsítě, která pokrývá všechny adresy v jednotlivých podsítích). Šetří se tím řádky ve směrovacích tabulkách a směrování probíhá rychleji.

Příklad:

Máme skupinu podsítí s rozsahem: 192.168.1.0/27 až 192.168.1.120/29. Určete síť a masku pro sumarizaci.

Postup: najdeme síť s nejmenší a největší adresou (určíme adresu všesměrového vysílání (broadcast) pro tuto síť s největší adresou) a odečteme od sebe. Inverzí rozdílu (včetně eventuálního doplnění binárních jedniček vpravo za vedoucí jedničku) získáme masku a odmaskováním zadaných podsítí touto maskou dostaneme sumární síť. 192.168.1.127 – 192.168.1.0 = 0.0.0.127 inverze (dvojkový doplněk) je 255.255.255.128 tj. /25 a odmaskováním adres podsítí získáme sumární síť 192.168.1.0/25.

Implicitní cesta

Implicitní cesta (*default route*, *Gateway of last resort*) se nastavuje jako speciální případ statické cesty pro cílové sítě mimo naši správu obvykle na našeho poskytovatele - ISP. Na tuto cestu je paket poslán, pokud směrovač nenalezl cílovou síť v předchozích řádkách směrovací tabulky. (Nezapomeňte, že směrovací tabulka je setříděna sestupně podle masky.) Implicitní cesta je ve směrovací tabulce označena hvězdičkou (*) jako kandidát implicitní cesty (*candidate default*).

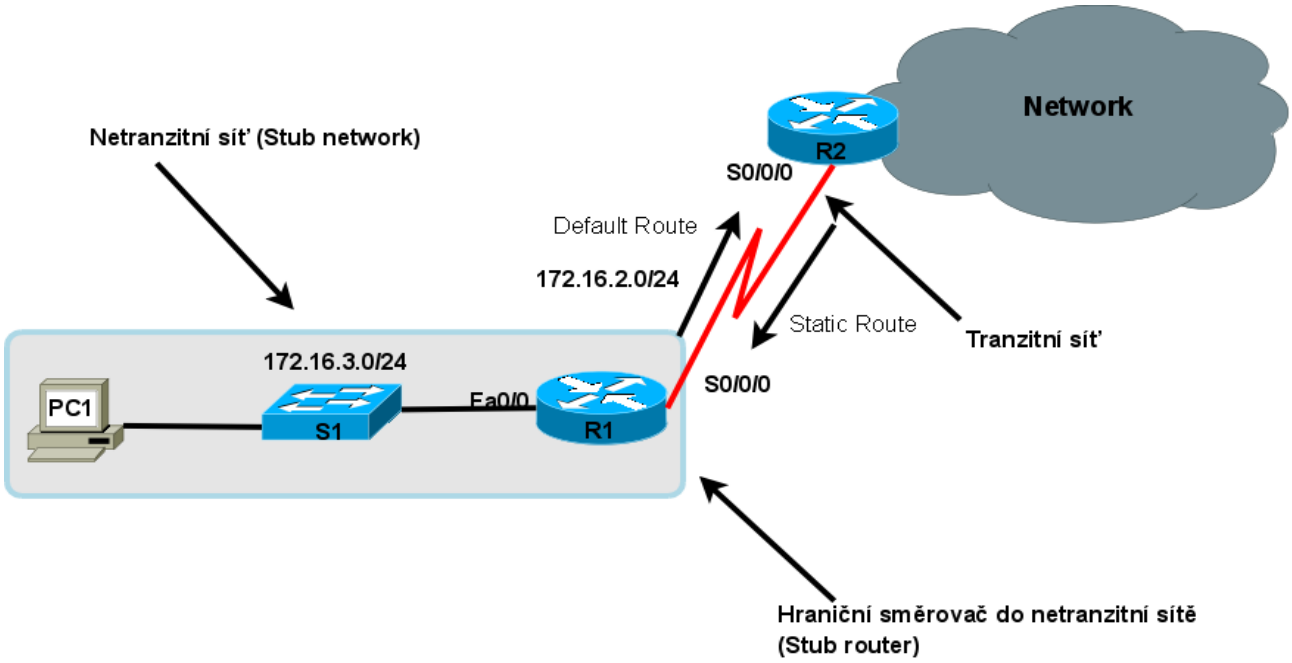
```
Router(config)#ip route 0.0.0.0 0.0.0.0 [exit-interface | ip-address ]
```

```
Router1#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

C    192.168.1.0/24 is directly connected, FastEthernet0/0
S    192.168.2.0/24 [1/0] via 192.168.3.1
     192.168.3.0/30 is subnetted, 1 subnets
C       192.168.3.0 is directly connected, FastEthernet0/1
S*   0.0.0.0/0 is directly connected, FastEthernet0/1
```

```
Router1#
```

Poznámka:
Plná syntaxe příkazu je:
Router(config)# **ip route** *prefix mask* {*ip-address* | *interface-type interface-number*} [*distance*] [**tag** *tag*] [**permanent**]

- *Distance* je vzdálenost – lze použít k nastavení tzv. ***Floating route***, záložní statické cesty, která připadne do úvahy pouze při výpadku dynamického směrování (parametr vzdálenost - *distance* se musí nastavit větší než je *administrativní vzdálenost* (*Administrative Distance, AD*)[11] dynamického směrovacího protokolu).
- *Tag* – poskytuje metodu pro rozlišení mezi interními cestami (získanými od stejného směrovacího protokolu jako na směrovači) a externími cestami (získaných od jiných protokolů). Tento nepovinný volitelný atribut můžete přidat během redistribuce směrovacích protokolů.
- *Permanent* - cesta nezmizí ze směrovací tabulky, ani když spadne odchozí interface.



Definice implicitní sítě

Jiný způsob jak nastavit bránu poslední záchrany (*gateway of last resort*) je nastavení implicitní sítě (pro přilehlou, přímo připojenou, síť).

Router(config)# **ip default-network** *network-number*

Ve směrovací tabulce je potom (místo statické implicitní cesty) přímo připojená síť (C) jako kandidát na implicitní síť, například :

```
C*   10.0.0.0/8 is directly connected, Serial0/0
```

11 Administrativní vzdálenosti viz kapitola 3.

Správa a modifikace cest

Pokud chcete některou cestu odstranit ze směrovací tabulky, provedete stejným příkazem, jako byla nastavena a přidáním příkazu **no**. Pouhé přidání nové cesty do stejné sítě nebo na stejné odchozí rozhraní by vedlo k existenci dvou cest.

To znamená, pokud chcete vytvořit novou cestu do stejné cílové sítě, musíte nejprve odstranit starou cestu a potom vložit novou.

Hledání a odstraňování chyb statické cesty

Příkazy:

- **ping**
- **traceroute**
- **show ip route**
- **show ip interface brief**
- **show cdp neighbors detail**

Odstraňování problémů se statickým směrováním

Hledání a odstraňování chyb může někdy být docela pracné. Je proto vhodné mít pečlivě zdokumentovanou strukturu sítě a nejprve ověřit, že je to tak opravdu nastavené.

Na ověření dostupnosti cílového zařízení slouží příkaz:

```
ping <cílová_adresa>
```

Zobrazit jednotlivé směrovače, kterými musí paket projít do cílového zařízení umožňuje příkaz:

```
traceroute <cílová_adresa>
```

Odpovědi (a jejich významy) na příkaz ping na směrovači

Znak	***Popis***
!	Přijetí jedné odpovědi (jednoho paketu).
.	Čas pro odpověď síťového serveru vypršel.
U	Cíl je nedostupný (*Unreachable*) – přijat PDU.
C	Zahlcení (*Congestion*) – přijat paket.
I	Uživatel přerušil (*Interrupted*) test.
?	Neznámý typ paketu.
&	Překročena životnost paketu.

Odpovědi (a jejich významy) na příkaz traceroute na směrovači

Znak	***Popis***
nn msec	Pro každý uzel, doba cyklu (v milisekundách) pro určený počet pokusů.
*	Doba testu vypršela. V určené době nebyla přijata žádná odpověď.
?	Neznámá chyba.
A	Administrativně nedostupné. Obvykle tento výstup indikuje, že ACL[12] blokuje provoz.
H	Hostitel je nedostupný.
N	Síť (*Network)* je nedostupná (mimo rozsah).
P	Protokol je nedostupný.
Q	Zdroj vypnut (*Quenche*).
P	Port je nedostupný.

Příkazy pro kapitolu 2, Statické směrování

Příkaz (Command)	***Popis (Description)***
Router# show controllers serial 0/0/0	Zobrazí, která strana kabelu je připojená do rozhraní: DCE nebo DTE
Router# show cdp neighbors	Zobrazí přehledné sumární informace o přímo připojených L2 zařízeních; používá proprietární protokol Cisco (CDP)
Router# show cdp neighbors detail	Zobrazí úplné informace o přímo připojených L2 zařízeních; používá proprietární protokol Cisco (CDP)
Router(config)# cdp run	Povolí Cisco L2 protokol pro celý směrovač.
Router(config-if)# cdp enable	Povolí Cisco L2 protokol pro rozhraní.
Router(config)# ip route 10.0.0.0 255.0.0.0 172.16.0.1	Nastaví statickou cestu do 10.0.0.0/8 s dalším skokem (*next hop*) 172.16.0.1 .
Router(config)# ip route 10.0.0.0 255.0.0.0 Serial0/0/0	Nastaví statickou cestu do 10.0.0.0/8 s odchozím (výstupním) rozhraním Serial 0/0/0 .
Router(config)# ip route 172.16.0.0 255.240.0.0 Serial 0/0/0	Nastaví sumární statickou cestu pro všechny adresy spadající do privátní sítě třídy B v rozsahu od 172.16.0.0/16 do 172.31.0.0/16; použije odchozí rozhraní Serial 0/0/0

12 ACL – Access Control List – přístupový seznam – povolení nebo blokování síťového provozu pro adresu popřípadě protokol.

Router(config)# ip route 0.0.0.0 0.0.0.0 Serial0/1/0	Nastaví implicitní statickou cestu (*default static route*) s odchozím rozhraním Serial 0/1/0 .

Komplexní praktické laboratorní cvičení – statické směrování

Mějme 3 směrovače R1, R2 a R3 (Cisco 1841) jsou zapojené do kruhu a ke každému je připojeno jedno PC: PC1, PC2 a PC3.
Mezi R1 a R2 je sériová linka s adresou sítě a maskou (délkou prefixu): 172.16.2.32/27.
Mezi R2 a R3 je FastEthernet s adresou sítě a maskou (délkou prefixu): 172.16.1.0/24.
Mezi R3 a R1 je FastEthernet s adresou sítě a maskou (délkou prefixu): 172.17.2.0/25.
Na jednotlivých směrovačích R1, R2 a R3 je připojeno PC1, PC2 a PC3 v sítích 192.168.1.0/24, 192.168.2.0/24 a 192.168.3.0/24.
IP adresy rozhraní jsou vypočteny dle následujících firemních směrnic vlastníka sítě („*best practices*“): na směrovačích jsou adresy těsně pod adresou všesměrového vysílání v příslušné síti a na klientech jsou IP adresy těsně nad adresou sítě dané sítě.

Zapojení zprovozněte pomocí statického směrování (cestu definujte vždy nejkratším směrem a směrem na odchozí rozhraní (*outgoing interface*), nikoliv na IP adresu dalšího skoku (*next hop*)).

Postup práce:

1. Nejprve si nakreslete topologické schéma zapojení včetně adres sítí.
2. Vyplňte (doplňte) následující tabulku adres síťových rozhraní:

Zařízení	*Rozhraní*	*IP adresa*	*Maska*	*Brána*
R1	Fa0/0	172.17.2.126	255.255.255.128	-
	Eth0/0/0	192.168.1.254	255.255.255.0	-
	S0/1/0	172.16.2.61	255.255.255.224	-
R2				-
R3				-
PC1	Fast Ethernet	192.168.1.1	255.255.255.0	192.168.1.254
PC2	Fast Ethernet	192.168.2.1	255.255.255.0	192.168.2.254
PC3	Fast Ethernet	192.168.3.1	255.255.255.0	192.168.3.254

3. Propojte zapojení odpovídající kabeláží, případě, při nedostatku portů, přidejte odpovídající zásuvné moduly. (POZOR: nelze použít 4 portový switch modul – nelze na něm nastavovat IP adresy.) Názvy síťových rozhraní si doplňte do topologického schématu sítě.
4. Nastavte na správná rozhraní adresy a masky a zprovozněte jednotlivé linky (zkontrolujte pomocí příkazu **show ip interteface brief** funkčnost vrstev L1, L2 (L3).
5. Nastavte příslušné jméno hostitele (**hostname**) (R1, R2 a R3) na každém ze směrovačů.
6. Na každém směrovači zprovozněte 4 linky Telnet s heslem: class.
7. Na každém směrovači zaheslujte šifrovaným heslem režim enable, heslo: cisco.
8. Zkontrolujte směrovací tabulky příkazem **show ip route** na každém směrovači (v této chvíli byste měli vidět na každém 3 přímo připojené sítě).
9. Na každém směrovači nastavte implicitní cestu ve směru hodinových ručiček. Ověřte

funkčnost popřípadě opravte chyby. Všimněte si, že je to asymetrické směrování. (**Vznikla směrovací smyčka**[13] pro všechny IP adresy ležící mimo zadaných 6 sítí.) Po vyzkoušení ODSTRAŇTE. Implicitní cesty se **nepoužívají** pro směrování do předem známých sítí. Takto navržené implicitní cesty do kruhu zapříčiňují vznik směrovací smyčky.

10. Na každém směrovači **nastavte statické cesty** do tří vzdálených sítí definované na odchozí rozhraní, směr zvolte vždy nejkratší cestou.
11. Zkontrolujte funkčnost a opravte případné chyby. (Použijte příkazy: **ping, traceroute, show ip route, show ip interface brief**).
12. Vyzkoušejte si reakce (změny ve směrovací tabulce) v závislosti na výpadku přímo připojené sítě (o výpadku vzdálené se směrovač při statickém směrování sám nedozví).
13. Nastavte a zaheslujte telnet a zaheslujte šifrovaným heslem režim enable.
14. Zkuste nakreslit zapojení znovu pouhým průzkumem pomocí protokolu CDP příkazem **show cdp neighbors detail** a přihlašováním se na sousední zařízení pomocí Telnet. (Před tím „jakoby zapomeňte“, co o síti již víte.)

Časté a „oblíbené“ chyby

Typy častých chyb vyskytující se na prvních třech vrstvách modelu OSI:

Vrstva 1:

- Porušené kabely,
- Odpojené kabely,
- Kabely zapojené **do nesprávných portů**,
- Použití nesprávných typů kabelů (přímé, křížené, roll-over apod),
- Problémy s příjmem a vysíláním signálu,
- DTE/DCE kabely (přehozené strany kabelu),
- Vypnuté porty (nezapnuté příkazem *no shutdown*)
- Vypnutá zařízení (vypnuté napájení).

Vrstva 2:

- Nesprávně nastavená sériová rozhraní,
- Nesprávně nastavená ethernetová rozhraní,
- Nesprávně nastavené zapouzdření (budeme brát později - ve třetím semestru),
- Nesprávně nastavené hodiny (*clock rate*) na sériovém rozhraní (souvisí s DCE kabelem na L1),
- Problémy se síťovými kartami (HW).

Vrstva 3:

13 Směrovací smyčka viz kapitola 4.

- Nekorektní IP adresy (například každý konec kabelu v jiné IP síti),
- Nekorektní masky podsítě (jiná maska => jiná síť, byť se třeba sítě překrývají),
- Použití zakázaných IP adres (použita například adresa sítě nebo broadcastu) – odhalí přímo IOS směrovače.
- Překryv sítí (*overlapping*) (na jednom směrovači odhalí přímo IOS směrovače).
- Na klientech špatně nastavena (nebo vůbec nenastavena) výchozí brána,
- Špatně nastavené statické směrování (např. vznik směrovací smyčky),
- Vypnutý směrovací protokol,
- Nesprávné nastavení směrovacího protokolu.

Testování sítí

Většina síťových problémů souvisí s nemožností připojit se k požadovanému hostiteli nebo službě. Problémy s konektivitou mají celou řadu podob, jako je například vypršení časového limitu při pokusu o spojení, pokus o terminálové spojení bez příslušné odpovědi ze strany hostitele a podobně.

Zapamatujte si že,

- Pokud je konkrétní hostitel A dostupný z hostitele B v jiné síti, naprosto to nevypovídá nic o tom, zda je dostupný hostitel B z opačného směru z A. Směrování jedním směrem je nezávislé na směrování opačným směrem.
- Vznik **směrovací smyčky** nezávisí na existenci fyzické smyčky na médiu.

„Chybičky“

Co bychom již měli opravdu znát:

1. Každý klient IP sítě musí mít nastaveno: IP adresu, masku podsítě a implicitní bránu (a DNS server, pokud ho používáme).
2. Každé použité síťové rozhraní na směrovači musí být zapnuté a mít nastavenou IP adresu a masku.
3. Všechny IP adresy hostitelů připojené k jednomu rozhraní (portu) směrovače musí ležet v jedné (stejné) IP síti (podsíti).
4. Každý port jednoho směrovače musí ležet v jiné síti (podsíti). Vznik překryvu sítí směrovač ohlásí (*overlap*).
5. Ve skupině routerů pod jednou správou směrování musí být všechny jednotlivé sítě různé.
6. Pro propojení portů je třeba použít správné typy kabelů.
7. Na DCE straně sériového kabelu je třeba mít nastaveny hodiny.
8. Při splnění předchozího, by již neměl být žádný problém při statickém směrování.
9. **Sumarizace** má smysl pouze na hranici (kořenu) třídní sítě. (Šetří potom řádky v aktualizacích směrovacích tabulek posílaných směrem výše.)
10. Při dynamickém směrování je třeba rozlišit, zda se protokolem přenáší maska či nikoliv

(**masku nepřenáší pouze RIPv1**) – a tomu musí odpovídat navržené adresní schéma, to budeme brát později.

11. Pokud se nepřenáší maska, směrovací protokol IP adresu odmaskuje implicitní maskou, to znamená, že dvě a více podsítí jedné sítě se mu jeví jako jedna nadsíť (v plné třídě) a pokud jsou tyto sítě ve dvou a více různých směrech v nesouvislé síti, tak je problém.
12. Funkční směrování jedním směrem naprosto nezaručuje směrování opačným směrem.
13. Příliš mnoho změn při konfiguraci směrovače „najednou“ může vést k jeho atypickým stavům – je vhodné uložit konfiguraci a restartovat směrovač (#copy run start, #reload).

Kontrolní opakovací otázky a odpovědi (kvíz):

1) Pravdivé tvrzení ohledně konfigurace statické cesty:
   a) Směrovače s nastavenou statickou cestou používající adresu dalšího skoku (*next hop*) musí mít buď v této cestě ještě nastaveno odchozí rozhraní (*outgoing interface*) nebo mít ještě jednu cestu se sítí (přímo připojenou), ve které je další skok, a s přiřazeným odchozím rozhraním. (V tomto případě se potom při směrování provádí rekurzivní dohledání cesty s odchozím rozhraním do adresy dalšího skoku.)
2) Ve směrovací tabulce je řádka „**S 10.0.0.0/8 [1/0] via 172.16.40.2**“. Administrátor má tuto cestu ze směrovací tabulky odstranit. Jakým příkazem?
   a) no ip route 10.0.0.0 255.0.0.0 172.16.40.2
3) Jaký příkaz by mohl vyprodukovat následující výstup?

```
Router1>
1     53 ms     43 ms     36 ms     10.0.0.1
2     106 ms    56 ms     40 ms     192.168.0.2
3     *         *         *         Request timed out
```

   a) traceroute
4) Tři charakteristiky statické cesty:
   a) snižuje požadavky na paměť a výpočetní výkon směrovače,
   b) používají se na směrovačích které jsou připojeny do netranzitních sítí (*stub networks*),
   c) používají se u takových sítí, ze kterých je pouze jedna cesta do konkrétní cílové sítě (= stromová struktura sítě).
5) Co je funkcí příkazu „show cdp neighbors“?
   a) Zobrazuje typ portů a HW platformu sousedících směrovačů nebo přepínačů Cisco. (CDP je sice proprietární protokol, ale používají ho i někteří jiní výrobci síťových zařízení.)
6) Pravdivé tvrzení ohledně přímo připojených směrů:
   a) Objeví se ve směrovací tabulce pokud je na rozhraní nastavená IP adresa a po vydání příkazu „show interfaces“ se ukáže, že příslušné rozhraní je administrativně zapnuté a L2 protokol linky je spuštěný.

7) Spárování příkazů a popisů jejich výstupů:
   a) show ip route = zobrazí všechny známé sítě
   b) show interfaces = zobrazí detailní informace o všech rozhraních
   c) degbug ip routing = zobrazí online informace potřebné pro odstraňování závad
   d) show interface brief = zobrazí struční informace o rozhraních (včetně stavu rozhraní a stavu L2 protokolu linky)
   e) show cdp neighbors = zobrazí přímo přípojené směrovače
   f) show controllers = zobrazí informace o DTE/DCE nastaveních.

Kapitola 3 - Protokoly pro dynamické směrování

V této kapitole se naučíme:

- Popsat roli dynamických směrovacích protokolů a místo těchto protokolů v kontextu návrhu moderních sítí
- Určit několik způsobů jak klasifikovat směrovací protokoly
- Popsat, jak směrovací protokol používá metriku a určit různé druhy metrik, které používají různé dynamické směrovací protokoly
- Určit administrativní vzdálenost (*Administrative Distance*, *AD*) cesty a popsat její důležitost při průběhu směrování
- Určit některé klíčové informace ve směrovací tabulce
- Realistický pohled na daná existující omezení na zařízeních, protokolech a adresních schématech.

Účel směrovacích protokolů

- Zjištění vzdálených (= ne přímo připojených) sítí (*Discovery of remote networks*)
- Udržování aktuálních směrovacích informací (*up-to-date routing information*)
- Výběr nejlepší cesty do cílových sítí (*the best path to destination networks*)
- Schopnost nalézt novou nejlepší cestu pokud současná cesta již není dále dostupná.

Které jsou **klíčové komponenty směrovacího protokolu**?

- **Datové struktury** – některé směrovací protokoly používají pro svou činnost tabulky nebo databáze. Tyto informace jsou uloženy v RAM.
- **Algoritmus** – algoritmus je konečný seznam kroků potřebných k dosažení určitého cíle. Směrovací protokoly používají algoritmy pro získání směrovacích informací a pro určení nejlepší cesty.
- **Zprávy směrovacího protokolu** (*Routing Protocol Messages*) – směrovací protokoly používají různé typy zpráv ke zjištění sousedních směrovačů, výměnu směrovacích informací a další úkoly, aby zjistily a udržely správné informace o síti.

Činnost směrovacího protokolu: všechny směrovací protokoly mají stejný účel – zjistit vzdálené sítě a rychle se přizpůsobit při změně topologie. Metoda, kterou směrovací protokol užívá závisí na použitém směrovacím algoritmu (typu směrovacího protokolu) a směrovacím protokolu samotném. Obecně může být činnost směrovacího protokolu popsána následovně:

- Směrovač posílá a přijímá směrovací zprávy na svých rozhraních.
- Směrovač sdílí směrovací zprávy a směrovací informace s jinými směrovači, které používají stejný směrovací protokol.
- Směrovače si vyměňují směrovací informace, aby zjistily vzdálené (*remote*) sítě.
- Když směrovač zjistí změnu topologie, může tuto změnu oznámit ostatním směrovačům

(pomocí tzv. Oznamovačů (*advertisement*)).

Porovnání dynamického a statického směrování

Charakteristika	*Dynamické směrování*	*Statické směrování*
Náročnost konfigurace	Obecně nezávislé na velikosti sítě	Čím větší síť, tím složitější
Požadované znalosti administrátora	Pokročilé	Nejsou třeba zvláštní
Změny topologie	Automatické přizpůsobení	Zásah administrátora je nutný
Škálovatelnost	Vhodné pro malé i velké sítě	Vhodné pro malé sítě
Bezpečnost	Méně bezpečné	Více bezpečné
Spotřeba systémových zdrojů	CPU, paměť, šířka pásma	Žádné zvláštní nejsou třeba
Předvídatelnost	Cesta je závislá na aktuální topologii	Cesta do cíle je vždy stejná
Výhody Pro (*Pros*)	• Administrátor má méně práce s údržbou při přidání nebo odpojení sítě. • Protokoly automaticky reagují na změny topologie. • Konfigurace je méně náchylná ke vzniku chyby. • Více škálovatelné, zvětšení sítě obvykle nepředstavuje problém.	• Minimální spotřeba CPU • Snadné na pochopení • Snadná konfigurace
Nevýhody Proti (*Cons*)	• Jsou využívány systémové zdroje směrovače (cykly CPU, paměť a šířka pásma linky). • Pro konfiguraci, ověření a hledání chyb jsou potřeba hlubší znalosti na straně administrátora.	• Časově náročná konfigurace i údržba • Snadný vznik chyby při nastavování, zvlášť u velkých sítí • Administrátor musí zasahovat při změně topologie a měnit obsah směrovací tabulky • Obtížně se rozšiřuje, aktualizace může být těžkopádná • Pro implementaci vyžaduje kompletní znalost celé sítě.

Klasifikace směrovacích protokolů

Evoluce směrovacích protokolů:

- EGP – 1982

- IGRP – 1985
- RIPv1 – 1988
- IS-IS – 1990
- OSPFv2 – 1991
- EIGRP – 1992
- RIPv2 – 1994
- BGP – 1995
- RIPng – 1997
- BGPv6 & OSPFv3 – 1999
- IS-ISv6 – 2000

Klasifikace dynamických směrovacích protokolů:

Vnitřní/vnější	Vnitřní (*Interior Gateway Protocol*)				Vnější (*Exterior*)
Algoritmus	Vektor vzdálenosti (*Distance Vector*)		Stav linky (*Link State*)		Vektor cesty (*Path Vector*)
Třídní	**RIPv1**	IGRP	-	-	EGP
Beztřídní	**RIPv2**	**EIGRP**	**OSPFv2**	IS-IS	BGPv4
IPv6	RIPng	EIGRP pro IPv6	OSPFv3	IS-IS pro IPv6	BGPv4 pro IPv6

V tomto kurzu se budeme zabývat pouze **zvýrazněnými** směrovacími protokoly. (IS-IS a BGP bude až v kurzu CCNP.)

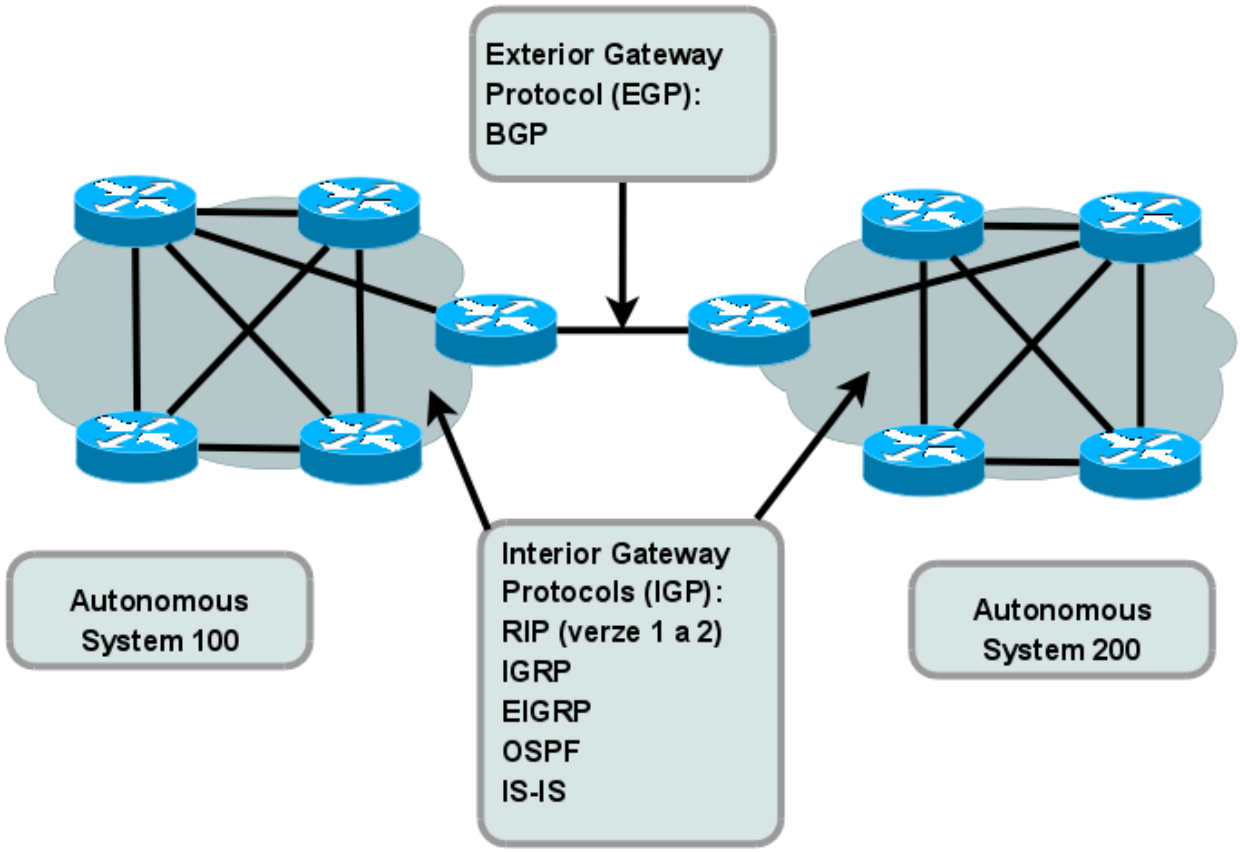
Rozdělení protokolů

Vnitřní a vnější směrovací protokol

- vnitřní směrovací protokoly - (*Interior Gateway Protocols, IGP*) se používají pro směrování uvnitř jednoho autonomního systému[14]
- vnější směrovací protokoly - (*Exterior Gateway Protocols, EGP*) se používají pro směrování mezi různými autonomními systémy (používají algoritmus vektor cesty).

14 Autonomní systém (= směrovací doména, *routing domain*) je oblast s jednou směrovací politikou – jednou správou směrování. (Například jeden ISP, jedna firma.)

Vnitřní versus vnější směrovací protokoly (IGP vs. EGP Routing Protocols)



Směrovací algoritmy u vnitřních směrovacích protokolů

Účel směrovacího algoritmu:

1. vysílání a příjem směrovacích aktualizací,
2. výpočet nejlepší cesty a její umístění do směrovací tabulky,
3. detekce změn topologie a reakce na tyto změny.

U vnitřních směrovacích protokolů se používají dva směrovací algoritmy: vektor vzdálenosti a stav linky.

- **vektor vzdálenosti** (*distance vector*) – vektor vzdálenosti znamená že směry (cesty) jsou inzerovány jako vektory vzdálenosti a směr. Vzdálenost je definována termíny **metrika** (*metric*) (například počet skoků) a **směr** (*direction*) (což jednoduše je následující směrovač nebo odchozí rozhraní tohoto směrovače). Protokoly typu vektor vzdálenosti obvykle používají pro určení nejlepší cesty algoritmus **Bellman-Ford**. Některé protokoly typu vektor vzdálenosti periodicky posílají kompletní směrovací tabulky na všechny připojené sousedy. Ve velkých sítích mohou být takovéto aktualizace enormně velké a budou příčinou významné části síťového provozu. Ačkoliv algoritmus Bellman-Ford umožňuje získat dostatek informací o topologii sítě, algoritmus směrovači neumožňuje znalost přesné topologie sítě.

Směrovač zná pouze informace potřebné pro směrování, které získal od svého souseda. Vyjádřeno analogií: směrovače používají vektor vzdálenosti jako automobilista používá silniční rozcestníky podél své cesty do konečného cíle. Jedinou informací, kterou směrovač ví o vzdálené síti, je vzdálenost neboli metrika do cílové sítě a který směr neboli odchozí rozhraní použít k jejímu dosažení. Nemá mapu topologie sítě.

- **stav linky** (*link state*) – na rozdíl od činnosti protokolů vektor vzdálenosti může směrovač s nastaveným protokolem typu stav linky vytvořit „úplný přehled“ nebo topologii sítě shromažďováním informací ze všech ostatních směrovačů. Jako pokračování naší analogie k rozcestníkům použitým u vektoru vzdálenosti mají směrovací mapy stavu linky kompletní mapu topologie sítě. Rozcestníky potom nejsou potřeba, protože všechny směrovače používají identickou „mapu“ celé sítě. Směrovač s protokolem stavu linky používá informace stavu linky k vytvoření topologické mapy a k výběru nejlepší cesty do všech cílových sítí v celé topologii. V některých směrovacích protokolech vektoru vzdálenosti, směrovače posílají periodické aktualizace svým sousedům. Protokoly stavu linky nepoužívají periodické aktualizace. Potom, co je síť zkonvergovaná, jsou aktualizace stavu linky posílány pouze při změně v topologii.

Kdy je vhodné použít ten který typ vnitřního směrovacího protokolu:

Vektor vzdálenosti	**Stav linky**
• Síť je jednoduchá a plochá (*flat*) a nevyžaduje zvláštní hierarchickou strukturu • Administrátor nemá dostatek znalostí o konfiguraci a odstraňování problémů při směrování proto, aby mohl použít algoritmus stavu linky. • Pokud jsou implementovány zvláštní typy sítí, jako je například topologie s jedním centrálním zařízením (*hub-and-spoke*). • Doba konvergence v síti je dlouhá (= doba synchronizace směrovacích tabulek tak, aby si vzájemně odpovídaly, byly ve vzájemně konzistentním stavu).	• Hierarchický návrh struktury sítě, obvyklý u rozsáhlých sítí. • Administrátor musí mít dobré znalosti jak implementovat směrovací protokol stavu linky. • Rychlá konvergence.

Směrování třídní versus beztřídní

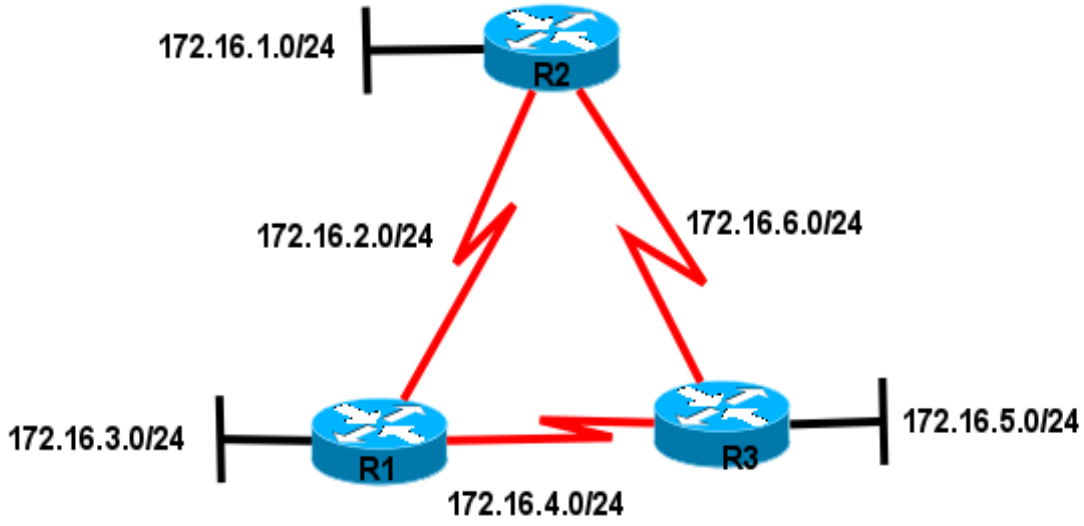
- **třídní** (*classful*) – v celé topologii sítě jsou stejné masky podsítě. První směrovací protokoly byly použité v době, kdy byly sítě pouze v plných třídách (masky byly implicitní určené dle prvního oktetu IP adresy) a nebylo tedy nutné přenášet ve směrovacích aktualizacích masku podsítě. I přesto lze třídní směrování použít i v současnosti: Pokud je podsíťována síť v plné třídě a používá jednu stejnou masku podsítě (= *CIDR, Classless Inter-Domain Routing*). (Beztřídní směrování nepodporuje VLSM.) **Sítě nesmí být nesouvislé** (*discontiguous networks*) musí být souvislé (*contiguous networks*).
    - => **tranzitní síť**[15] může být sice podsíťována (stejnou maskou, tj. adresní schéma

15 Tranzitní síť je síť, která pokračuje na dalším směrovači (je mezi dvěma směrovači). Netranzitní síť (*stub network*)

typu CIDR), ale musí ležet vzhledem k implicitní masce příslušné třídy v jedné síti.

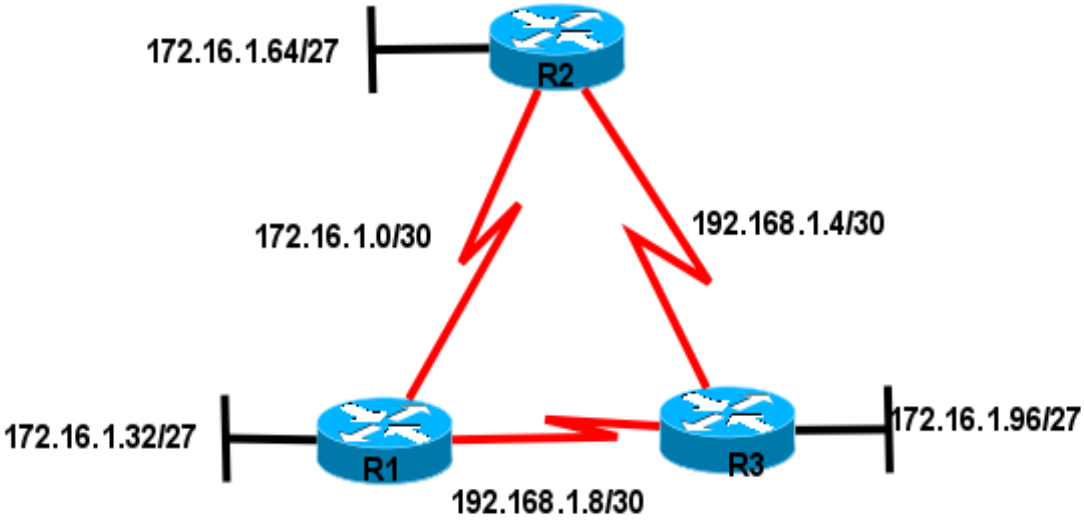
- **beztřídní** (*classless*) – v topologii sítě může být více různých masek podsítě (= podporuje VLSM, *Variable Length Subnet Masking*). Ve směrovacích aktualizacích jsou masky podsítě. Sítě mohou být nesouvislé.

Třídní versus beztřídní směrování (Classful vs. Classless Routing)



Třídní: maska podsítě je stejná v topologii

Classful: Subnet mask is the same throughout the topology



Beztřídní: může být více různých masek v topologii

Classless: Subnet masks can vary in the topology

je přístupná pouze přes jednu cestu (typicky lokální sítě s jedním připojením k ISP) (nebo obsahuje pouze samá koncová zařízení).

Konvergence

Konvergence (*convergency*) – směrovací tabulky na všech směrovačích jsou konzistentní (*state of consistency*). Síť je zkonvergovaná pokud všechny směrovače mají úplné a přesné informace o celé síti (směrovací tabulky jsou vzájemně konzistentní). Doba konvergence je doba, kterou směrovačům zabere výměna informací, výpočet nejlepších cest a aktualizace jejich vlastních směrovacích tabulek. Dokud není síť zkonvergovaná, není síť úplně funkční, proto se vyžaduje, aby doba konvergence byla co nejkratší.

Konvergence je obojí – spolupracující i nezávislá. Směrovače jednak sdílejí informace s každými jinými směrovači, ale zároveň musí samostatně počítat dopady změn topologie na jejich vlastní cesty (*routes*). Charakteristiky konvergence zahrnují: rychlost propagace směrovacích informací a výpočet optimálních cest.

- Pomalejší konvergence – RIP, IGRP
- Rychlejší konvergence – EIGRP, OSPF

Metriky cest

Metrika je číselné vyjádření kvality, ceny (*cost*) cesty. Pokud je více cest do jedné cílové sítě, vybírá se cesta s nejmenší metrikou – nejlevnější cesta.

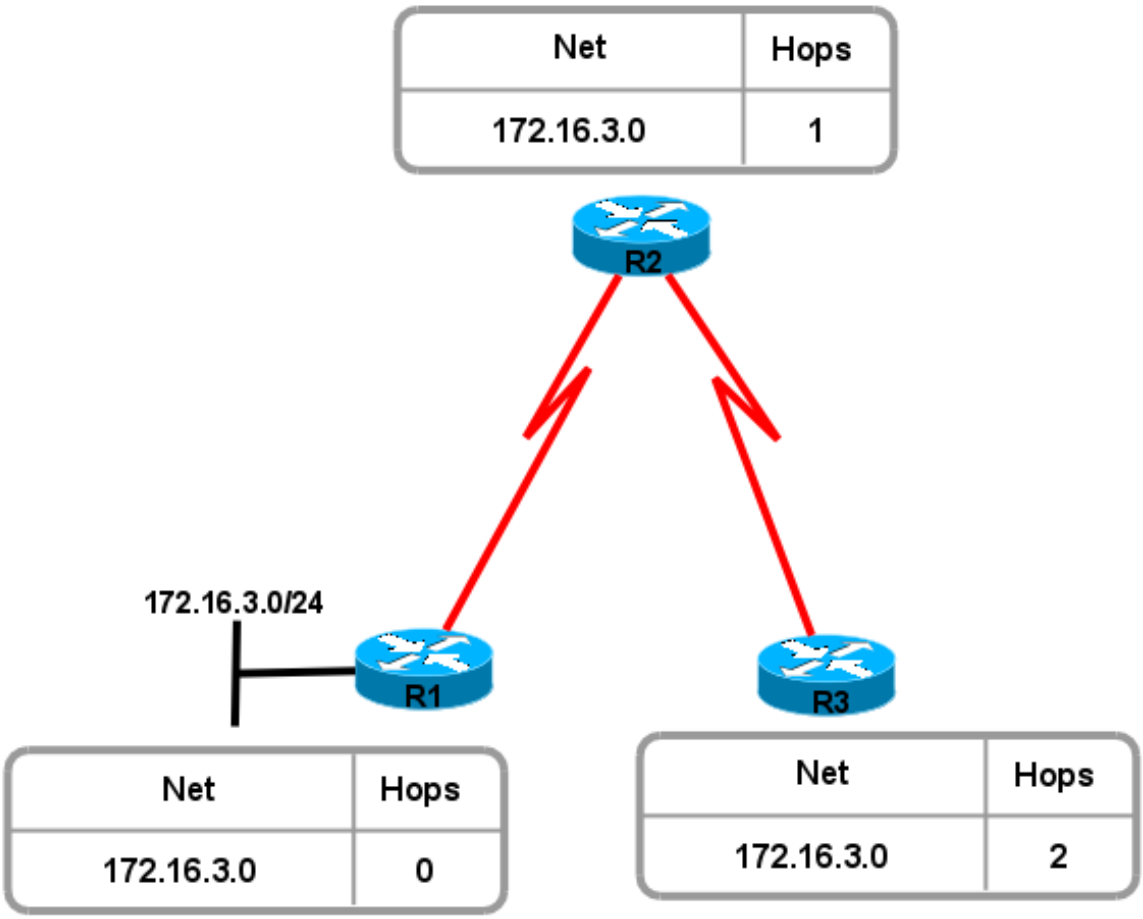
Metriky se u různých směrovacích protokolů počítají různým způsobem:

- **Počet skoků** (*Hop count*) – jednoduchá metrika, která počítá počet směrovačů přes které musí paket cestovat do cílové sítě
- **Digitální přenosová rychlost, přenosová kapacita, šířka pásma** (*Bandwidth*) – při výběru cesty se preferuje linka s větší přenosovou rychlostí
- **Zatížení** (*Load*) – bere v úvahu vytížení dané linky síťovým provozem
- **Zpoždění** (*Delay*) – bere v úvahu dobu, kterou paket potřebuje ke své cestě přes síť
- **Spolehlivost** (*Reliability*) – vyhodnocuje pravděpodobnost výskytu chyby na lince, vypočteno z počtu chyb rozhraní nebo předchozích selhání linky
- **Cena** (*Cost*) – hodnota určená buď IOS nebo administrátorem vyznačující preferování dané cesty.

Metriky každého z následujících směrovacích protokolů jsou:

- **RIP**: počet skoků (*Hop count*) – jako nejlepší cesta je vybrán směr s nejnižším počtem skoků.
- **IGRP a EIGRP**: Přenosová rychlost (přenosová kapacita), zpoždění, spolehlivost a zatížení *(Bandwidth, Delay, Reliability, and Load*) – jako nejlepší cesta je vybrán směr s nejmenší hodnotou složené metriky vypočtené z více různých parametrů. Implicitně je pro výpočet použitá pouze rychlost a zpoždění.
- **IS-IS a OSPF**: Cena (*Cost*) - jako nejlepší cesta je vybrán směr s nejmenší cenou. Implementace OSPF od Cisco používá přenosovou rychlost. IS-IS je diskutován v kurzu CCNP.

Metriky
Hops = přeskoky



Administrativní vzdálenosti protokolů

Administrativní vzdálenost (*Administrative Distance, AD*) slouží k odlišení metriky u cest (na jednom směrovači) **získaných z různých směrovacích protokolů**[16]. AD vyjadřuje kvalitu celého směrovacího protokolu, někdy se také říká důvěryhodnost cesty (*trustwothiness*). Pokud existuje více cest do jedné cílové sítě, vybírá se cesta, která má nejmenší administrativní vzdálenost, pokud je více cest se stejnou administrativní vzdáleností, vybírá se cesta s nejmenší metrikou.

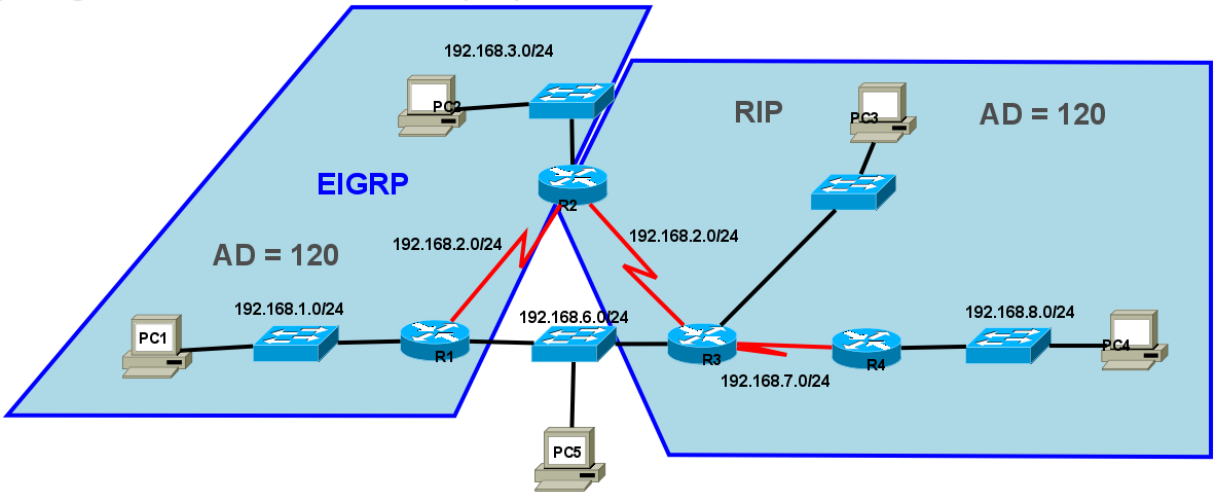
Kódy směrovacích protokolů ve směrovací tabulce směrovače:

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
```

16 Na jednom směrovači může být spuštěno více různých směrovacích protokolů.

```
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route
```

Porovnání administrativních vzdáleností
Comparing Administrative Distances (AD)



R1 a R3 "nemluví" stejnými směrovacími protokolydo not "spe

Administrativní vzdálenosti jednotlivých směrovacích protokolů

Směrovací protokol	***Kód***	***Administrativní vzdálenost Administrative Distance***
Přímo připojená síť (Directly connected)	**C**	**0**
Statická cesta (Static route)	**S**	**1**
EIGRP sumarizovaná cesta (summary route)		5
External BGP (Border Gateway Protocol)		20
Internal EIGRP	**D**	**90**
IGRP	I	100
OSPF (Open Shortest Path First)	**O**	**110**
IS-IS (Intermediate System to Intermediate System Routing Exchange Protocol)	i	115
RIP (Routing Information Protocol)	**R**	**120**
EGP (Exterior Gateway Protocol)	E	140
ODR (On-Demand Routing)		160
External EIGRP		170

Směrovací protokol	***Kód***	***Administrativní vzdálenost Administrative Distance***
Internal BGP		200
Neznámý protokol (Unknown)		**255**

Vybírá se vždy cesta s nejnižší administrativní vzdáleností (např. Vybere se cesta od OSPF než od RIP) a metrikou („cenou“ cesty).

Vyrovnávání zátěže

Pokud do jedné cílové sítě existuje více stejně nákladných cest (= se stejnou administrativní vzdáleností a stejnou metrikou) umí směrovací protokoly vyvažovat zátěž (*load balance*) (= cyklicky přepínat mezi jednotlivými cestami) mezi implicitně až čtyřmi takovýmito cestami. (V EIGRP lze vyvažovat/vyrovnávat zátěž až mezi 6-ti cestami a tyto cesty navíc nemusejí být stejně nákladné.)

Identifikace prvků směrovací tabulky

```
R    192.168.1.0/24 [120/1] via 172.16.3.253, 00:00:14, FastEthernet1/0
```

- směrovací protokol, kterým byla získána řádka
- IP adresa cílové sítě/maska (délka prefixu)
- administrativní vzdálenost/metrika
- IP adresa vstupního rozhraní dalšího směrovače (*next-hop router*)
- stáří aktualizace řádky hod:min:sec
- odchozí rozhraní směrovače pro danou cestu

Kontrolní opakovací otázky a odpovědi (kvíz):

1) Dvě výhody statického směrování proti dynamickému:
   a) je bezpečnější, protože směrovače neinzerují cesty,
   b) není režie (zátěž na směrovači) pro výměnu směrovacích informací.
2) Spárování popisků směrovacích protokolů:
   a) RIP = (*Routing Information Protocol*) interní směrovací protokol typu vektor vzdálenosti
   b) IGRP = (Interior Gateway Routing Protocol) proprietární vnitřní směrovací protokol Cisco
   c) EIGRP = (Enhanced Interior Gateway Routing Protocol) vylepšený proprietární vnitřní směrovací protokol Cisco
   d) OSPF = (Open Shortest Path First) vnitřní směrovací protokol typu stav linky

e) BGP = (Border Gateway Protocol) vnější směrovací protokol typu vektor cesty

3) Tvrzení popisující konvergenci sítě:
   a) čas, který směrovače v síti potřebují pro aktualizaci svých směrovacích tabulek po změně topologie sítě
4) Který směrovací protokol má implicitně nejdůvěryhodnější administrativní vzdálenost? (rozuměj nejmenší hodnotu AD)
   a) EIGRP (AD = 90) (OSPF = 110, RIP = 120, ...)
5) Pro kolik cest se stejnou metrikou mohou implicitně směrovací protokoly vyvažovat jejich zatížení (zátěž) (*load balancing*)?
   a) 4
6) Kterým příkazem můžete vypsat administrativní vzdálenost směru (cesty)?
   a) show ip route (vypíše obsah směrovací tabulky)
7) Kdy se objeví přímo připojené cesty ve směrovací tabulce?
   a) Ihned jakmile jsou nastaveny adresy a jsou funkční na L3 OSI.
8) Směrovač se spuštěným protokolem RIPv2 má více různých cest do cílové sítě. Jak RIPv2 určí nejlepší cestu?
   a) Podle nejmenší metriky
9) Zopakujte si administrativní vzdálenosti pro základní směrovací protokoly.
10) Základní rozdíly mezi třídním a beztřídním směrováním
   a) třídní: neposílají v svých aktualizacích masku podsítě, nepodporují nesouvislé sítě, RIP-v1 a IGRP.
   b) beztřídní: ve svých aktualizacích posílají masku podsítě, podporují nesouvislé sítě, EIGRP, OSPF a BGP.

Kapitola 4 - Směrovací protokoly typu vektor vzdálenosti

V této kapitole se naučíme:

- Identifikovat charakteristiky směrovacích protokolů založených na algoritmu vektor vzdálenosti (*distance vector*)
- Popsat postup průzkumu sítě protokoly vektoru vzdálenosti s použitím protokolu RIP (*Routing Information Protocol*)
- Popsat proces údržby a aktualizace přesného obsahu směrovacích tabulek tak, jak ho provádějí směrovací protokoly založené na vektoru vzdálenosti
- Popsat podmínky vedoucí ke vzniku směrovacích smyček a jejich dopady na výkon směrovače
- Určit typy směrovacích protokolů založené na algoritmu vektoru vzdálenosti, které se v současnosti používají.

Směrovací protokoly typu vektor vzdálenosti

Dynamické směrovací protokoly administrátorovi šetří čas nutný pro časově náročné a přesné konfigurování i údržbu statických cest. Například: dovedete si představit spotřebovaný čas potřebný pro nastavení statického směrování skupiny několika desítek různě vzájemně propojených směrovačů? Co se stane, když nějaká linka spadne? Jak zajistíte, aby byla dostupná náhradní linka? Pro takovéto velké sítě je nejobvyklejší volbou dynamické směrování.

Účel směrovacího algoritmu:

1. vysílání a příjem směrovacích aktualizací,
2. výpočet nejlepší cesty a její umístění do směrovací tabulky,
3. detekce změn topologie a reakce na tyto změny.

Protokoly typu vektor vzdálenosti (se směrovacím algoritmem vektor vzdálenosti) zahrnují tyto směrovací protokoly: RIP, IGRP, a EIGRP.

RIP

Routing Information Protocol (RIP[17]) byl původně specifikován v RFC 1058 (pro verzi 1 - RIPv1). Má následující klíčové charakteristiky:

- Jako metrika pro výběr cest je použit počet skoků (*hop count*).
- Jestliže je počet skoků pro nějakou síť větší než 15, nelze RIP použít pro směrování do takové sítě.
- Směrovací aktualizace jsou implicitně všesměrové (*broadcast*) nebo skupinové (*multicast*) (pro verzi 2) každých 30 sekund.

17 Nejde tedy o zkratku pro „Odpočívej v pokoji“ - *Rest in Peace – latinsky Requiescat in pace*, jak tvrdí zlomyslníci.

IGRP

Interior Gateway Routing Protocol (IGRP) je proprietární protokol vyvinutý firmou Cisco. Má následující klíčové charakteristiky:

- Pro vytvoření kompozitní metriky (*composite metric*) jsou použity: přenosová kapacita (šířka pásma) (*bandwidth*), zpoždění (*delay*), zátěž (*load*) a spolehlivost (*reliability*).
- Směrovací aktualizace jsou implicitně všesměrové (*broadcast*) každých 90 sekund.
- IGRP je předchůdce protokolu EIGRP a je dnes již zastaralý.

EIGRP

Enhanced IGRP (EIGRP) je proprietární protokol vyvinutý firmou Cisco. Má následující klíčové charakteristiky:

- Může provádět vyvažování zátěže cest s různou cenou (*unequal cost load balancing*).
- Pro výpočet nejkratší cesty používá difuzní aktualizační algoritmus DUAL (*Diffusing Update Algorithm)*.
- Nemá periodické aktualizace jako RIP a IGRP. Směrovací aktualizace jsou zasílány pouze při změnách topologie sítě.

Porovnání vlastností směrovacích protokolů

	Vektor vzdálenosti				Stav linky	
	RIPv1	**RIPv2**	**IGRP**	**EIGRP**	OSPF	IS-IS
Rychlost konvergence	**Pomalá**	**Pomalá**	**Pomalá**	**Rychlá**	Rychlá	Rychlá
Rozšiřitelnost – velikost sítě	**Malá**	**Malá**	**Malá**	**Velká**	Velká	Velká
Použití VLSM	**Ne**	**Ano**	**Ne**	**Ano**	Ano	Ano
Využití systémových zdrojů	**Nízké**	**Nízké**	**Nízké**	**Střední**	Vysoké	Vysoké
Implementace a údržba	**Jednoduchá**	**Jednoduchá**	**Jednoduchá**	**Komplikovaná**	Komplikovaná	Komplikovaná

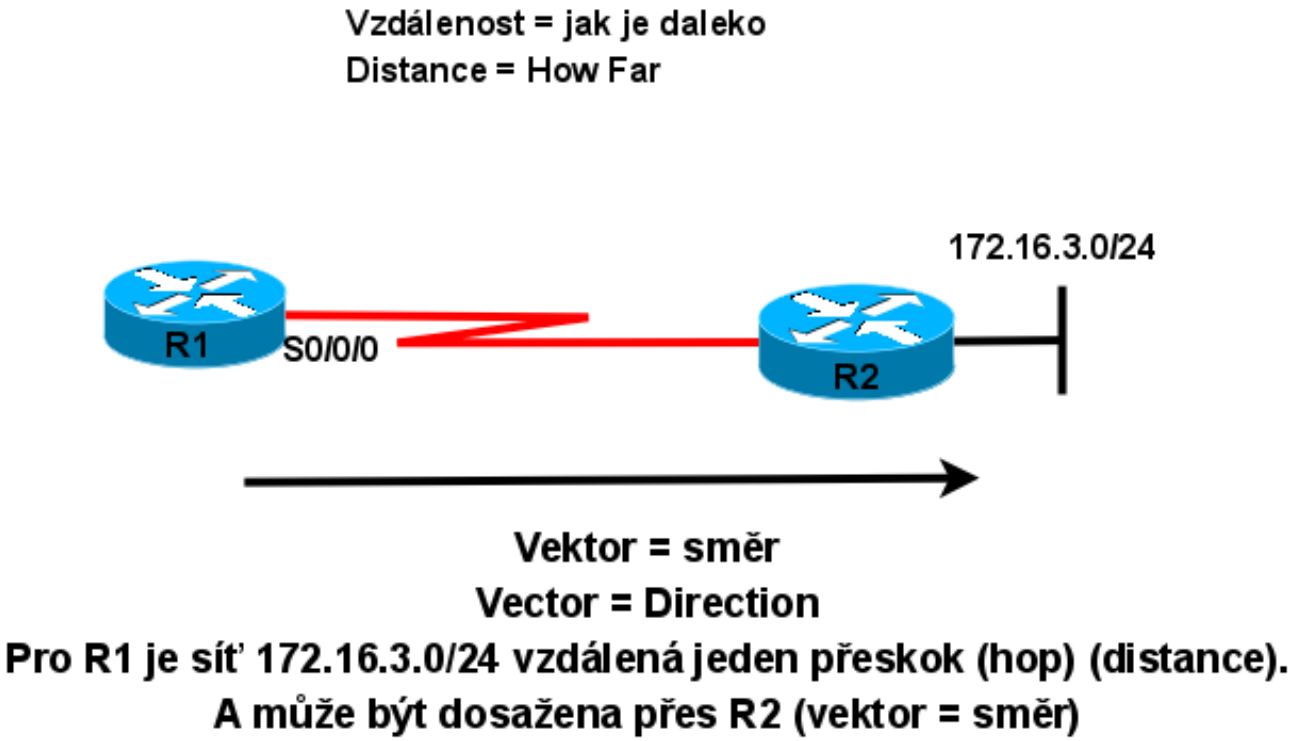
Význam vektoru vzdálenosti

Jak už implikuje samotný název, vektor vzdálenosti znamená, že cesty (směry) jsou inzerovány jako vzdálenost a směr. Vzdálenost je definována termínem metrika (jako je počet skoků) a směr je jednoduše směrovač dalšího skoku nebo odchozí rozhraní.

Směrovač používající vektor vzdálenosti nemá vědomost o celé cestě do cílové sítě. Místo toho směrovač zná pouze:

- **směr** neboli rozhraní, kterým by měl být paket přeposlán a
- **vzdálenost** neboli jak je do cílové sítě daleko.

Význam vektoru vzdálenosti (The Meaning of Distance Vector)



Funkce směrovacích protokolů typu vektor vzdálenosti

Některé směrovací protokoly vektoru vzdálenosti vyžadují, aby směrovač periodicky všesměrově posílal celou směrovací tabulku všem svým sousedům. Tato metoda je neefektivní, protože aktualizace pro svoji činnost konzumují nejen šířku pásma, ale i systémové zdroje směrovače.

Protokoly vektor vzdálenosti mají společné určité charakteristiky:

- **Periodické aktualizace** (*Periodic Updates*) jsou vysílány v pravidelných intervalech (30 sekund u RIP, 90 sekund u IGRP) všem sousedům. Bez ohledu na to, že se topologie nemění třeba několik dnů.
- **Sousedi** (*Neighbors*) jsou směrovače, které sdílejí linku a mají nastavený stejný směrovací protokol. Směrovač je si vědom pouze síťové adresy svého vlastního rozhraní a adresy vzdálené sítě, kterou může dosáhnout přes svého souseda. **Směrovač nemá žádnou vědomost o topologii sítě.**
- **Všesměrové aktualizace** (*Broadcast Updates*) jsou vysílány na adresu 255.255.255.255.

Sousedící směrovače s nastaveným stejným směrovacím protokolem aktualizaci provedou. Všechna ostatní zařízení aktualizaci také zpracují až do L3 před tím, než ji zahodí. Některé protokoly vektoru vzdálenosti používají skupinové adresy místo všesměrových adres.

- **Aktualizace obsahují celé směrovací tabulky** (kromě výjimek, které budou diskutovány později) a jsou vysílány periodicky na všechny sousedy. Sousedi musí zpracovat celou aktualizaci, aby našli patřičnou informaci a zahodili zbytek. Některé směrovací protokoly (jako EIGRP) neposílají periodické aktualizace.

Účel algoritmu směrovacího protokolu

Jádrem protokolu typu vektor vzdálenosti je algoritmus. Algoritmus je použit pro výpočet nejlepší cesty a poté co je tato informace vyslána sousedům.

Algoritmus je postup pro dosažení určitého konkrétního cíle (úkolu), začíná v daném počátečním stavu a je ukončen v definovaném koncovém stavu. Různé směrovací protokoly používají různé algoritmy pro instalaci směrů do směrovacích tabulek, vysílání aktualizací sousedům a určení cesty.

Algoritmus použitý pro směrovací protokoly definuje následující procesy:

- mechanizmus pro vysílání a příjem směrovacích informací,
- mechanizmus pro výpočet nejlepší cesty a instalaci tohoto směru (cesty) do směrovací tabulky,
- mechanizmus pro detekci změn topologie a reakce na tyto změny.

Charakteristiky směrovacích protokolů

Směrovací protokoly mohou být porovnávány na základě následujících charakteristik:

- **Doba potřebná pro konvergenci** (*Time to Convergence*) – definuje jak rychle směrovače v síti sdílejí směrovací informace a dosáhnou stavu konzistence znalostí. Čím rychlejší konvergence, tím preferovanější směrovací protokol. Když nejsou (z důvodu pomalé konvergence v měnící se síti) aktualizované nekonzistentní směrovací tabulky, mohou se vyskytnou **směrovací smyčky**.
- **Rozšiřitelnost** (škálovatelnost, *Scalability*) – definuje pro jak velkou síť může být ten který směrovací protokol použit. Čím větší ta síť je, tím více rozšiřitelný směrovací protokol musí být.
- **Beztřídnost** (*Classless*) (použití VLSM) **nebo plnotřídnost** (*Classful*) – Beztřídní směrovací protokoly ve svých aktualizacích obsahují masku podsítě. Tato funkce podporuje použití VLSM (*Variable Length Subnet Masking*) a lepší sumarizaci směru. Třídní směrovací protokoly nemají v aktualizaci směrovací masku a nepodporují VLSM.
- **Spotřeba zdrojů** (*Resource Usage*) – zahrnuje požadavky směrovacího protokolu na systémové zdroje jako jsou velikost operační paměti, využití procesoru, využití šířky pásma linky. Vyšší požadavky na systémové zdroje si vynucují použití silnějšího HW.
- **Implementace a údržba** (*Implementation and Maintenance*) – požadavky na znalosti administrátora potřebné pro implementaci a údržbu použitého směrovacího protokolu.

Výhody a nevýhody směrovacích protokolů typu vektor vzdálenosti jsou uvedeny v následující

tabulce:

Výhody *směrovacích protokolů typu vektor vzdálenosti*	***Nevýhody*** *směrovacích protokolů typu vektor vzdálenosti*
Jednoduchá implementace a údržba – administrátor nepotřebuje příliš hluboké znalosti.	**Pomalá konvergence** – použití periodických aktualizací způsobuje pomalou konvergenci. I když jsou použité pokročilé metody jako je automaticky spouštěná aktualizace, jsou tyto protokoly pomalejší než protokoly typu stav linky.
Nízké požadavky na zdroje – nevyžaduje výkonnou CPU a velkou paměť. Požadavky však vzrůstají se zvětšující se sítí.	**Malá rozšiřitelnost** – pomalá konvergence ve velkých sítích způsobuje dlouhou nekonzistentnost směrovacích tabulek.
	Směrovací smyčky – ty se mohou objevit, když nejsou nekonzistentní směrovací tabulky aktualizovány kvůli pomalé konvergenci.

Objevování sítí

Studený start

Když se směrovač zapne (*cold start*), neví nic o síťové topologii ani neví nic o tom, že je na druhé straně linky nějaké síťové zařízení. Má pouze informace uložené v počáteční konfiguraci uložené v NVRAM. Jakmile směrovač úspěšně zavede operační systém použije uloženou konfiguraci. Jestliže jsou korektně nastaveny IP adresy, směrovač detekuje svoje vlastní přímo připojené sítě a příslušné masky. Tyto informace jsou přidány do jeho směrovací tabulky. To vše proběhne ještě před jakoukoliv výměnou směrovacích informací pomocí dynamických směrovacích protokolů.

Počáteční výměna směrovacích informací

Jestliže je nakonfigurován směrovací protokol, směrovače si začnou vyměňovat směrovací aktualizace. Na počátku tyto aktualizace obsahují pouze informace o jejich přímo připojených sítích. Potom co přijme aktualizaci v směrovač vyhledá nové informace. Všechny směry do sítí, které aktuálně nejsou ve směrovací tabulce jsou do ní přidány.

Další aktualizace

V té chvíli mají směrovače vědomost o svých vlastních přímo připojených sítích a o připojených sítích jejich bezprostředních sousedů. Pokračujíce v konvergenci si směrovače vymění další kolo (*next round*) periodických aktualizací. Každý směrovač opět ověří aktualizace a vyhledá nové informace a nové směry do konkrétních sítí přidá do směrovací tabulky.

Konvergence

Doba konvergence je přímo úměrná velikosti sítě. Rychlost dosažení zkonvergované sítě závisí na:

- rychlosti propagace změn topologie v aktualizacích pro sousedy daného směrovače,

- rychlosti výpočtu nejlepších směrů (cest) na základě získaných informací z aktualizací.

Síť není plně funkční dokud není zkonvergovaná, proto administrátoři preferují směrovací protokoly s krátkou dobou konvergence.

Údržba směrovací tabulky

Mnoho protokolů typu vektor vzdálenosti používá periodické aktualizace (příkladem je RIP a IGRP).

Například RIP zasílá aktualizace každých 30 sekund jako broadcast (255.255.255.255) bez ohledu na to, zda došlo či nedošlo ke změně topologie. Tento interval je **aktualizační časovač** (*Update Timer*), který také slouží k sledování stáří informací ve směrovací tabulce. Nulové stáří směrovacích informací je obnoveno po každém přijetí aktualizace. Tento způsob sledování je možné využít, když dojde ke změně topologie. Ke změně topologie může dojít z následujících důvodů:

- selhání linky,
- připojení nové linky,
- selhání směrovače,
- změna parametrů linky.

Časovače u protokolu RIP

K **aktualizačnímu časovači** (*Update Timer*) = perioda pravidelných aktualizací (30 sekund), zavádí ještě IOS další tři časovače:

- **časovač neplatné cesty** (*Invalid Timer*) – pokud nepřijde aktualizace existující cesty do 180 sekund (implicitně), je cesta označena jako neplatná nastavením nekonečné metriky (na 16). Cesta zůstává jako neplatná ve směrovací tabulce dokud nevyprší **vyprazdňovací časovač**.
- **vyprazdňovací časovač** (*Flush Timer*) – implicitně je nastavený na 240 sekund, o 60 sekund delší než časovač neplatné cesty. Když tento časovač vyprší, je cesta smazána ze směrovací tabulky.
- **zadržovací časovač** (*Holddown Timer*) – tento časovač stabilizuje směrovací informace a **předchází vzniku směrovacích smyček** (*routing loop*). Během doby, kdy topologie konverguje, neinzeruje nové informace vzniklé po změně topologie. Jakmile je cesta (*route*, směr) označena jako nedostupná (*unreachable*), musí zůstat zadržená (*holddown*) dostatečně dlouho, aby se všechny ostatní směrovače byly schopné dozvědět o nedostupné síti. Zadržovací časovač musí být nastaven o chvíli (o několik sekund) delší než je celková doba konvergence celé sítě (implicitní hodnota je 180 sekund).

Na směrování je třeba pohlížet jako na dynamický systém měnící se v čase. Smyslem časovačů je tlumení tohoto dynamického systému. Vždy je lepší mít směr do cílové sítě označený jako neplatný, než nevědět vůbec nic a směrovat přes implicitní cestu, pokud je nastavena.

Periodické aktualizace: RIPv1, IGRP

viz časovače u protokolu RIP

- zjištění stáří záznamu ve směrovací tabulce:
  - `show ip route`
- zjištění spuštěných směrovacích protokolů, jejich konfigurace a poslední aktualizace:
  - `show ip protocols`

Svázané aktualizace: EIGRP

Na rozdíl od ostatních protokolů vektoru vzdálenosti, EIGRP nevysílá periodické aktualizace. Místo toho EIGRP vysílá svázané aktualizace (*bounded update*) o směru ve kterém se změnila cesta nebo metrika. Když se objeví nový směr nebo když má být směr smazán, EIGRP posílá aktualizaci pouze o této síti a nikoliv celou směrovací tabulku. Tato informace je vyslána pouze těm směrovačům, které ji potřebují.

Aktualizace EIGRP tedy jsou:

- **neperiodické**, protože nejsou vysílány pravidelně,
- **částečné** (*partial update*), protože se posílají informace pouze o změnách topologie, které mají vliv na směrování,
- **svázané** (*bounded*), protože se propagují částečné aktualizace automaticky svázané s určitými směrovači (s vytvořeným vztahem sousedství), takže jsou aktualizovány pouze ty směrovače, které ty informace potřebují.

Poznámka: podrobněji budou funkce EIGRP zmíněny v kapitole 9.

Událostí spouštěné aktualizace: RIPv1 i RIPv2

Aby se urychlila konvergence po změně topologie, používá RIP událostí vyvolané **spouštěné aktualizace** (*triggered updates*). Tyto aktualizace jsou vyslány bezprostředně po změně směrovací tabulky a nečekají na vypršení aktualizačního směrovače. Směrovač detekující změnu ihned vyšle aktualizační zprávu na své sousedící směrovače (*adjacent routers*). Přijímající směrovače postupně generují spouštěné aktualizace, které informují zase jejich sousedy o této změně.

Spouštěné aktualizace se vysílají po výskytu jedné z následujících událostí:

- změna stavu rozhraní (zapnuto nebo vypnuto)
- směrovač vstoupil (nebo vystoupil) z/do „nedosažitelného“ stavu (*"unreachable" state*)
- do směrovací tabulky byl instalován (nový) směr (cesta)

Použití pouze spouštěných aktualizací by mohlo být dostačující, pokud by zde byla záruka, že vlna aktualizací dosáhne každý příslušný směrovač ihned. Přesto jsou se spouštěnými aktualizacemi dva problémy:

1. Pakety obsahující aktualizace mohou být na některých linkách v síti zahozeny nebo poškozeny.
2. Spouštěné aktualizace nejsou okamžité. Je tedy možné, že směrovač, který ještě nepřijal spouštěnou aktualizaci vyšle pravidelnou periodickou aktualizaci právě ve špatný čas, což způsobí, že na souseda, který právě přijímá spouštěnou aktualizaci bude znovu vložen v této

chvíli už špatný směr.

Náhodné kolísání (Random Jitter) aktualizačního časovače

Problémy se synchronizovanými aktualizacemi

Když více směrovačů přenáší směrovací aktualizace najednou ve stejnou chvíli (například v segmentu sítě LAN s vícenásobným přístupem a rozbočovačem), mohou pakety kolidovat a způsobit zpoždění nebo spotřebovávat příliš velkou šířku pásma. (Poznámka: kolize se vyskytnou pouze v případě použití rozbočovače a nikoliv přepínače.)

Vysílání aktualizací ve stejný čas je známé jako **synchronizace aktualizací**. Synchronizace může způsobovat problémy spolu s protokoly typu vektor vzdálenosti, protože ty používají periodické aktualizace. Jak začne být několik směrovačů synchronizováno, objeví se v síti více a více kolizí aktualizací a prodlužující se zpoždění. Na počátku aktualizace nebudou synchronizovány, ale postupně se časovače v síti sesynchronizují.

Řešení

Aby se zabránilo synchronizaci aktualizací mezi směrovači, Cisco IOS používá náhodnou proměnnou nazývanou RIP_JITTER, která odečítá určitý časový úsek od aktualizačního intervalu na každém směrovači v síti. Toto náhodné kolísání (*random jitter*) se mění mezi 0 až 15 procenty doby periody aktualizací. Tímto způsobem potom implicitní 30-ti sekundový aktualizační časovač náhodně kolísá mezi 25 až 30 sekundami.

Směrovací smyčka

Směrovací smyčka (*Routing Loop*) je stav ve kterém je paket trvale přenášen uvnitř skupiny na sebe navazujících směrovačů aniž kdy dosáhne zamýšlené cílové sítě. Směrovací smyčka se může vyskytnout když dva nebo více směrovačů mají ve své směrovací tabulce zdánlivě platnou cestu do ve skutečnosti nedosažitelného cíle. Důsledkem je nedoručování některých paketů a zbytečná zátěž sítě.

Čtyři možné způsoby vzniku směrovací smyčky:

1. nesprávně nastavené statické cesty
2. nesprávně nastavený proces redistribuce cest (na hraničních směrovačích mezi různými směrovacími protokoly) (podrobněji se probírá až v kurzu CCNP)
3. nekonzistentní směrovací tabulky při pomalé konvergenci v měnící se síti
4. nesprávně nastavené nebo nainstalované vyřazené cesty (po výpadku linky)

Pět mechanismů pro eliminaci směrovacích smyček:

1. definování maximální metriky pro eliminaci počítání do nekonečna (*count to infinity*)
2. zadržovací časovače (*holddown timers*)
3. rozložený horizont (*split horizon rule*)
4. otrávení/znehodnocení cest (*route poisoning*) nebo otrávené/znehodnocené zpětné informace

(*poison reverse*)

5. aktualizace vyvolané událostí (*triggerd updates*)

Rozložený horizont

Pravidlo **rozložený horizont** (*split horizon*) předchází vzniku směrovacích smyček tak, že směrovač neinzeruje síť prostřednictvím toho rozhraní, ze kterého se o této síti dozvěděl (ze kterého přišla původní aktualizace).

Rozložený horizont s otrávenou zpětnou informací neboli otrávení cest

Otrávení/znehodnocení cest

Otrávení/znehodnocení cest (*route poisoning*) slouží k označení cesty jako nedostupné ve směrovací aktualizaci, která je zasílána na jiné směrovače. Jako nedostupná je interpretována cesta, která má nastavenu metriku větší než maximální možnou (= nekonečnou). Pro RIP je otrávena cesta s metrikou 16.

Rozložený horizont s otrávenou/znehodnocenou zpětnou informací

Pravidlo **rozložený horizont s otrávenou/znehodnocenou zpětnou informací** (*Split Horizon with Poison Reverse*) označuje, když vysílá směrovací aktualizace z určitého rozhraní, všechny sítě, které byly naučeny z tohoto rozhraní, jako nedostupné. Obecně je lepší říci směrovači, že určité cesty (směry) jsou nedostupné a že je má ignorovat, než mu o nich neříci vůbec nic (a směrovač se potom může snažit použít implicitní cestu).

Životnost IP paketu

Životnost IP paketu (*Time to Live (TTL)*) je 8-bitové pole v záhlaví IP, které omezuje počet skoků (směrovačů), přes které může paket cestovat přes síť před tím, než je zahozen. Účelem pole TTL je předcházet situaci, kdy by nedoručitelný paket cirkuloval v síti nekonečně dlouho. Hodnota TTL je v paketu nastavena zdrojovým zařízením paketu. **Na každém směrovači na cestě do cíle je TTL snižováno o jedničku.** Pokud TTL dosáhne na nějakém směrovači hodnotu nula ještě před svým doručením do cíle, paket je zahozen a tento směrovač zašle zpět zdrojovému zařízení chybovou zprávu v protokolu ICMP (*Internet Control Message Protocol*) o překročení životnosti (*TTL exceeded message*).

Porovnání směrovacích protokolů (typu vektor vzdálenosti):

	RIPv1	RIPv2	IGRP	EIGRP
Vektor vzdálenosti (*Distance Vector*)	Yes	Yes	Yes	Yes
VLSM	No	Yes	No	Yes
Autentizace zdroje aktualizací (*Authentication*)	No	Yes	No	Yes

Aktualizační časovač (*Update Timer*) (sec)	30	30	90	n/a
Časovač neplatné cesty (*Invalid Timer*) (sec)	180	180	270	n/a
Vyprazdňovací časovač (*Flush Timer*) (sec)	240	240	630	n/a
Zadržovací časovač (*Holddown Timer*) (sec)	180	180	280	n/a
Protokol/port (*Protocol/port*)	UDP 520	UDP 520	IP 9	IP 88
Administrativní vzdálenost (*Administrative Distance*)	120	120	100	90

Výpočet metriky u algoritmu typu vektor vzdálenosti

Pro jednu konkrétní cestu se vezme metrika inzerovaná v aktualizaci ze sousedního směrovače a k ní se **přičte** metrika cesty z rozhraní, na které aktualizace přišla, do směrovače, ze kterého aktualizace přišla. To znamená, že k metrice cesty do cílové sítě propagované ze sousedního směrovače se přičte metrika přilehlého segmentu fyzické sítě (linky), ze kterého tato propagace přišla. Tento algoritmus se také nazývá **Bellman-Ford**.

Kontrolní opakovací otázky a odpovědi (kvíz):

1) Uveďte čtyři pravdivá tvrzení o směrovacích protokolech typu vektor vzdálenosti:
   a) jako metriku mohou používat počet skoků (RIP),
   b) aktualizace rozesílají v pravidelných časových intervalech všesměrově (RIP),
   c) EIGRP umí vyvažování zátěže i na různě nákladných cestách,
   d) směrovač s RIP zasílá celou svoji směrovací tabulku na všechny sousední směrovače.
2) Za jakých podmínek směrovací protokoly typu vektor vzdálenosti rozesílají svoje směrovací aktualizace? (uveďte tři)
   a) když se objeví změna topologie (volitelně nastavitelné automaticky spouštěné aktualizace, *triggered update*),
   b) když vyprší aktualizační časovač (periodické aktualizace),
   c) když přijde automaticky spuštěná aktualizace z jiného směrovače.
3) Dvě charakteristiky aktualizací z EIGRP:
   a) pouze automaticky spouštěné aktualizace (*triggered update*) při změně topologie (EIGRP nemá pravidelné aktualizace),
   b) svázané aktualizace (*bounded update*) se zasaženými směrovači dalšího skoku (*next hop*)
4) Jaká funkce byla přidána do protokolu RIP, aby předcházela vzniku chyb při synchronizaci aktualizací (to znamená aktualizací probíhajících ve stejných časech paralelně)?
   a) RIP_JITTER (aktualizační časovače mají vloženou náhodnou časovou odchylku 0-15%,

aby se předešlo synchronizaci aktualizací))

5) Časovače použité u RIP:
   a) invalid, flush, holddown a update (naopak není použit časovač *hello*, ten je u EIGRP a OSPF)
6) Co je pravda ohledně výhod použití protokolů typu vektor vzdálenosti:
   a) jednoduchá implementace a snadná konfigurace
7) Jaký mechanismus lze použít pro předcházení vzniku směrovací smyčky typu počítání do nekonečna (*count to infinity loop*)?
   a) Zadržovací časovač.
8) Jak se jmenuje postup, kdy směrovací protokol předchází vzniku smyčky pomocí propagace směru s nastavenou nekonečnou metrikou?
   a) Otravováním cest
9) Které políčko v záhlaví paketu IP zabrání, aby paket necestoval v síti nekonečně dlouho?
   a) TTL
10) Spárování názvu a popisu metod pro předcházení vzniku směrovacích smyček:
   a) rozložený horizont = cesty zjištěné (naučené) z jednoho rozhraní nejsou z tohoto rozhraní propagovány
   b) otrávení cest (otrávení zpětných informací) = cesty naučené z jednoho rozhraní jsou z tohoto rozhraní inzerovány zpět jako nedosažitelné (neplatné, s nekonečnou metrikou)
   c) automaticky spouštěné aktualizace = změny topologie jsou ihned zasílány sousedícím směrovačům,
   d) zadržovací časovače = umožní časovou prodlevu, aby se informace o změně topologie mohly rozšířit po celé síti.

Kapitola 5 - Protokol RIP verze 1

V této kapitole se naučíme:

- Popsat funkce, charakteristiky a činnost protokolu RIPv1.
- Konfigurovat síťové zařízení s použitím RIPv1
- Ověřit správnou činnost RIPv1
- Popsat, jak RIPv1 provádí automatickou sumarizaci
- Konfigurovat, ověřit a odstranit chyby propagace implicitní cesty v sítích směrovaných protokolem RIPv1
- Používat k řešení problémů souvisejících s RIPv1 doporučované techniky

RIPv1: třídní směrovací protokol typu vektor vzdálenost

Směrovací protokoly byly postupem času vyvinuty tak, aby uspokojily rostoucí požadavky po složitých sítích. První takový použitý směrovací protokol byl **Routing Information Protocol (RIP)**. RIP stále těší popularitě díky své jednoduchosti a široké podpoře.

Pochopení protokolu RIP je pro vaše studia sítí důležité z následujících dvou důvodů:

1. RIP je dnes stále ještě používaný. Můžete se setkat s implementací sítě, která je dostatečně velká, aby byl potřeba nějaký směrovací protokol, ale zároveň dostatečně jednoduchá, aby bylo použití RIP efektivní.
2. Obeznámenost se základními koncepty RIP vám pomůže porovnat RIP s jinými protokoly. Pochopení činnosti RIP a jeho implementace umožní snazší pochopení jiných směrovacích protokolů.

Tato kapitola se vztahuje na detaily první verze RIP, včetně trošky historie, vlastností, provozu, konfigurace, ověřování a řešení problémů. V průběhu této kapitoly, můžete používat aktivity Packet Traceru pro procvičení toho, co jste se naučili. Na konci této kapitoly jsou tři praktická laboratorní cvičení a aktivita Packet Traceru pro integraci dovedností, aby vám pomohly zařadit RIPv1 do rostoucí množiny vašich síťových znalostí a dovedností.

RIP vliv minulosti

RIP je nejstarší ze směrovacích protokolů typu vektor vzdálenosti. Přestože RIP postrádá propracovanost pokročilejších směrovacích protokolů jeho jednoduchost a pokračující široké využití je důkazem jeho životnosti. RIP není protokol "na odpis". Ve skutečnosti je nyní k dispozici forma RIP pro IPv6 tzv. RIPng (*Next Generation* = další generace).

Přehled historických souvislostí RIP

Vývoj síťových protokolů			Vývoj RIP
Počátek 70-tých let	Ranný vývoj TCP/IP		
Střed 70-tých let		Xerox PARC Universal Protocol (PUP)	Gateway Information Protocol (GWINFO)
Konec 70-tých let		Xerox Network System (XNS)	Routing Information Protocol
Počátek 80-tých let	Standardizován TCP/IP RFC 791, RFC 793	Berkeley Software Distribution (UNIX BSD 4.2)	Směrovací démon - Routed Daemon („route-dee“)
1988			RFC 1058: RIP
1994			RFC 1723: RIPv2
1997			RFC 2080: RIPng

RIP se vyvinul z dřívějšího protokolu vyvinutého ve firmě Xerox, tzv. Gateway Information Protocol (GWINFO). S rozvojem společnosti Xerox Network System (XNS), se GWINFO vyvinul v RIP. Později získal popularitu protože byl implementován v distribuci Berkeley Software Distribution (BSD) jako démon *routed* (anglicky se vyslovuje "route-dee", nikoli "rout-ed"). Množství dalších prodejců vytvořilo své vlastní mírně odlišné implementace RIP. Cítíc potřebu standardizace protokolu, napsal Charles Hedrick v roce 1988 RFC 1058, v němž dokumentoval stávající protokol a specifikoval některá vylepšení. Od té doby by RIP vylepšen s RIPv2 v roce 1994 a s RIPng v roce 1997.

Poznámka: První verze RIP je často nazývána RIPv1 pro odlišení od RIPv2. Nicméně, obě verze mají mnoho stejných vlastností. Při diskusi vlastností společných pro obě verze budeme odkazovat na RIP. Při diskusi jedinečných vlastností pro každou verzi budeme používat RIPv1 a RIPv2. RIPv2 je diskutován v pozdější kapitole.

Odkazy
"RFC 1058: Routing Information Protocol," http://www.ietf.org/rfc/rfc1058.txt

Charakteristiky RIP

Jak je popsáno v kapitole 4, "Směrovací protokoly typu vektor vzdálenosti", má RIP tyto hlavní vlastnosti:

- RIP je směrovací protokol typu vektor vzdálenosti.
- RIP používá jako jeho jedinou metriku pro výběr cesty počet přeskoků.
- Inzerované trasy s počty skoků většími než 15 jsou nedosažitelné.
- Zprávy jsou všesměrově vysílány každých 30 sekund.

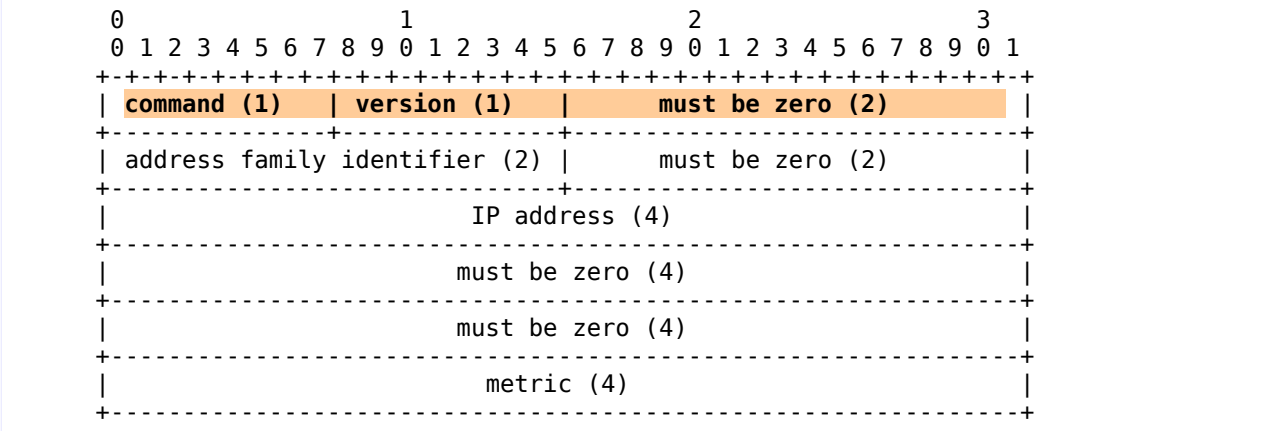
Datová část zprávy RIP je zapouzdřena do segmentu UDP, se zdrojovým i cílovým číslem portu nastaveným na 520. Než je zpráva rozeslána ze všech nakonfigurovaných rozhraní RIP jsou do záhlaví IP a linkové přidány všesměrové cílové adresy.

Zapouzdření zprávy protokolu RIPv1

<table>
<tr><td>Záhlaví linkové vrstvy</td><td>Záhlaví paketu IP</td><td>Záhlaví UDP Datagramu</td><td>Zpráva RIP (512 bajtů: až 25 tras)</td></tr>
<tr><td colspan="4">Rámec linkové vrstvy
Zdrojová MAC adresa = adresa vysílajícího rozhraní
Cílová MAC adresa = Broadcast: FF-FF-FF-FF-FF-FF</td></tr>
<tr><td></td><td colspan="3">Paket IP
Zdrojová IP adresa = adresa vysílajícího rozhraní
Cílová IP adresa = Broadcast: 255.255.255.255
Protokol = 17 pro UDP</td></tr>
<tr><td colspan="2"></td><td colspan="2">Segment (datagram) UDP
Zdrojový port = 520
Cílový port = 520</td></tr>
<tr><td colspan="3"></td><td>Zpráva RIP
Command: Request (1); Response (2)
Version = 1
Address Family ID (AFI) = 2 pro IP
Trasy: IP adresa sítě
Metrika: počet přeskoků</td></tr>
</table>

Formát zprávy RIP: Záhlaví RIP

(V závorkách je u názvu pole uvedena jeho velikost v bajtech.)



Ve čtyřbajtovém záhlaví (na předchozím obrázku zvýrazněném oranžově) jsou 3 pole:

- Pole Příkaz (*Command*) určuje typ zprávy, podrobněji se tím budeme zabývat v následující části.
- Pole Verze (*Version*) je nastaveno na 1 pro RIP verze 1.
- Třetí označené pole musí být nulové. Pole "Musí být nula" (*Must be zero*) poskytuje prostor pro budoucí rozšíření protokolu.

Formát zprávy RIP: Vstup trasy

Část zprávy se vstupem trasy se skládá ze tří polí s obsahem:

- Identifikátor skupiny adres (Address Family ID, AFI) (nastaven na 2 pro IP; pokud požaduje úplné směrovací tabulky, je v takovém případě pole nastaveno na nulu),
- IP adresa sítě,
- Metrika.

Tato část Vstup trasy představuje jednu trasu do cíle a s ní spojenou metriku. Jedna aktualizace RIP

může obsahovat až 25 řádek tras. Maximální velikost datagramu je 504 bajtů, bez jednotlivých záhlaví IP nebo UDP.
Proč je zde tolik polí nastavených na nulu? RIP byla vyvinut ještě před protokolem IP a byl používán pro jiné síťové protokoly (jako XNS). Distribuce BSD měla také svůj vliv. Na počátku bylo do zprávy přidáno další volné místo s cílem podpořit větší adresové prostory v budoucnosti. Jak uvidíme v kapitole 7, RIPv2 nyní používá většinu z těchto prázdných polí.

Provoz RIP

Zpracování Poptávky/Odpovědi RIP

RIP používá dva typy zpráv uvedených v poli Příkaz (*Command*):

- Žádost (*Request*)
- Odpověď (*Response*).

Každé rozhraní s nastaveným RIP vysílá při startu zprávu typu Žádost požadující, aby všechny RIP sousedé zaslali své kompletní směrovací tabulky. Zpráva typu Odpověď je odeslána zpět sousedy se spuštěným protokolem RIP. Když žádající směrovač přijme odpovědi, vyhodnotí každou řádku s trasou. Je-li trasa nová, přijímající směrovač trasu přidá do směrovací tabulky. Je-li trasa již ve směrovací tabulce, je existující řádka nahrazena pouze pokud má nový vstup lepší (to jest menší) počet přeskoků (*hop count*). Směrovač po svém spuštění pošle automaticky spouštěnou aktualizaci, obsahující jeho vlastní směrovací tabulku, ze všech rozhraní s povoleným protokolem RIP, takže sousedi (s běžícím RIP) mohou být informováni o všech nových trasách.

Třídy IP adresy a třídní směrování

Možná si vzpomínáte z předchozích studií, že IP adresy přidělené počítačům byly původně rozděleny do 3 tříd: třídy A, třídy B a třídy C. Každá třída měla přidělenu implicitní masku podsítě, jak je uvedeno na obrázku. Znát výchozí masku podsítě pro každou třídu je důležité pro pochopení toho, jak funguje RIP.

RIP je (plno)třídní směrovací protokol. Jak jste možná pochopili z předchozí diskuse formátu zprávy, RIPv1 neposílá v aktualizaci masku podsítě. Proto směrovač použije buď masku podsítě nastavenou na lokálním rozhraní nebo použije výchozí masku podsítě na základě třídy adresy. Kvůli tomuto omezení nesmí být sítě RIPv1 nesouvislé ani nemohou implementovat VLSM.

IP adresace je dále probírána v kapitole 6, "VLSM a CIDR." Můžete také navštívit níže uvedené odkazy pro zopakování tříd sítí a tvorbu podsítí.

Odkazy:
"Internet Protocol," http://www.ietf.org/rfc/rfc791.txt
"IP adresace a tvorba podsítí pro nové uživatele" http://www.cisco.com/en/US/tech/tk365/technologies_tech_note09186a00800a67f5.shtml

Administrativní vzdálenost

Jak víte z kapitoly 3, "Úvod do dynamických směrovacích protokolů," administrativní vzdálenost (AD) je důvěryhodnost (neboli preference) zdroje trasy. RIP má standardní administrativní vzdálenost 120. Ve srovnání s jinými vnitřními směrovacími protokoly je RIP nejméně preferovaný smě-

rovací protokol. Protokoly IS-IS, OSPF, IGRP, EIGRP mají všechny nižší implicitní hodnoty AD (než RIP).

Pamatujte si, že administrativní vzdálenosti můžete zkontrolovat pomocí příkazů "show ip route" nebo "show ip protocols".

```
R2#sh ip protocols
Routing Protocol is "rip"
Sending updates every 30 seconds, next due in 10 seconds
Invalid after 180 seconds, hold down 180, flushed after 240
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Redistributing: rip
Default version control: send version 1, receive any version
  Interface             Send  Recv  Triggered RIP  Key-chain
  Serial0/1/0           1     2 1
Automatic network summarization is in effect
Maximum path: 4
Routing for Networks:
      172.16.0.0
Passive Interface(s):
      FastEthernet0/0
Routing Information Sources:
      Gateway         Distance      Last Update
      172.16.2.253         120      00:00:06
Distance: (default is 120)
R2#
```

Základní konfigurace RIPv1

Zapnutí RIP: příkazem "router rip"

Chcete-li zapnout dynamický směrovací protokol, vstupte do globálního konfiguračního módu a použijte příkaz router. Jak je patrné z obrázku, pokud zadáte mezerník následovaný otazníkem, IOS zobrazí seznam všech dostupných směrovacích protokolů.

Pro vstup do konfigurace směrovacího protokolu RIP zadejte v globálním konfiguračním režimu "router rip". Všimněte si, že se změnila systémová výzva z globálního konfiguračního režimu na následující:

R1 (config-router) #

Tento příkaz nespouští přímo proces RIP. Místo toho poskytuje přístup ke konfiguraci nastavení směrovacích protokolů. Nejsou odeslány žádné směrovací aktualizace.

Pokud potřebujete ze zařízení zcela odstranit proces směrování RIP, negujte příkaz s "no" - "no router rip".Tento příkaz zastaví proces RIP a vymaže všechny stávající konfigurace RIP.

Zadání sítí

```
Router(config-router)#network directly-connected-classful-network-address
```

Příkaz network:

- Zapíná protokol RIP na všech rozhraních, které patří do specifikované sítě. Přidružená rozhraní budou nyní vysílat i přijímat aktualizace RIP.
- Inzeruje specifikované sítě ve směrovacích aktualizacích RIP na ostatní směrovače každých 30 sekund.

Klíčové charakteristiky RIPv1:

- **Protokol typu vektor vzdálenosti (*Distance-vector protocol*)**
- Používá UDP port 520.
- **Třídní protokol (*Classful protocol*)** (nepodporuje VLSM a nebo CIDR[18]).
- **Metrika (*metric*) = počet skoků (*router hop count*)**
- **Maximální počet skoků je 15., nedosažitelné cesty mají metriku 16. Cesty s počtem skoků větším než 15 jsou inzerované jako neplatné, nedostupné (*unreachable*)**
- **Periodické směrovací aktualizace jsou vysílány všesměrově (*broadcast*) každých 30 sekund na adresu 255.255.255.255.**
- 25 cest v jedné zprávě RIP
- Nepodporuje autentizaci (*authentication*)
- Implementuje rozložený horizont s otrávenými zpětnými informacemi (*split horizon with poison reverse*)
- Implementuje automaticky spouštěné aktualizace (*triggered updates*) při změně přímo připojené sítě
- V aktualizaci není obsažena maska podsítě (*subnet mask*)
- Administrativní vzdálenost (*administrative distance*) RIPv1 (i verze 2) je 120
- Použití: pouze v malých, plochých sítích nebo na okraji velkých sítí.

Základní nastavení RIPv1

```
Router(config)#router rip
Router(config-router)#network <přímo připojená síť v plné třídě>
... pro všechny přímo připojené sítě v plné třídě, které a do kterých chceme propagovat směrovací aktualizace
```

18 CIDR lze v RIPv1 použít **pouze na souvislé tranzitní síti**.

Příkaz ***network <síť>***:

- Zapíná protokol RIP na všech rozhraních spadajících pod uvedenou síť. Přidružená rozhraní budou jak vysílat, tak i přijímat směrovací aktualizace z protokolu RIP.
- Inzeruje uvedenou síť ve směrovacích aktualizacích protokolu RIP vysílaných na ostatní směrovače každých 30 sekund.

Poznámka: Pokud vložíte adresu podsítě, IOS ji automaticky převede na adresu sítě v plné třídě. Například, jestliže vložíte příkaz *network 192.168.1.32*, směrovač ho převede na *network 192.168.1.0*.

Propagace implicitní cesty:

```
Router(config)#router rip
Router(config-router)#default-information originate
```

Nastavení implicitní cesty manuálně administrátorem je u dynamické směrování obvyklé pouze na hraničním směrovači.

Ověření a hledání chyb konfigurace RIPv1

Příkazy:

show ip route

Zobrazená řádka směrovací tabulky

```
R      192.168.1.0/24 [120/1] via 172.16.0.1, 00:00:04, Serial0/0/1
```

Interpretace cesty:

Výstup	Popis
R	Identifikuje jako zdroj cesty RIP.
192.168.1.0	Indikuje adresu vzdálené sítě.
/24	Zobrazuje masku podsítě použitou na tuto síť.
[120/1]	Zobrazuje administrativní vzdálenost (120) a metriku (1 hop).
via 172.16.0.1,	Specifikuje next-hop IP adresu směrovače (adresu dalšího skoku), přes kterou se posílá provoz do uvedené vzdálené sítě.
00:00:04,	Specifikuje dobu uběhlou od poslední aktualizace (zde 4 sekundy). Další aktualizace přijde za 26 sekund.
Serial0/0/1	Specifikuje lokální rozhraní, přes které lze dosáhnout uvedenou vzdálenou síť.

show ip rip database

Zobrazí všechny nastavené trasy protokolu RIP v jeho interní databázi (včetně nastavených cest směrem na, v této chvíli, nedostupné přilehlé sítě, které proto nejsou ve směrovací tabulce):

```
R2#show ip rip database
0.0.0.0/0
    [0] via 0.0.0.0, 00:00:19
172.16.1.0/24
    [1] via 172.16.2.253, 00:00:13, Serial0/1/0
172.16.2.0/24     directly connected, Serial0/1/0
172.16.3.0/24     directly connected, FastEthernet0/0
R2#
```

V uvedeném příkladě je implicitní cesta nastavena na dosud nezprovozněnou přilehlou síť a je v databázi RIP, ale není ve směrovací tabulce.

```
R2#show ip route
<vynecháno>
Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 3 subnets
R       172.16.1.0 [120/1] via 172.16.2.253, 00:00:14, Serial0/1/0
C       172.16.2.0 is directly connected, Serial0/1/0
C       172.16.3.0 is directly connected, FastEthernet0/0
R2#
```

show ip protocols

Zobrazí aktuální konfiguraci všech směrovacích protokolů na spuštěných na směrovači, na kterém tento příkaz vložíte.

Z výstupu tohoto příkazu zjistíte (pro RIP) následující informace:

1. název směrovacího protokolu,
2. nastavené hodnoty časovačů a kdy bude další periodická aktualizace,
3. nastavenou filtraci vysílaných aktualizací a nastavenou redistribuci do jiných směrovacích protokolů,
4. jednotlivá rozhraní, která vysílají a přijímají aktualizace a ve které verzi RIP (1 nebo 2),
5. zda je v činnosti automatická sumarizace (ta je vždy na hranici plné třídy) a maximální počet cest RIP (*Maximum Path*) se stejnou cenou (*equal-cost*) do jedné konkrétní sítě,
6. směrování do uvedených sítí v plné třídě nastavené při konfiguraci směrovacího protokolu,
7. zdroje směrovacích informací – tj. sousední směrovače, ze kterých směrovač přijímá aktualizace; včetně informace o adrese dalšího skoku, administrativní vzdálenosti a času, kdy byla

přijata poslední aktualizace; poslední řádka výpisu zobrazuje administrativní vzdálenost tohoto směrovače.

```
R1#show ip protocols
Routing Protocol is "rip"
Sending updates every 30 seconds, next due in 25 seconds
Invalid after 180 seconds, hold down 180, flushed after 240
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Redistributing: rip
Default version control: send version 1, receive any version
  Interface             Send  Recv  Triggered RIP  Key-chain
  FastEthernet0/0       1     2 1
  FastEthernet1/0       1     2 1
  FastEthernet1/1       1     2 1
Automatic network summarization is in effect
Maximum path: 4
Routing for Networks:
      172.16.0.0
      192.168.1.0
Passive Interface(s):
Routing Information Sources:
      Gateway         Distance      Last Update
      172.16.3.254         120      00:00:00
      172.16.1.254         120      00:00:04
Distance: (default is 120)
R1#sh ip protocols
Routing Protocol is "rip"
Sending updates every 30 seconds, next due in 23 seconds
Invalid after 180 seconds, hold down 180, flushed after 240
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Redistributing: rip
Default version control: send version 1, receive any version
  Interface             Send  Recv  Triggered RIP  Key-chain
  FastEthernet0/0       1     2 1
  FastEthernet1/0       1     2 1
  FastEthernet1/1       1     2 1
Automatic network summarization is in effect
Maximum path: 4
Routing for Networks:
      172.16.0.0
      192.168.1.0
Passive Interface(s):
Routing Information Sources:
      Gateway         Distance      Last Update
      172.16.3.254         120      00:00:03
      172.16.1.254         120      00:00:06
Distance: (default is 120)
R1#
```

debug ip rip

Ladění (*debugging*) směrovacího protokolu RIP. On-line okamžitý výpis všech událostí (aktualizací, změn ve směrovací tabulce, ...) o příslušném směrovacím protokolu.

POZOR: ladění (*debugging*) velice zatěžuje CPU směrovače, zapínejte ho proto pouze na nezbytnou dobu pro dohledávání chyb (a nikoliv při běžném provozu).

```
R1#debug ip rip
```

```
RIP protocol debugging is on
RIP: received v1 update from 172.16.1.6 on Serial0/0
      172.16.1.8 in 1 hops
      192.168.2.0 in 1 hops
      192.168.3.0 in 2 hops
RIP: sending  v1 update to 255.255.255.255 via Loopback1 (10.0.0.1)
RIP: build update entries
      network 0.0.0.0 metric 1
      network 172.16.0.0 metric 1
      network 192.168.1.0 metric 1
      network 192.168.2.0 metric 2
      network 192.168.3.0 metric 2
RIP: sending  v1 update to 255.255.255.255 via Serial0/0 (172.16.1.5)
RIP: build update entries
      network 0.0.0.0 metric 1
      network 10.0.0.0 metric 1
      network 172.16.1.12 metric 1
      network 192.168.1.0 metric 1
      network 192.168.3.0 metric 2
RIP: sending  v1 update to 255.255.255.255 via Serial0/1 (172.16.1.14)
< ... >
R1#no debug all
All possible debugging has been turned off
```

Automatická sumarizace na hraničním směrovači

Jak víte, RIP (verze 1) je třídní směrovací protokol, který automaticky sumarizuje podsítě do třídní sítě (to znamená odmaskuje implicitní maskou) na hranicích hlavních (třídních) sítí *(major (classful) network)* (na tzv. hraničním směrovači (*boundary router*)).

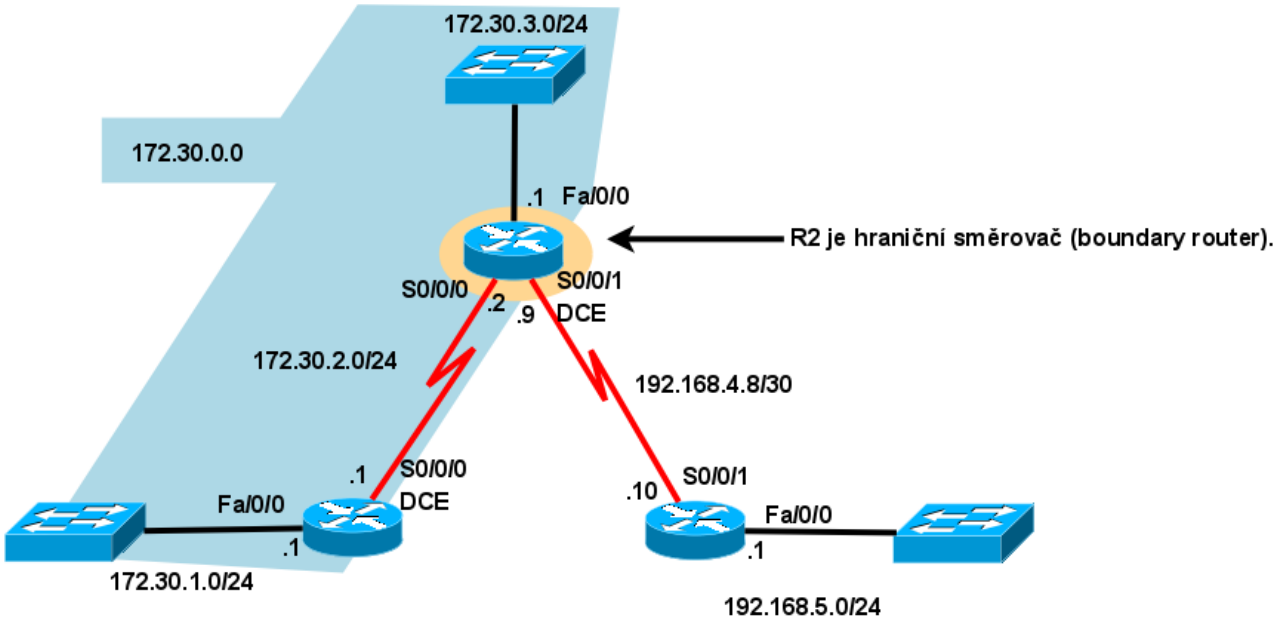
Pro aktualizace protokolu RIPv1 platí následující dvě pravidla:

- Jestliže síť ve směrovací aktualizaci a rozhraní na které je ta aktualizace přijatá spadají do stejné plnotřídní sítě, je na síť v řádce směrovací aktualizaci aplikována maska podsítě z tohoto síťového rozhraní.
- Jestliže síť ve směrovací aktualizaci a rozhraní na které je ta aktualizace přijatá spadají do různých plnotřídních sítí, je na síť v řádce směrovací aktualizace aplikována implicitní třídní síťová maska.

Směrovače, na kterých běží RIPv1 jsou omezeny používat tu samou masku podsítě pro všechny podsítě ve stejné třídní síti (CIDR).

Beztřídní směrovací protokoly (jako je RIPv2), dovolují stejné hlavní (třídní) síti používat různé masky podsítě v různých podsítích (VLSM).

Hraniční směrovač u RIP (RIP boundary Router)



Příkazy pro kapitolu 5, RIPv1

Příkaz (Command)	***Popis (Description)***
Router(config)# router rip	Spouští směrování RIP
Router(config-router)# network 10.0.0.0	Umožní aktualizace RIP na rozhraních spadajících pod síť (**v plné třídě**) 10.0.0.0
Router(config-router)#passive-interface fa0/0	Ukončuje vysílání aktualizací z rozhraní FastEthernet 0/0 (příjem ale trvá).
Router(config-router)#default-information originate	Tento směrovač propaguje ve svých aktualizacích RIP implicitní cestu. Nastaví se pouze na tom směrovači, kde je nastavena implicitní cesta (= hraniční směrovač (*boundary router*) s ISP).
Router# show ip protocols	Zobrazí detailní informace o všech procesech dynamického směrování na tomto směrovači.
Router# debug ip rip	Umožní monitoring aktualizací RIP, jak jsou vysílány a přijímány.
Router# no debug all	Vypíná veškeré ladění

Odstraňování chyb RIP

Router#debug ip rip	Zobrazí všechny aktivity RIP v reálném čase.
Router#show ip rip database	Zobrazí obsah databáze RIP.

Další příkazy pro RIP viz souhrn v kapitole 7.

Komplexní praktické laboratorní cvičení – dynamické směrování RIPv1 a sumarizace

Mějme 4 směrovače:

- R1, R2 a R3 (Cisco 2620XM se zásuvným modulem NM-2FE2W), které jsou zapojené do kruhu a ke každému z nich (s výjimkou R3) je připojen jeden přepínač (2950-24) a za přepínačem jedno PC.
- Za R3 je připojen směrovač R4 (1841) a teprve za ním přepínač (2950) s PC.
- Směrovače jsou propojeny do kruhu přes rozhraní FastEthernet. Použita je privátní adresa sítě ve třídě B podsíťovaná implicitní maskou třídy C (/24).
- Lokální sítě (s přepínači a hostitelskými počítači) jsou připojeny prostřednictvím privátních sítí v rozsahu třídy C s implicitní maskou.

Pro směrování použijte vnitřní beztřídní směrovací protokol RIPv1. Do sítí s pouze koncovými stanicemi zakažte propagaci RIP, ale tyto sítě samotné propagujte do ostatních sítí.

Postup práce:

1. Nejprve si nakreslete topologické schéma zapojení včetně adres sítí a názvů portů.
2. Vyplňte (doplňte) následující tabulku adres síťových rozhraní:

Zařízení	*Rozhraní*	*IP adresa*	*Maska*	*Brána*
R1	Fa0/0	192.168.1.0	255.255.255.0	
	Fa1/0	172.16.1.0	255.255.255.0	
	Fa1/1	172.16.3.0	255.255.255.0	
R2	...			
...				

3. Nastavte a ověřte směrování, **automaticky sumarizovanou cestu v plné třídě na R4** a konvergenci po změně topologie (výpadku linky).
4. Na směrovači R4 nastavte implicitní cestu na odchozí rozhraní směrem na switch a PC. Tuto implicitní cestu pak propagujte na ostatní směrovače.
5. Na směrovačích s připojenými sítěmi, které obsahují pouze koncová zařízení (PC) (= netranzitní síť) zakažte propagaci RIP dovnitř těchto sítí natavením příslušného rozhraní jako pasivního (*passive-interface*).

Příklad konfigurace RIP na R2:

```
!
router rip
 passive-interface FastEthernet0/0
 network 172.16.0.0
```

```
 network 192.168.2.0
!
```

Kontrolní opakovací otázky a odpovědi (kvíz):

1) Co je pravda ohledně příkazu „**debug ip rip**“?
   a) Vypisuje aktualizace RIP online, tak jak jsou aktuálně vysílány a přijímány
2) Jaký problém pomáhá řešit příkaz „**passive interface**“?
   a) Pasivní rozhraní (= neposílá aktualizace, ale pouze je přijímá) předchází zbytečné zátěži linky a procesoru nepotřebnými směrovacími aktualizacemi (do netranzitních sítí, *stub network*)
3) Co ze směrovače činí hraniční směrovač (*boundary router*) pro RIP?
   a) Směrovač má několik rozhraní ve více než jedné hlavní třídní síti (=> probíhá na něm automatická sumarizace směru do plné třídy, u RIPv2 lze sumarizece vypnout u RIPv1 sumarizace nelze vypnout => je to vlastnost tohoto směrovacího protokolu)
4) Který příkaz je v RIP použit pro propagaci implicitní cesty?
   a) **default-information originate**
5) Který příkaz vytvoří kandidáta na implicitní cestu v RIP?
   a) Např.: **ip route 0.0.0.0 0.0.0.0 serial0/0/0**
6) Máte zapojené čtyři směrovače do kruhu pomocí čtyř propojovacích (= tranzitních) sítí: 192.168.8.0/30, 192.168.10.4/30, 192.168.11.12/30 a 192.168.9.0/30. Všechna rozhraní směrovačů jsou nastavená a zapnutá. L2 protokoly funkční. Na směrovačích je spuštěn RIP-v1. Nelze se dopinknout mezi dvěmi netranzitními sítěmi (v třídě A).
   a) Tranzitní síť je tvořena nesouvislými sítěmi (podsítě různých sítí v plné třídě - vzhledem k implicitní masce třídy C /24).
7) Jak směrovač, na kterém běží RIPv1, masku podsítě, kterou přijal ze směrovací aktualizace?
   a) Směrovač buď použije implicitní masku, nebo pokud jde o podsíť plné třídy, ve které leží rozhraní, na které aktualizace přišla, použije masku nastavenou na přijímajícím rozhraní.
8) Jaký je účel příkazů **network** při konfiguraci směrovacího protokolu RIP?
   a) Identifikují všechny přímo připojené sítě, které budou zahrnuté do směrovacích aktualizací
9) Spárování příkazů a jejich popisů (pro příkazy, které síťový administrátor používá pro ověření konfigurace směrovače):
   a) výpis aktuální konfigurace rozhraní a směrovacích protokolů = show running-config
   b) zobrazí zda jsou rozhraní zapnutá a protokoly funkční = show ip interfaces
   c) online výpis směrovacích aktualizací, tak jak jsou aktuálně vysílány a přijímány = debug

ip rip

d) vypíše všechny běžící směrovací protokoly na směrovači, a které sítě propagují = show ip protocols

e) ověří, zda jsou všechny požadované směry (cílové sítě) nainstalované do směrovací tabulky = show ip route

Kapitola 6 - VLSM a CIDR

V této kapitole se naučíme:

- Porovnat a zdůraznit rozdíly třídní (*classful*) a beztřídní (*classless*) IP adresace
- Přehled VLSM a vysvětlit přínosy beztřídní IP adresace
- Popsat roli beztřídního standardu CIDR (*Classless Inter-Domain Routing*) při efektivním použití nedostatkových adres IPv4

Adresní systémy pro IPv4

Před rokem 1981, používaly IP adresy k určení síťové části adresy pouze prvních 8 bitů, omezujících Internet, tehdy známý jako ARPANET, na pouhých 256 sítí. Brzo začalo být zřejmé, že to nebude stačit, aby byl dostatek volných adres.

V roce 1981 modifikoval dokument RFC 791 32-bitovou adresu protokolu IPv4 na tři různé třídy: A,B a C. Třída A používala 8 bitů pro síťovou část, třída B používala 16 bitů a třída C používala 24 bitů. Tento formát se stal známým jako třídní IP adresace.

Tento vývoj třídní adresace na čas řešil problém omezení na 256 sítí. O desetiletí později bylo ale jasné, že se adresní prostor rychle vyčerpává. Jako odpověď na to IETF zavedl adresní systémy CIDR (*Classless Inter-Domain Routing*) (RFC 1519 - 1993) a VLSM (*Variable Length Subnet Masking*).

Jednotliví poskytovatelé ISP nyní mohou přiřadit jednu část třídní sítě jednomu zákazníkovi a další část jinému zákazníkovi. Toto nesouvislé přiřazování adresy ze strany ISP bylo paralelně následováno vývojem beztřídních směrovacích protokolů. Pro srovnání: Třídní směrovací protokoly vždy sumarizují na hranici třídy a neobsahují masku v aktualizacích. Beztřídní směrovací protokoly obsahují ve směrovacích aktualizacích masku. Beztřídní směrovací protokoly diskutované v tomto kurzu jsou RIPv2, EIGRP a OSPF.

Se zavedením VLSM a CIDR musí ale síťoví administrátoři začít používat další znalosti z podsíťování. VLSM je jednoduše podsíťování podsítí. Podsítě mohou být podsíťovány v různých úrovních (s různými délkami masek). A co více, stalo se možným sumarizovat celou sadu třídních sítí do agregovaného směru neboli nadsítě (*supernet*) s maskou kratší než je implicitní třídní maska.

Počet sítí a hostitelů v jedné síti podle tříd

Třída adres (Address Class)	***Rozsah prvního oktetu (First Octet Range)***	***(Počet možných sítí) Number of Possible Networks***	***Počet hostitelů v jedné síti (Number of Hosts per Network)***
A	0 až 127	128 (2 jsou rezervované)	16 777 214
B	128 až 191	16 384	65 534
C	192 až 223	2 097 152	254

Třídní a beztřídní adresace a směrování

Pojmy:

	Třídní (Classful)	***Beztřídní (Classless)***
IP adresace	Pouze sítě ve třídách A, B, C	• CIDR • VLSM
Směrování	V aktualizaci není maska	V aktualizaci je i maska

IP adresace

- **Třídní** - IP adresy pouze v celých třídách (plýtvá nedostatkovými IP adresami)
- **Beztřídní – CIDR** – IP adresy jsou v podsítích třídní adresy a všechny masky jsou stejné (adresní bloky jsou stejně velké) (mírně snižuje plýtvání adresami, **při sumarizaci** snižuje velikost směrovací tabulky, a tím také snižuje aktualizační provoz) (RFC 1517)
- **Beztřídní – VLSM** – IP adresy jsou v podsítích jiné podsítě (masky jsou různé), je třeba dávat pozor na překrývání adresních bloků (*overlapping*). Překrývání adres je detekováno a odmítnuto přímo operačním systémem směrovače (IOS). Použití VLSM snižuje plýtvání IP adresami, využívá adresní prostor ještě lépe než CIDR. Postup při vytváření: postupujte od největšího adresního bloku k nejmenšímu. Nezapomeňte, že prodloužení masky o jeden bit zmenší adresní blok na polovinu.

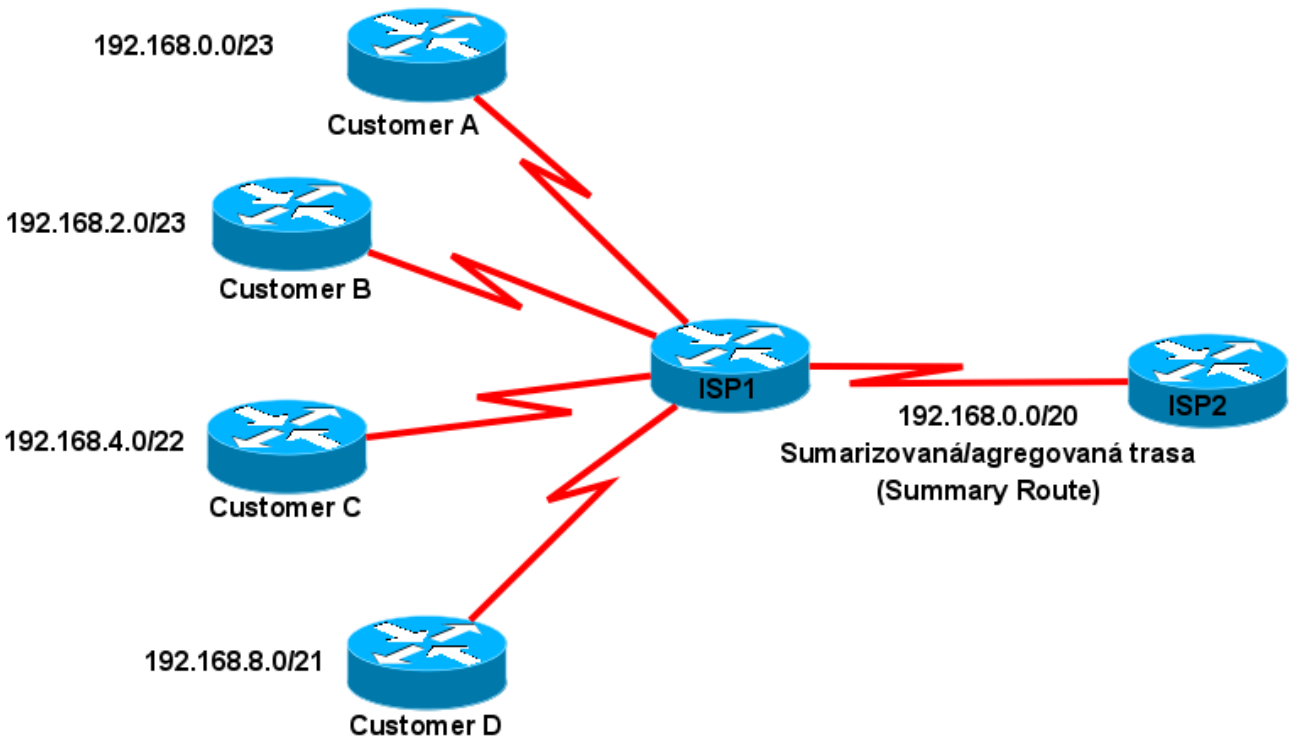
Směrování

- **Třídní** – nepřenáší se masky, jako maska je použita buď implicitní maska nebo, pokud cílová adresa leží ve stejné síti (vzhledem k implictní masce) jako rozhraní směrovače (na které aktualizace přišla (*ingress interface*)), tak maska nastavená na tomto síťovém rozhraní. Tzn. Lze použít i **CIDR v souvislé tranzitní sítí**. **Souvislá , na sebe navazující, síť (*contiguous network*)** = všechny podsítě jsou v jedné supersíti, nadsíti (*supernet*) v plné třídě (tj. vzhledem k implicitní masce třídy).
- **Beztřídní** – lze použít VLSM.

Zvláštní typy rozhraní směrovače

- **Loopback (zpětná smyčka)** – tato virtuální rozhraní mají v konfiguračním režimu pro rozhraní nastavenou IP adresu a masku, jsou implicitně administrativně zapnuté a mají nahozený L2 protokol, aniž je na nich vůbec něco připojeného. Na toto rozhraní lze pinknout a dostat odpověď. Používá se pro nastavení implicitní cesty, a simulaci (dosud) neexistujícího připojení na ISP. Nastavenou implicitní cesta lze dynamicky propagovat (*default-information originate*).
- **Null (neexistující prázdné rozhraní)** – tato rozhraní lze použít pro nastavení statických cest, které lze dynamiky propagovat (*redistribute static*).

CIDR a sumarizace trasy (CIDR and Route Summarization)



Výpočet sumarizované (agregované) cesty

CIDR:

192.168.64.0/22	11000000.10101000.01000000.00000000
192.168.68.0/22	11000000.10101000.01000100.00000000
192.168.72.0/22	11000000.10101000.01001000.00000000
192.168.76.0/22	11000000.10101000.01001100.00000000
Agregovaná cesta : 192.168.64.0/20	11000000.10101000.01000000.00000000

VLSM:

172.16.0.0/19	10101100.00010000.00000000.00000000
172.16.32.0/19	10101100.00010000.00100000.00000000
172.16.64.0/18	10101100.00010000.01000000.00000000
172.16.128.0/17	10101100.00010000.10000000.00000000

Agregovaná cesta : 172.16.0.0/16	10101100.00010000.00000000.00000000

Příklady sumarizace

Příklad

Máte 4 následující třídní sítě: 172.20.0.0/16 , 172.21.0.0/16 , 172.22.0.0/16 a 172.23.0.0/16.

Spočtěte sumarizaci do nadsítě (*supernet*) bez použití binárního tvaru.

Řešení:

B/C numericky nejvyšší sítě mínus síťová adresa nejnižší sítě, dvojkový doplněk z toho je roven masce nadsítě.

172.23.255.255 -172.20.0.0 = 0.3.255.255 dvojkový doplněk je maska nadsítě 255.252.0.0 = /14.

Sumarizovaná nadsíť je tedy 172.20.0.0/14.

Další podobný příklad

Máte 6 následujících třídních sítí: 172.16.0.0/16, 172.17.0.0/16, 172.18.0.0./16, 172.19.0.0/16 , 172.20.0.0/16 a 172.21.0.0/16.

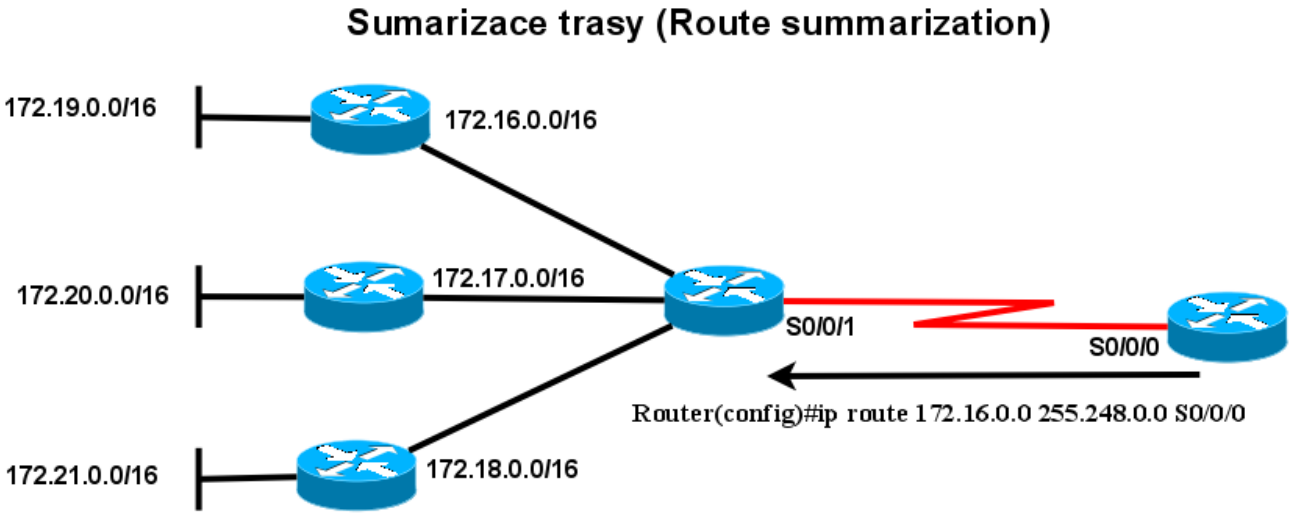
Spočtěte sumarizaci do nadsítě (*supernet*) bez použití binárního tvaru.

Řešení:

B/C numericky nejvyšší sítě mínus síťová adresa nejnižší sítě, dvojkový doplněk z toho je roven masce nadsítě.

172.21.255.255 -172.16.0.0 = 0.5.255.255 → oprava na číslo, které má binárně zprava samé jedničky: 0.7.255.255, jeho dvojkový doplněk je maska nadsítě 255.248.0.0 = /13.

Sumarizovaná nadsíť je tedy 172.20.0.0/14.



Příklad VLSM

Máte přidělený adresní blok 172.16.128.0/17. Tento rozsah máme v následujícím příkladu **chybně** rozdělen do v následující tabulce uvedených sítí (**viz originální Aktivita 6.4.3**). Zkontrolujte je a opravte nalezené chyby v chybně navržených adresách sítí.

Kontrolujte vždy automaticky všechna zadání jednotlivých cvičení! Dá to vždy méně práce, než opravovat chybnou konfiguraci.

Podsíť (Subnet)	***Počet potřebných IP adres***	***Adresa sítě – původně navržená***	***Velikost bloku - Adresa sítě - opravená***
HQ LAN1 (HQ = Centrála)	16000	172.16.128.0/19	0.0.64.0 - 172.16.128.0/18
HQ LAN2	8000	172.16.192.0/18	0.0.32.0 - 172.16.192.0/19
Branch1 LAN1 (Branch = Pobočka)	4000	172.16.224.0/20	0.0.16.0 - 172.16.224.0/20
Branch1 LAN2	2000	172.16.240.0/21	0.0.8.0 - 172.16.240.0/21
Branch2 LAN1	1000	172.16.244.0/24	0.0.4.0 - 172.16.248.0/22
Branch2 LAN2	500	172.16.252.0/23	0.0.2.0 - 172.16.252.0/23
Linka z HQ do Branch1	2	172.16.254.0/28	0.0.0.4 - 172.16.254.0/30
Linka z HQ do Branch2	2	172.16.254.6/30	0.0.0.4 - 172.16.254.4/30
Linka z Branch1 do Branch2	2	172.16.254.8/30	0.0.0.4 - 172.16.254.8/30

Laboratorní cvičení - příklad

1. Dva směrovače R1 a R2 propojte přes Ethernet v privátní síti třídy B, podsíťované implicitní maskou pro třídu C (/24).
2. Na R1 nakonfigurujte Loopback 0 (IP adresa a maska), na toto odchozí rozhraní Loopback 0 nastavte implicitní cestu, implicitní cestu dynamicky propagujte pomocí RIPv2.
3. Na R2 je nastavena statická cesta na neexistující rozhraní Null 0, nastavené statické cesty dynamicky propagujte pomocí RIPv2 se zvolenou metrikou = 12 (lze zvolit v rozsahu 0 až 16).

Částečná konfigurace R1:

```
!
interface Loopback0
 ip address 172.16.100.254 255.255.255.0
!
<vynecháno>
!
```

```
router rip
 version 2
 network 172.16.0.0
 default-information originate
!
ip classless
ip route 0.0.0.0 0.0.0.0 Loopback0
!
```

Částečná konfigurace R2:

```
!
router rip
 version 2
 redistribute static metric 12
 network 172.16.0.0
!
ip classless
ip route 192.168.1.0 255.255.255.0 Null0
!
```

Kontrolní opakovací otázky a odpovědi (kvíz):

1) Směrovací protokoly, které podporují a nepodporují VLSM:
   a) podporují VLSM: RIPv2, EIGRP, IS-IS, OSPF
   b) nepodporují VLSM: RIPv1, IGRP.
2) Přiřaďte k termínům odpovídající popisky:
   a) VLSM:
      i. schopnost pro jednu (pod)síť vytvářet její podsítě s různými maskami,
      ii. šetří adresy (lépe využívá daný adresní blok).
   b) sumarizace směru:
      i. je také známa jako supernetting,
      ii. snižuje počty řádek ve směrovací tabulce,
      iii. v jedné IP adrese kombinuje více sítí.
3) Které dvě metody se používají, aby bylo možné stále používat IPv4, i když už jsou třídní a CIDR adresní bloky vyčerpané?
   a) VLSM,
   b) privátní adresy a jejich překlad na veřejné pomocí NAT (Network Address Translation) =

IP maškaráda.

4) Pro síť 192.168.16.0 byly použity následující masky: 255.255.255.252, 255.255.255.240, 255.255.255.192. Popište nejefektivnější použití každé jednotlivé masky:
   a) maska /30 pro spojení point-to-point k síti typu WAN,
   b) maska /28 pro malé sítě do 14 hostitelů včetně,
   c) maska /26 pro velké sítě do 62 hostitelů včetně.
5) Když použijete třídní adresu ve třídě A, kolik oktetů tvoří síťovou část IP adresy?
   a) 1
6) Vyberte VLSM podsítě sítě 172.16.0.0, které poskytnou uvedený celkový počet hostitelů v každé jednotlivé podsíti.
   a) 2 hostitelé = 172.16.16.64/30
   b) 60 hostitelů = 172.16.5.128/26
   c) 250 hostitelů = 172.16.18.0/24
   d) 8000 hostitelů = 172.16.128.0/19
   e) 16000 hostitelů = 172.16.64.0/18
7) Inženýr sumarizuje na směrovači A dvě skupiny směrů (cest). Skupinu A: 192.168.0.0/30, 192.168.0.4/30, 192.168.0.8/30, 192.168.0.16/29 a Skupinu B: 192.168.4.0/30, 192.168.5.0/30, 192.168.6.0/30, 192.168.7.0/29. Jaká sumarizace bude funkční pro všechny uvedené podsítě?
   a) 192.168.0.0/21 (řešení: 192.168.7.0-192.168.0.0= 0.0.7.0 => zleva 21 nul)
8) Kolik bitů je použito v adresním prostoru protokolu IPv4?
   a) 32
9) Rozdělte adresy do tříd:
   a) A: 123.90.78.45, 125.33.33.33, 126.0.0.0,
   b) B: 128.44.30.1, 129.68.11.45, 191.254.45.0.
10) Do jednoho směrovače (R2) máte z levé strany připojeny tři směrovače na kterých jsou připojeny následující netranzitní sítě: 172.16.0.0/16, 172.17.0.0/16 a 172.18.0.0/16 (na každém směrovači jedna). Na tento R2 směrovač je potom z pravé strany připojen jeden směrovač R1. Co by měl administrátor této sítě aplikovat, aby snížil počet řádek ve směrovací tabulce na směrovači R1.
    a) CIDR

Kapitola 7 - Protokol RIP verze 2

V této kapitole se naučíme:

- Střetnout se s nimi a popsat, jaká jsou omezení protokolu RIPv1
- Použít základní konfigurační příkazy pro RIPv2 (*Routing Information Protocol version 2*) a vyhodnotit aktualizace beztřídního směrování
- Analyzovat výstup ze směrovače, abychom poznali, jak RIPv2 podporuje beztřídní VLSM (*Variable Lenght Subnet Mask*) a CIDR (*Classless Interdomain Routing*)
- Určit příkazy pro ověření správné činnosti RIPv2 a identifikovat běžné problémy
- Konfigurovat, ověřit a odstranit chyby RIPv2 v praktickém laboratorním cvičení

RIP verze 2 a verze 1

Protokol RIP verze 2 (RIPv2) je definován v RFC 1723. Je to první beztřídní směrovací protokol popisovaný v tomto kurzu. Tabulka *Klasifikace dynamických směrovacích protokolů* v kapitole 3 dává protokol RIPv2 do správné souvislosti vzhledem k jiným směrovacím protokolům. Přestože je RIPv2 vhodným směrovacím protokolem pro některá prostředí, ztratil postupně popularitu ve srovnání s jinými směrovacími protokoly, jako jsou EIGRP, OSPF a IS-IS, které nabízejí více funkcí a jsou více škálovatelné.

I když mohou být méně populární než jiné směrovací protokoly, jsou obě verze RIP ještě vhodné v některých situacích. Přestože RIP postrádá schopnosti mnoha pozdějších protokolů, jeho jednoduchost a široké využití v různých operačních systémech z něj činí ideálního kandidáta pro menší, homogenní sítě, kde je nezbytná podpora více dodavatelů - zejména v prostředích UNIX.

Protože je třeba abyste pochopili RIPv2 - i když ho nebudete chtít používat - zaměří se tato kapitola na rozdíly mezi třídním směrovacím protokolem (RIPv1) a beztřídním směrovacím protokolem (RIPv2) spíše než na samotné detaily RIPv2. Hlavním omezením RIPv1 je to, že je třídním směrovacím protokolem. Jak víte, třídní směrovací protokoly nezahrnují se síťovou adresou do směrovacích aktualizací masku podsítě, což může způsobit problémy v nesouvislých podsítích nebo sítích, které používají adresní strukturu s proměnnou délkou masky (VLSM). Vzhledem k tomu, že RIPv2 je beztřídní směrovací protokol, jsou masky podsítě zahrnuty ve směrovacích aktualizacích, což činí RIPv2 více kompatibilní s moderními směrovacími prostředími.

RIPv2 je vlastně vylepšení funkcí a rozšíření RIPv1, spíše než zcela nový protokol. Některé z těchto rozšířených funkce obsahují:

- Adresu dalšího přeskoku (*Next-hop adres*) ve svých směrovacích aktualizacích
- Použití skupinových adres (*multicast*) v zasílaných aktualizacích
- Je k dispozici možnost ověřování (autentizace)

Stejně jako RIPv1, je RIPv2 směrovacím protokolem typu vektor vzdálenosti. Obě verze RIP sdílejí následují funkce a omezení:

- Použití zadržovacího (*holdown*) a dalších časovačů pro zabránění vzniku směrovacích smyček.

- Použití rozloženého horizontu (*split horizon*) nebo rozloženého horizontu s otrávenou zpětnou informací (*split horizon with poison reverse*) také pro zabránění směrovacích smyček.
- Použití automaticky spouštěné aktualizace (*triggered update*), když dojde ke změně v topologii, pro rychlejší konvergenci.
- Omezení maximálního počtu přeskoků (*hop counts*) na 15 přeskoků, přičemž počet přeskoků rovný 16 znamená nedosažitelnost sítě.

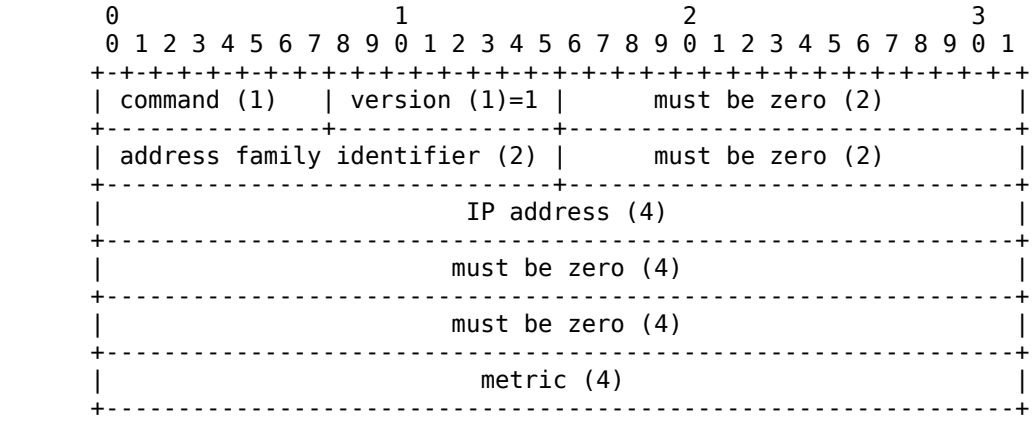
Omezení protokolu RIPv1

- Nepodporuje nesouvislé (tranzitní) sítě
- Nepodporuje VLSM
- Nepodporuje CIDR, které mají sumarizované trasy s kratší maskou než je třídní (implicitní) maska této sítě.

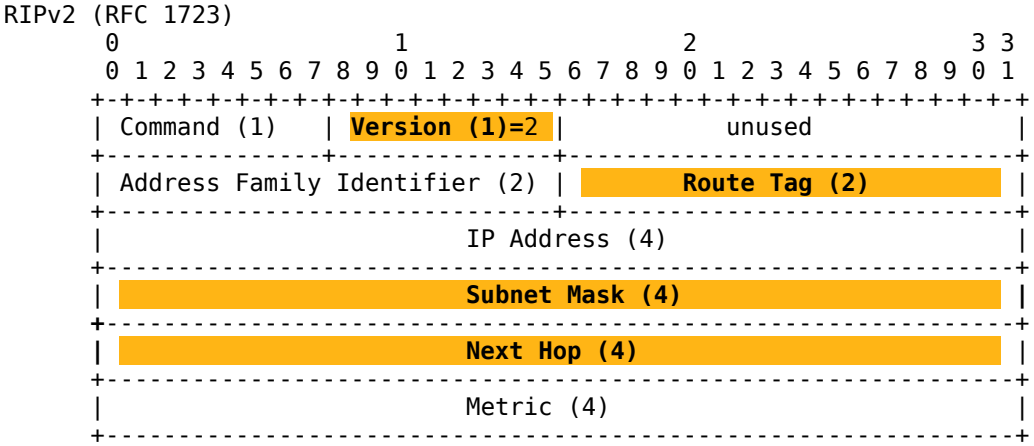
Formát zpráv RIPv1 a RIPv2

(V závorkách je u názvu pole uvedena jeho velikost v bajtech.)

RIPv1 (RFC 1058)



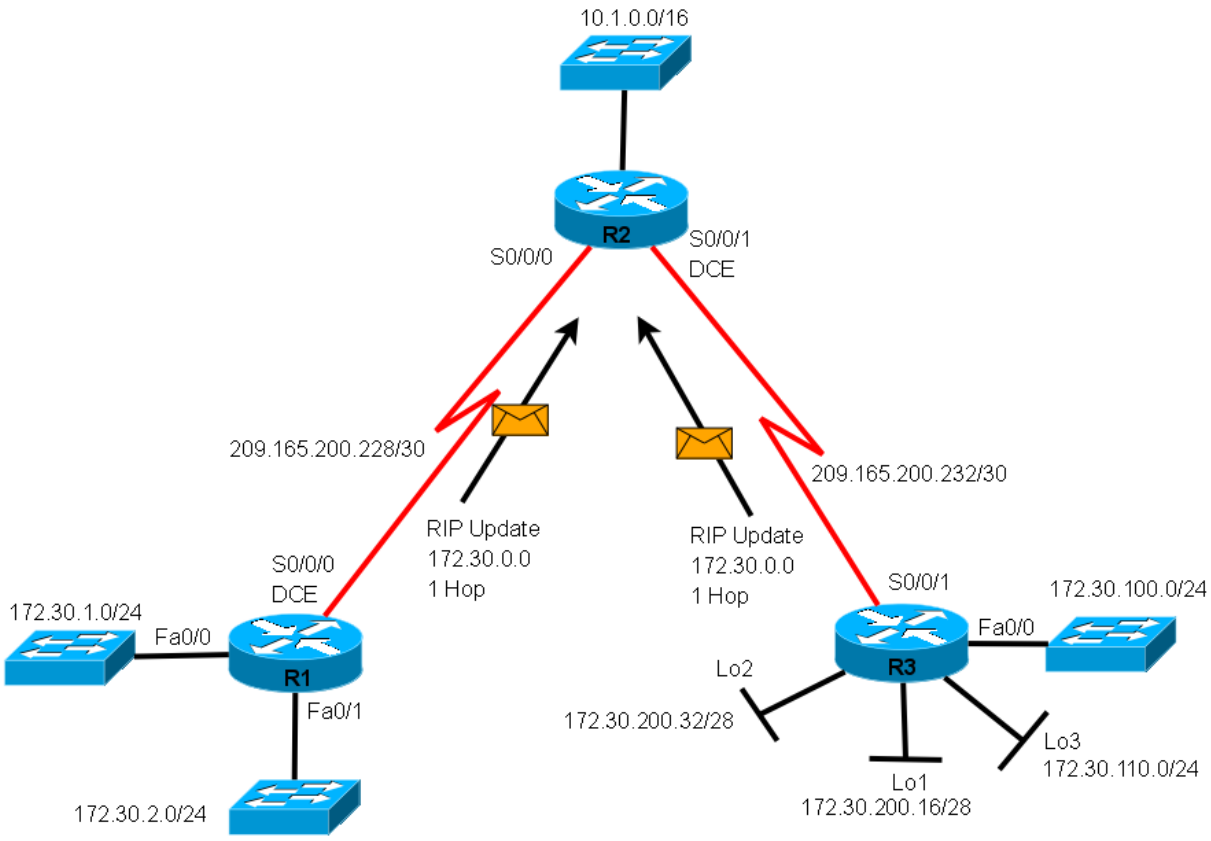
RIPv2 (RFC 1723)



Automatická sumarizace a RIPv2

Implicitně RIPv2 automaticky sumarizuje sítě na hranicích plnotřídní sítě (odmaskováním implicitní maskou třídy), přesně tak jako RIPv1. Zda je sumarizace zapnuta zjistíte příkazem „**show ip protocols**“ a jeho odpovědí "*automatic summarization is in effect.*"

Automatická sumarizace na hranici třídní sítě (Automatic Summarization)



Vypnutí automatické sumarizace

K vypnutí automatické sumarizace použijte příkaz „**no auto-summary**“ v konfiguračním režimu směrování (config-router)# . Pro RIPv1 je tento příkaz neplatný a ačkoliv ho můžete v IOSu pro RIPv1 také zadat nemá žádný efekt.

Jakmile je automatická sumarizace vypnuta, RIPv2 již dále na hranicích třídní sítě nesumarizuje do plné třídy. RIPv2 nyní do směrovacích aktualizací zahrne všechny podsítě a jejich masky. Pro ověření, že je automatická sumarizace vypnutá použijte příkaz **show ip protocols**, který v tomto případě vrátí, že "*automatic network summarization is not in effect.*"

POZNÁMKA: Uvědomte si, že mohou nastat případy konfigurace, kdy ani přepnutí RIP z verze 1 do verze 2 nestačí ke zprovoznění a je nutné ještě vypnout automatickou sumarizaci. (Například: netranzitní podsítě a mezi nimi alespoň dvě třídní tranzitní sítě. Viz předchozí obrázek.) Takže je vhodné si vždy rozmyslet, zda je nutné sumarizaci vypnout či nikoliv.

```
R2#show ip protocols
Routing Protocol is "rip"
Sending updates every 30 seconds, next due in 5 seconds
Invalid after 180 seconds, hold down 180, flushed after 240
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Redistributing: rip
Default version control: send version 2, receive 2
  Interface             Send  Recv  Triggered RIP  Key-chain
  FastEthernet0/0       2     2
  Serial0/1/1           2     2
Automatic network summarization is in effect
Maximum path: 4
Routing for Networks:
      10.0.0.0
      172.16.0.0
Passive Interface(s):
Routing Information Sources:
      Gateway         Distance      Last Update
      10.0.0.253           120      00:00:13
Distance: (default is 120)
R2#
```

Charakteristiky RIPv2

- **Protokol typu vektor vzdálenosti (*Distance-vector protocol*)**
- Používá UDP port 520.
- **Beztřídní (*classless*) směrovací protokol** (podporuje CIDR i VLSM)
- **Metrika = počet skoků (*router hop count*)**
- **Maximální počet skoků = 15, nekonečné (nedosažitelné) (*infinite (unreachable)*) cesty mají metriku 16**
- **Periodické aktualizace jsou posílány každých 30 sekund na skupinovou (*multicast*) adresu 224.0.0.9.**
- 25 cest (*routes*) v jedné zprávě RIP (24 jestliže používáte autentizaci (*authentication*))
- Podporuje autentizaci (*authentication*)[19] (formát zprávy pro toto viz RFC 1723)

19 Nebudeme procvičovat. Pro zájemce: jsou to příkazy: v globální konfiguraci: key chain ..., key 1, key-string ..., a v konfiguraci rozhraní: ip rip authentication key-chain ... .

- Implementuje rozložený horizont s otrávenými zpětnými informacemi (*split horizon with poison reverse*)
- Implementuje automaticky spouštěné aktualizace (*triggered updates*) při změně přímo připojené sítě
- V aktualizaci je obsažena maska podsítě (*subnet mask*)
- Administrativní vzdálenost (*administrative distance*) RIPv2 je 120
- Použití: v malých, plochých sítích nebo na okraji velkých sítí.

Postup hledání chyb konfigurace

Je několik způsobů, jak hledat a odstraňovat chyby v konfiguraci RIPv2. Mnoho z těchto příkazů lze použít i u ostatních směrovacích protokolů.

Je vždy nejlepší začít od základů:

1. Přesvědčte se, že jsou rozhraní zapnutá a funkční. Příkazem show ip interface brief.
2. Ověřte kabely. (show controllers ...)
3. Na každém síťovém rozhraní zkontrolujte IP adresu a masku. (show ip interface brief)
4. Odstraňte v konfiguraci všechny nepotřebné příkazy, které buď nejsou již dále potřeba, nebo budou přepsány jinými příkazy.

Obvyklé problémy s RIPv2

Když odstraňujete chyby specifické pro RIPv2, je zde několik oblastí, kde je dobré začít hledat:

1. ověřit, zda je příkaz „version 2“ nastaven na všech směrovačích,
2. nesprávně vložené nebo chybějící příkazy „network“ – jsou příčinou, že aktualizace nejsou vysílány nebo přijímány na/z určitého rozhraní,
3. pokud není speciální důvod k posílání podsítí do třídní sítě, vypněte automatickou sumarizaci příkazem „no auto-summary“,
4. též je dobré zjistit, zda vysílání aktualizací není bezděčně vypnuto příkazem „passive-interface“.

Autentizace

Bezpečnostní problémem jakéhokoliv směrovacího protokolu je možnost přijetí a použití neplatných směrovacích aktualizací. Zdrojem těchto neplatných směrovacích aktualizací může být útočník ve zlé vůli se pokoušející narušit síť a nebo zkoušející zachytávat pakety podvedením směrovače, aby posílal data do nesprávného cíle. Jiným zdrojem neplatných aktualizací může být nesprávně nastavený směrovač a nebo hostitelský počítač, na kterém běží směrovací protokol, o kterém jeho uživatel neví.

Ať už z jakéhokoliv důvodu, je dobré autentizovat směrovací informace. RIPv2, Enhanced IGRP (EIGRP), Open Shortest Path First (OSPF), Intermediate System–to–Intermediate System (IS-IS) a Border Gateway Protocol (BGP) mohou nastaveny tak, aby šifrovaly a autentizovaly směrovací

informace.

Tato praxe potom jednak utajuje obsah směrovacích informací (šifruje aktualizace ale samozřejmě nešifruje směrovací tabulku). Směrovače potom také akceptují informace z druhého směrovače pouze pokud je nastaven se stejným heslem nebo autentizačními informacemi.

Poznámka: Autentizace nebude v této kapitole dále diskutována. (Bude diskutováno později v souvislosti se zabezpečením – ve čtvrtém semestru.)

Příkazy pro kapitolu 7, RIPv2

Příkaz (Command)	***Popis (Description)***
Router(config)# router rip	Zapne směrování RIP.
Router(config-router)# version 2	Zapne verzi 2 protokolu RIP.
Router(config-router)# no version 2	Přepne zpět do verze 1 protokolu RIP. RIP přijímá aktualizace verze 2 i 1, vysílá pouze verzi 1.
Router(config-router)# network a.b.c.d	Nastaví přímo připojené sítě v plných třídách, které chcete propagovat, a dokterých se bude propagovat.
Router(config-router)# no auto-summary	RIPv2 automaticky sumarizuje sítě na směrovači na hranici plné třídy. (Sumarizace jse zapnuta z důvodu zpětné kompatibility s RIPv1.) Tento příkaz vypne automatickou sumarizaci (na hranici plné třídy). (V RIPv1 vypnout nelze (tam se sumarizuje do plné třídy na hranici plné třídy vždy a bezpodmínečně).
Router# debug ip rip	Zobrazí všechny aktivity RIP v reálném čase (tak jak postupně přicházejí). Pozor zbytečně zatěžuje směrovač a proto použijte pouze při ladění problémů.
Router# undebug all	Vypne veškeré ladění na směrovači.

Souhrn příkazů pro obě verze RIPv1 i RIPv2

Povinné příkazy pro směrování protokolem RIP (pro obě verze 1 i 2)

Router(config)#router rip	Zapne směrovací protokol RIP.
Router(config-router)#network w.x.y.z	w.x.y.z je síťová adresa (číslo sítě) přímo připojené sítě, kterou chceme inzerovat (propagovat).

POZNÁMKA: Inzerovat v RIP lze pouze sítě v plné třídě nikoliv podsítě:

Router(config-router)#network 172.16.0.0

nikoliv

Router(config-router)#network 172.16.10.0

Jestliže inzerujete podsíť, nedostanete zpět chybovou zprávu, protože směrovač automaticky konvertuje podsíť na třídní adresu (odmaskuje implicitní maskou pro příslušnou třídu).

Volitelné příkazy pro RIP (souhrn pro obě verze 1 i 2)

Router(config)#no router rip	Vypne směrovací proces RIP.
Router(config-router)#no network w.x.y.z	Odstraní síť w.x.y.z ze směrovacího procesu RIP.
Router(config-router)#version 2	RIP nyní bude vysílat a přijímat pakety RIPv2.
Router(config-router)#version 1	RIP nyní bude vysílat a přijímat pouze pakety RIPv1.
Router(config-if)#ip rip send version 1	Toto rozhraní bude vysílat pouze pakety RIPv1.
Router(config-if)#ip rip send version 2	Toto rozhraní bude vysílat pouze pakety RIPv2.
Router(config-if)#ip rip send version 1 2	Toto rozhraní bude vysílat oboje pakety RIPv1 i RIPv2.
Router(config-if)#ip rip receive version 1	Toto rozhraní bude přijímat pouze pakety RIPv1.
Router(config-if)#ip rip receive version 2	Toto rozhraní bude přijímat pouze pakety RIPv2.
Router(config-if)#ip rip receive version 1 2	Toto rozhraní bude přijímat oboje pakety RIPv1 i RIPv2.
Router(config-router)#no auto-summary	RIPv2 sumarizuje sítě na hranici plné třídy. Tento příkaz automatickou sumarizaci vypíná.
Router(config-router)#passive-interface s0/0/0	Z tohoto rozhraní nebudou odesílány aktualizace RIP. (Aktualizace mohou ale být přijímány.)
Router(config-router)#neighbor a.b.c.d	Definuje konkrétní sousedy, se kterými si vyměňuje informace.
Router(config-router)#no ip split-horizon	Vypne rozložený horizont (*split horizon*) (implicitně je rozložený horizont zapnut).
Router(config-router)#ip split-horizon	Znovu zapne rozložený horizont.
Router(config-router)#timers basic 30 90 180	Změní časovače RIP:

270 360	30 = Aktualizační, Update timer (v sekundách) 90 = Neplatné cesty, Invalid timer (v sek.) 180 = Zadržovací, Hold-down timer (v sek.) 270 = Vyprazdňovací, Flush timer (v sekundách) 360 = Doba spánku, Sleep time (v milisek.)
Router(config-router)#maximum-paths *N*	Omezí počet cest pro vyrovnávání zátěže na N (4 = implicitní hodnota, 6 = maximum).
Router(config-router)#default-information originate	Generuje implicitní cestu do aktualizací RIP.

Komplexní praktické laboratorní cvičení – RIPv2

Použijte příklad pro RIPv1.

1. Na jednotlivých směrovačích zapněte RIPv2.
2. Síť mezi směrovači R2 a R3 (172.16.2.0/24) rozdělte na dvě poloviny (172.16.2.0/25 a 172.16.2.128/25) vložte další směrovač R23 a zprovozněte s RIPv2.
3. Na směrovači na hranici plné třídy (= R3) vypněte automatickou sumarizaci do plné třídy.
4. Nepropagujte RIPv2 do netranzitních sítí tam, kde jsou pouze koncová zařízení.

Příklad konfigurace RIPv2 na R2:

```
!
router rip
 version 2
 passive-interface FastEthernet0/0
 network 172.16.0.0
 network 192.168.2.0
!
```

Dokumentace nastavení

Po dokončení a ověření úlohy pro každý směrovač uložte do textového souboru výstupy následujících příkazů:

- show running-config
- show ip route
- show ip interface brief
- show ip protocols

Kontrolní opakovací otázky a odpovědi (kvíz):

1) Které dva příkazy identifikují, zda je u RIPv2 použita automatická sumarizace?
   a) show running-config
   b) show ip protocols
2) Které tvrzení o RIPv2 je pravdivé?
   a) RIPv2 bude provádět automatickou sumarizaci na hranici hlavní sítě (*major network boundary*) (= hranici sítě v plné třídě)
3) Jaké je implicitní chování protokolu RIP při příjmu a vysílání aktualizací v jednotlivých verzích, pokud není specifikováno žádné číslo verze?
   a) Vysílá aktualizace pouze ve verzi 1, přijímá aktualizace ve verzích 1 i 2.
4) Co by umožnil RIPv2 a nikoliv RIPv1?
   a) Například síť 192.168.0.0/16 (síť má kratší masku než je implicitní pro danou třídu) (maximální počet skoků a možnosti redistribuce cest mají stejné)
5) Čím se liší RIPv2 od RIPv1?
   a) RIPv2 obsahuje ve svých aktualizacích i masku podsítě
6) Na směrovači B, kterému jsou přímo připojené dvě sítě 192.168.1.0/30 a 192.168.1.4/30, se po zapnutí příkazu „**debug ip rip events**“ objevil následující výpis:

```
B#debug ip rip events
- vynecháno -
RIP ignored v2 packet from 192.168.1.1 (illegal version)
RIP ignored v2 packet from 192.168.1.6 (illegal version)
```

   Co je pravděpodobnou příčinou tohoto hlášení?
   a) Tento směrovač B má spuštěnou jinou verzi RIP než oba jeho sousedi. (=> B má spuštěnou verzi 1 a sousední směrovače mají spuštěnou verzi 2 protokolu RIP)
7) Máte tři směrovače A, B a C zapojené v linii za sebou a k nim přilehlé následující čtyři sítě: 192.168.1.32/27, 192.168.1.64/30, 192.168.2.0/24 a 192.168.3.0/24. Na všech třech směrovačích je spuštěn směrovací protokol RIPv2. Proč na směrovači C vidíte pouze síť 192.168.1.0/24 a nikoliv jednotlivé dvě sítě 192.168.1.32/27 a 192.168.1.64/30?
   a) RIPv2 má implicitně spuštěnou automatickou sumarizaci do plné třídy (z důvodu, aby byl v této věci kompatibilní s RIPv1)
8) Na směrovači máte následující výpis: Co z něj lze vyčíst?

```
R2#debug ip rip events
RIP event debugging is on
R2#RIP: received v2 update from 172.16.2.126 on FastEthernet1/0
     172.16.2.128/25 via 0.0.0.0 in 1 hops
     172.16.3.0/24 via 0.0.0.0 in 2 hops
     192.168.3.0/24 via 0.0.0.0 in 2 hops
     192.168.4.0/24 via 0.0.0.0 in 3 hops
```

a) V aktualizacích směrovací protokol předává masky (přijímá RIP verzi 2).

Kapitola 8 - Směrovací tabulka – bližší pohled

V této kapitole se naučíme:

- Popsat jednotlivé typy cest, které je možné nalézt ve směrovací tabulce
- Popsat postup vyhledání cesty do cílové sítě
- Popsat chování procesu směrování ve směrovaných sítích

V předchozích kapitolách jsme prozkoumávali směrovací tabulku pomocí příkazu *show ip route*. Viděli jsme jak jsou přidávány a vymazávány ze směrovací tabulky přímo připojené statické i dynamické cesty.

Pro správce sítě je při odstraňování síťových problémů důležité znát směrovací tabulku do hloubky. Pochopení struktury směrovací tabulky i vyhledávacího procesu v ní vám pomohou diagnostikovat jakékoli problémy směrovací tabulky bez ohledu na Váš stupeň znalosti konkrétního směrovacího protokolu. Například se můžete setkat se situací, kdy jsou ve směrovací tabulce všechny trasy, které byste očekávali že uvidíte, ale paket není očekávaným způsobem přeposílán. Znalost postupu vyhledávání cílové IP adresy pro paket vám umožní určit, zda je paket přeposílán dle očekávání, nebo zda a proč je přeposlán jinam, nebo zda byl zahozen.

V této kapitole se podíváme na směrovací tabulky trochu blíže. První část kapitoly se zaměřuje na strukturu směrovací tabulky Cisco pro IP. Budeme zkoumat formát směrovací tabulky a dozvíme se trasách úrovně 1 a úrovně 2. Druhá část kapitoly analyzuje proces prohledávání směrovací tabulky. Budeme diskutovat třídní směrovací chování, stejně tak jako beztřídní směrovací chování, která používají příkazy *no ip classless* a *ip classless*.

Mnoho podrobností o struktuře a vyhledávacím procesu ve směrovací tabulce IP Cisco bylo z této kapitoly vypuštěno. Máte-li zájem o četbu většího množství informací o tomto tématu a vnitřním fungování Cisco IOS, které se týká směrování, podívejte se do knihy *Cisco IP Routing* od Alexe Zinina. Poznámka: Uvedená kniha ale není knihou pro začátečníky ve směrovacích protokolech, je to důkladné prozkoumání protokolů, procesů a algoritmů používaných operačním systémem Cisco IOS.

Podrobnější pohled na směrování

Terminologie: *route* česky směr, cesta, trasa (překlad kolísá dle aktuálního kontextu).

Směrovací tabulka je databáze, která má hierarchickou strukturu. Důvodem je rychlý postup vyhledávání v ní (*speed lookup process*). Tato struktura má několik úrovní, pro jednoduchost budeme diskutovat pouze úroveň 1 a 2.

Její obsah na směrovači zobrazíme příkazem: **show ip route**

Poznámka: Hierarchie směrovací tabulky v Cisco IOS byla původně implementována s třídním směrovacím schématem. Přestože směrovací tabulka obsahuje oboje třídní i beztřídní adresy, je její celková struktura stále vybudována na tomto třídním schématu.

Ve směrovací tabulce mohou být směry trojího druhu (podle druhu zdroje informací pro tuto řádku):

- přilehlý, přímo připojený (*directly connected*) – kód C,
- statický (*static*) – kód S,
- dynamický (*dynamic*) – kódy R(RIP), D(EIGRP), O(OSPF), ...

Přidání přilehlého, přímo připojeného, směru, cesty (*connected route*) do směrovací tabulky:

- nastavte IP adresu a masku rozhraní,
- administrativně zapněte rozhraní příkazem „no shutddown“,
- přilehlý (přímo připojený) směr (cesta) je okamžitě přidán do směrovací tabulky,
- vyzkoušejte si příkaz „debug ip routing“, abyste to viděli přímo v akci.

Směry úrovně 1 (Level 1 routes)

Úrovně jedna (*Level 1*) je **směr s maskou podsítě (*subnet mask*) rovnou nebo menší než (*equal to or less than*) implicitní maska plné třídy** (*classful mask*)

Trasa úrovně 1 může mít 3 různé typy (funkce):

- 192.168.1.0/24 **síťový směr** (***network route***) (rozumí se do třídní směr). /24 (= *classful mask*). Má implicitní třídní masku. Síťový směr může zároveň být i tzv. Rodičovský směr = úrovně 1 (viz dále).
- 192.168.128.0/20 **nadsíťový směr** (***supernet route***). Má kratší než implicitní třídní masku.
- 0.0.0.0/0 **implicitní směr** (***Default route***). Má masku /0.

Trasa první úrovně je do směrovací tabulky přidána okamžitě po zapnutí rozhraní do přilehlé (přímo připojené) sítě příkazem no shutdown.

Příklad: určete, zda je směr do uvedené sítě **úrovně 1**:

192.168.1.0/24	ano (network route)
192.168.1.32/27	ne
192.168.4.0/22	ano (supernet route)
0.0.0.0/0	ano (default route)

Trasa první úrovně může být ultimátní (ale také být nemusí).

Základní, ultimátní trasa (*Ultimate Route*)

Úroveň 1 může být dále navíc definována jako **základní, nepominutelná trasa (*ultimate route*),** což je směr, který zahrnuje buď:

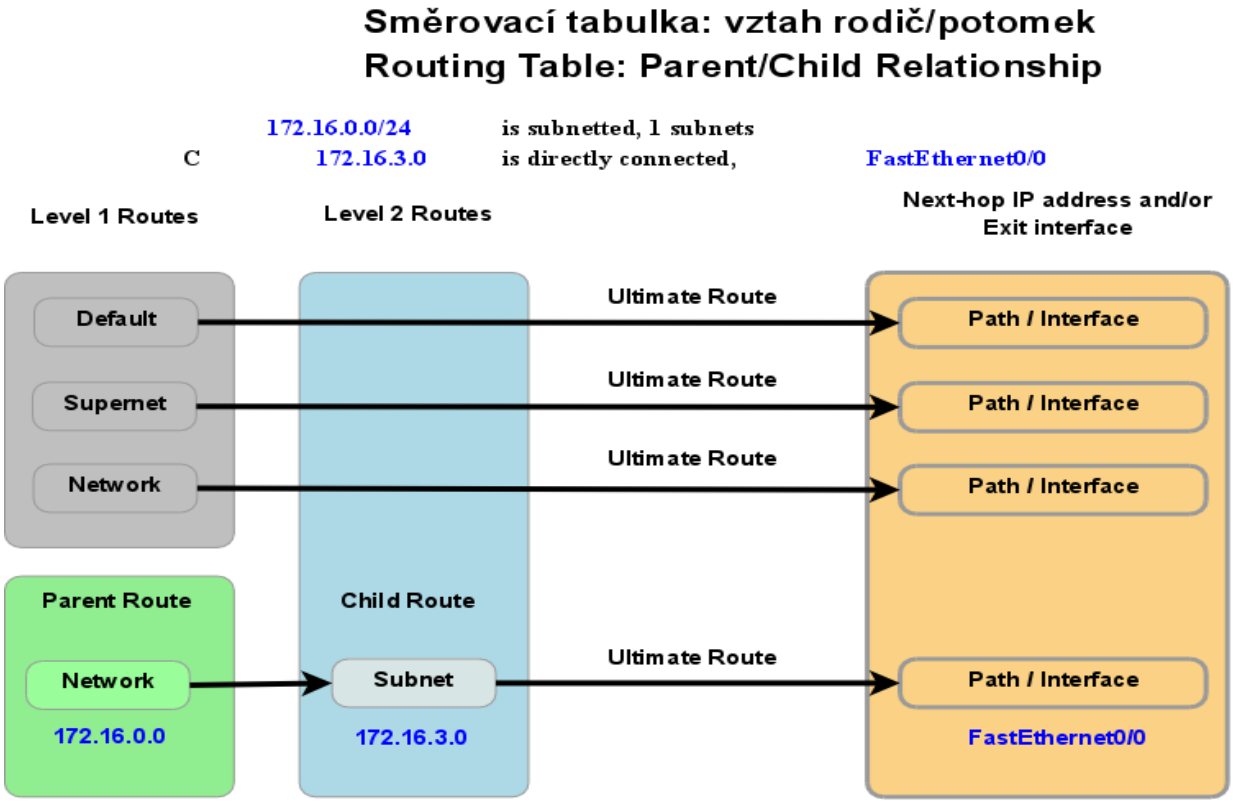
- IP adresu next-hop (jinou cestu),
- a/nebo výstupní, odchozí rozhraní (*exit interface*).

C 192.168.1.0/24 is directly connected, **Serial0/0/1**

Pokud je ve skupině sítí definována podsíť jsou řádky směrovací tabulky navíc ještě rozděleny na typ rodič a potomek.

Rodič a potomek (Parent and child)

- **Trasa typu rodič (*parent route*)** = **trasa úrovně 1 (*level 1 route*)**. Nemá žádnou informaci o výstupním směru (*no exit information*), next-hop nebo odchozí rozhraní. Je automaticky přidán, když je do směrovací tabulky přidána podsíť, tj. když je přidán směr typu potomek (*added when child route is added*).
- **Trasa typu potomek (*child route*)** = **trasa úrovně 2 (*level 2 route*)**. Je to **podsíť** sítě v plné třídě (*subnet of classful network*). Může být také považována za základní, ultimátní, protože obsahuje odchozí rozhraní a/nebo next-hop.



Obsah směrovací tabulky

Příklad (podsíťování třídní sítě, adresní struktura CIDR)

```
      10.0.0.0/30 is subnetted, 2 subnets
R        10.10.10.0 [120/1] via 10.10.10.5, 00:00:28, Serial0/0/0
C        10.10.10.4 is directly connected, Serial0/0/0
```

Směr (Route)	***Rodič (Parent)***	***Potomek (Child)***
10.0.0.0/16	X	
10.10.10.0/30		X
10.10.10.4/30		X

Poznámka: pamatujete, že hierarchie směrovací tabulky v Cisco IOS má třídní směrovací schéma. Směr úrovně 1 je třídní síťová adresa směru podsítě. V tomto případě dokonce i když je zdrojem směru do podsítě beztřídní směrovací protokol.

Výstup z příkazu (Command Output)	***Popis (Description)***
10.0.0.0	Třídní síť (rodičovská)
/30	Maska podsítě **pro směry typu potomek**
is subnetted, 2 subnets	Rodič se dvěma směry typu potomek.
R	Zdrojem směru je RIP
10.10.10.0	První cesta typu potomek
120	Hodnota administrativní vzdálenosti pro cesty jejichž zdrojem je RIP
1	Metrika RIP, 1 hop
via 10.10.10.5	IP adresa Next-hop pro tuto cestu typu potomek
00:00:28	Doba od poslední aktualizace ze souseda
Serial0/0/0	Odchozí rozhraní pro první cestu typu potomek (je *ultimate*)
C	Zdrojem je připojená, přilehlá cesta (*route*)
10.10.10.4	Druhá cesta typu potomek
Serial0/0/0	Odchozí rozhraní pro druhou cestu typu potomek (*ultimate route*)

Příklad (beztřídní adresní struktura VLSM)

```
     172.16.0.0/16 is variably subnetted, 3 subnets, 3 masks
R          172.16.0.0/18 [120/1] via 10.10.10.5, 00:00:28, Serial0/0/0
C          172.16.68.0/22 is directly connected, FastEthernet0/0
C          172.16.72.0/23 is directly connected, FastEthernet0/1
```

Směr (Route)	***Rodič (Parent)***	***Potomek (Child)***
172.16.0.0/16	X	
172.16.0.0/18		X
172.16.68.0/22		X
172.16.72.0/23		X

Bez ohledu na adresní schéma použité sítí (třídní nebo beztřídní), směrovací tabulka bude používat třídní schéma.

Výstup z příkazu (Command Output)	***Popis (Description)***
172.16.0.0	Třídní rodičovský směr (*Classful parent route*)
/16	Třídní maska (*Classful mask*)
is variably subnetted,	Směry potomků mají různé masky
3 subnets, 3 masks	Počet podsítí a masek pro tento rodičovský směr
R	Zdrojem směru je RIP
172.16.0.0	První směr typu potomek
/18	Maska pro první směr typu potomek
120	Hodnota AD směry jejichž zdrojem je RIP
1	Metrika RIP, 1 hop
via 10.10.10.5	IP adresa Next-hop pro první směr typu potomek
00:00:28	Doba od poslední aktualizace ze souseda
Serial0/0/0	Odchozí rozhraní pro první cestu typu potomek (je *ultimate*)
C	Zdrojem je přilehlá cesta (*source is connected route*)
172.16.68.0	Druhá cesta typu potomek
/22	Maska pro druhou cestu typu potomek
FastEthernet0/0	Odchozí rozhraní pro druhou cestu typu potomek

Rozdíl v obsahu směrovací tabulky podle typu sítě

Typ sítě	Pro trasu typu rodič je zobrazena maska v plné třídě	Pro trasu typu rodič je uveden termín „variably subnetted“	Pro trasu typu rodič je uveden počet různých masek pro podřízené řádky typu potomek	V každé řádce typu potomek je maska podsítě
Třídní	Ne	Ne	Ne	Ne
Beztřídní (VLSM)	Ano	Ano	Ano	Ano

Postup vyhledání nejlepšího směru

1. Směrovač prohledá řádky směrovací tabulky se směry úrovně 1, zahrnující třídní směry a nadsíťové směry, pro nejlepší spárování s cílovou adresou IP paketu.

   1.1. Jestliže je nejlepším spárováním ultimátní směr úrovně 1, => plná síť v plné třídě, nadsíť nebo implicitní cesta, je tento směr použit pro přeposlání paketu.

   1.2.Jestliže nejlepší spárování je rodičovský směr úrovně 1, provede se krok 2.

2. Směrovač prohledá směry typu potomek (podsíťové směry) pro příslušný rodičovský směr pro nejlepší spárování.

   2.1.Jestliže je zde odpovídající směr typu potomek úrovně 2, bude tato podsíť použita pro přeposlání paketu.

   2.2.Jestliže zde není žádný odpovídající řádek úrovně 2, provede se krok 3.

3. Má směrovač implementované třídní nebo beztřídní směrování (směrovací chování = způsob prohledávání směrovací tabulky)?

   3.1.Třídní směrování (*classful routing behavior*): pokud je funkční třídní směrování, ukončí se proces vyhledávání a **odhodí paket (bez ohledu na případné nastavení implicitní cesty).**

   3.2.Beztřídní směrování (*classless routing behavior*): pokud je funkční beztřídní směrování, pokračuje hledání nadsíťového směru úrovně 1 ve směrovací tabulce pro spárování, včetně implicitní cesty, pokud je nastavena.

4. Pokud je zde nyní kratší spárování s nadsíťovou nebo implicitní cestou úrovně 1, směrovač použije tento směr pro přeposlání paketu.

5. Jestliže zde **není spárování s žádnou cestou** ve směrovací tabulce, směrovač tento **paket odhodí.**

Později budeme třídní a beztřídní chování směrování diskutovat podrobněji.

Poznámka: směr, který odkazuje pouze na IP adresu následujícího skoku (next-hop) a nemá odchozí rozhraní, musí být převeden na směr s odchozím rozhraním, pomocí rekurzivního vyhledání ve směrovací tabulce.

Poznámka k terminologii: je třeba si uvědomit, že rozlišujeme následující kategorie:

1. systém adresace sítě

1.1. třídní – pouze implicitní masky,

1.2. beztřídní

1.2.1.podsítě pouze sítě v plné třídě, všechny masky stejné - CIDR

1.2.2.podsítě podsítí, masky mohou být různé – VLSM

2. směrovací protokol

2.1. třídní – aktualizace neobsahuje masku

2.2. beztřídní – aktualizace obsahuje masku

3. způsob prohledávání směrovací tabulky, směrovací chování

3.1. třídní – v případě nenalezení přesného spárování u cesty do podsítě typu potomek úrovně 2, je paket odhozen

3.2. beztřídní - v případě nenalezení přesného spárování u cesty do podsítě typu potomek úrovně 2 je dále prohledávána 1. úroveň na supersíť nebo implicitní cestu.

Nejdelší spárování

Nejlepší spárování (*the best match*) termín, který byl použit v předchozím popisu, je také někdy uváděn jako (*a.k.a = also known as*) **nejdelší spárování** (*longest match*).

Cílová IP adresa paketu	172.16.0.10	10101100.00010000.00000000.00001010
Route 1	172.16.0.0/12	10101100.00010000.00000000.00000000
Route 2	172.16.0.0/18	10101100.00010000.00000000.00000000
Route 3	172.16.0.0/26	10101100.00010000.00000000.00000000

Příklad

Použijte obsah části výpisu směrovací tabulky na směrovači C, na kterém je IOS verze 12.3. Směrovač přijal paket s cílovou IP adresou 172.16.1.130. Který nabídnutý směr (cestu) směrovač použije pro přeposlání paketu a proč?

```
C#show ip route
<vynecháno>
Gateway of last resort is 0.0.0.0 to network 0.0.0.0

S    172.16.0.0/13 is directly connected, FastEthernet0/0
     172.16.0.0/16 is variably subnetted, 3 subnets, 2 masks
R       172.16.0.0/24 [120/3] via 172.16.1.1, 00:00:03, FastEthernet0/0
C       172.16.1.0/25 is directly connected, FastEthernet0/0
     172.17.0.0/25 is subnetted, 1 subnets
C       172.17.1.0 is directly connected, FastEthernet0/1
S*   0.0.0.0/0 is directly connected, FastEthernet0/0
C#
```

a) 172.16.1.0/25

b) 172.16.0.0/16

c) 172.16.0.0/24

d) 172.16.0.0/13

e) implicitní směr (cesta)

f) nic, paket bude zahozen

Jak postupovat při řešení:

Protože je použita verze IOS 12.3, je tam implicitní nastavení beztřídního způsobu prohledávání směrovací tabulky *ip classless*. Prohledávají se nejprve rodičovské směry a to podle délky masky sestupně. Nejprve se prohledávají potomci rodičovského směru 172.16.0.0/16 a protože žádný nevyhovuje, začnou se prohledávat další rodičovské směry s kratší maskou. Vyhovuje potom ultimátní cesta 172.16.0.0/13 (d).

Pokud by bylo dodatečně nastaven třídní způsob prohledávání směrovací tabulky *no ip classless*, tak by byl daný paket zahozen (protože by se v rámci rodičovského směru 172.16.0.0/16 nenašel odpovídající potomek) a prohledávání by se v tomto případě již nevracelo na rodičovskou úroveň.

Příkazy pro kapitolu 8, Směrovací tabulka – bližší pohled

Příkaz (Command)	***Popis (Description)***
Router(config)# no ip classless	Nastavuje chování směrovače při prohledávání směrovací tabulky jako třídní (*classful*). Bylo implicitní v IOS ve verzi před Release 11.3.
Router(config)# ip classless	Nastavuje chování směrovače při prohledávání směrovací tabulky jako beztřídní (*classless*). Je implicitní v IOS ve verzi Release 11.3 a pozdější.

Kontrolní opakovací otázky a odpovědi (kvíz):

1) Které charakteristiky mohou být požity k určení, zda je směr ultimátní?

   a) Směr obsahuje odchozí rozhraní (*exit interface*).

2) Směrovač R1 je nakonfigurován s příkazy: R1(config)#**ip classless** a R1(config)#**ip route 0.0.0.0 0.0.0.0 serial0/0/0**. Co udělá R1 s paketem, který bude spárovaný s rodičovským směrem, ale v rámci něho už nebude spárovaný se žádným směrem typu potomek?

   a) Pošle paket přes implicitní cestu (implicitním směrem).

3) Která akce na směrovači umožní třídní chování při směrování (způsob prohledávání směrovací tabulky)?

   a) Vydání příkazu **no ip classless**.

4) Během procesu hledání směru při směrování, co ustavuje preferovaný směr?
   a) Nejdelší spárování bitů zleva.
5) Jestliže je paket spárovaný s (= odpovídá) cestou 1. úrovně typu rodič, co je v procesu směrování druhý krok?
   a) Směrovač bude hledat směr 2. úrovně typu potomek s odchozím rozhraním.
6) Co dělají příkazy **ip classsles** a **no ip classless**?
   a) Určují beztřídní nebo třídní způsob prohledávání směrovací tabulky během směrování.
7) Směrovač R1 je nakonfigurován s příkazy: R1(config)#**no ip classless** a R1(config)#**ip route 0.0.0.0 0.0.0.0 serial0/0/0**. Co udělá R1 s paketem, který bude spárovaný s rodičovským směrem, ale v rámci něho už nebude spárovaný se žádným směrem typu potomek?
   a) Zahodí paket.

Kapitola 9 - Protokol EIGRP

V této kapitole se naučíme:

- Popsat předchůdce a historii EIGRP
- Popsat funkce a činnost EIGRP
- Prozkoumat základní konfigurační příkazy EIGRP a určit jejich účel
- Vypočítat složenou metriku používanou EIGRP
- Popsat koncept a činnost konvergenčního algoritmu DUAL
- Popsat použití dalších konfiguračních příkazů EIGRP

EIGRP (*Enhanced Interior Gateway Routing Protocol*) je směrovací protokol typu vektor vzdálenosti, beztřídní směrovací protokol, který byl uvolněn v roce 1992 spolu s IOS 9.21. Jak už jeho název napovídá, EIGRP, je vylepšení protokolu Cisco IGRP (*Interior Gateway Routing Protocol*). Oba dva jsou proprietární protokoly Cisco a pracují pouze na směrovačích Cisco.

Hlavním cílem společnosti Cisco při vývoji EIGRP bylo vytvořit beztřídní verzi IGRP. EIGRP obsahuje několik funkcí, které se běžně nevyskytují v jiných směrovacích protokolech typu vektor vzdálenosti jako jsou RIP (RIPv1 a RIPv2) a IGRP. Mezi tyto funkce patří:

- spolehlivý transportní (L4) protokol RTP (*Reliable Transport Protocol*),
- omezené aktualizace,
- konvergentní algoritmus DUAL (*Diffusing Update Algorithm*),
- vytváření vztahů sousedství (*adjacencies*),
- tabulky Sousedů (*neighbor*) a Topologickou (*topology*).

Přestože EIGRP může působit jako směrovací protokol typu stavu linky, je to stále ještě směrovací protokol typu vektor vzdálenosti.

Poznámka: Pro definici EIGRP je někdy používáno termínu *hybridní směrovací protokol*. Nicméně tento termín je zavádějící, protože EIGRP není kříženec mezi směrovacími protokoly typu vektor vzdálenosti a typu stav linky - je to je pouze směrovací protokol typu vektor vzdálenosti. Proto společnost Cisco při odkazu na EIGRP již tento termín nadále nepoužívá.

V této kapitole se dozvíte, jak nastavit EIGRP a ověřit si konfiguraci s novými příkazy show. Naučíte se také vzorec, který EIGRP používá pro výpočet složené (kompozitní) metriky.

Jedinečný pro EIGRP je jeho spolehlivý transportní protokol RTP (*Reliable Transport Protocol*), který poskytuje spolehlivé i nespolehlivé doručování paketů EIGRP. Kromě toho, EIGRP vytváří vztahy sousedství (*adajcency*) s přímo připojenými směrovači, které mají též spuštěný EIGRP. Sousedské vztahy se používají ke sledování stavu těchto sousedů. RTP a sledování vztahů sousedství (*adjacencies*) připravují půdu pro tahouna EIGRP – algoritmus DUAL (*Diffusing Update Algorithm*).

Vzhledem k tomu, že výpočetní motor, který pohání EIGRP, DUAL sídlí v samotném centru směrovacího protokolu, zaručuje to v celé směrovací doméně cesty bez smyček a záložní cesty. Naučíte se, jak přesně DUAL zvolí trasy k instalaci do směrovací tabulky, a to, co DUAL dělá s potenciální-

mi záložními trasami.

Stejně jako RIPv2, EIGRP může pracovat s třídním nebo beztřídním chováním směrování. Naučíte se, jak vypnout automatické sumarizace a pak, jak ručně sumarizovat sítě, aby se zmenšila velikost směrovacích tabulek. Nakonec se naučíte, jak používat implicitní směrování s EIGRP.

Úvod do EIGRP

EIGRP – vylepšený protokol typu vektor vzdálenosti

Ačkoli je EIGRP popisován jako vylepšení směrovacího protokolu typu vektor vzdálenosti, je to stále ještě směrovací protokol typu vektor vzdálenosti. To někdy může být zdrojem nejasností. Abychom ocenili vylepšení EIGRP a odstranili jakékoliv nedorozumění, musíme se nejprve podívat na jeho předchůdce, IGRP.

Kořeny EIGRP: IGRP

Společnost Cisco vyvinula svůj proprietární protokol IGRP v roce 1985 v reakci na některá omezení RIPv1 zahrnující použití počtu přeskoků jako metriky a maximální velikosti sítě 15 přeskoků.

Místo počtu přeskoků používají jak IGRP tak i EIGRP jako (složenou, kompozitní) metriku šířku pásma, zpoždění, spolehlivost a zatížení. Ve se výchozím nastavení oba směrovací protokoly používají pouze šířky pásma a zpoždění. Nicméně, protože IGRP je třídní směrovací protokol, který používá Bellman-Fordův algoritmus a periodické aktualizace, je jeho využitelnost v mnoha dnešních sítí omezená.

Proto společnost Cisco vylepšila IGRP s novým algoritmem DUAL a dalšími funkcemi. Příkazy pro IGRP i EIGRP jsou podobné a v mnoha případech totožné. To umožňuje snadnou migraci z IGRP na EIGRP. Společnost Cisco přerušila podporu IGRP počínaje IOS verze 12.2(13)T a 12.2(R1s4)S.

Přestože budou podrobněji popsány v celé této kapitole, dovolte probrat některé z rozdílů mezi tradičním směrovacím protokolem typu vektor vzdálenosti jako je RIP i IGRP a mezi vylepšeným směrovacím protokolem typu vektor vzdálenosti EIGRP.

Následující tabulka shrnuje hlavní rozdíly mezi tradičním směrovacím protokolem typu vektor vzdálenosti jako je RIP a a mezi vylepšeným směrovacím protokolem typu vektor vzdálenosti EIGRP.

Přehled činnosti směrovacích protokolů	
Tradiční protokol typu vektor vzdálenosti	***Vylepšený protokol - EIGRP***
Používá algoritmus Bellman-Ford neboli Ford-Fulkerson	Používá rozprostřený aktualizační algoritmus *Diffusing Update Algorithm* (DUAL)
Sleduje stáří záznamů ve směrovací tabulce a používá periodické aktualizace	Nesleduje stáří záznamů (*not age out*) ve směrovací tabulce a nepoužívá periodické aktualizace
Eviduje pouze nejlepší trasy, nejlepší cesty do cílové sítě	Odděleně udržuje tabulku topologie (*topology table*) obsahující nejlepší trasu a všechny záložní cesty neobsahující smyčky (*loop free backup*

	paths), nezávisle na směrovací tabulce
Když se směr stane nedostupným, musí směrovač počkat na novou aktualizaci	Když se směr stane nedostupným, DUAL použije záložní cestu pokud existuje v topologické tabulce
Pomalejší konvergence z důvodu použití zadržovacích časovačů	Rychlejší konvergence z důvodu nepoužití zadržovacích časovačů a použití systému koordinovaných výpočtů tras (*coordinated route calculations*)

Algoritmus

Všechny tradiční směrovací protokoly typu vektor vzdálenosti používají některou variantu algoritmu Bellman-Ford či Ford-Fulkerson. Tyto protokoly, jako jsou RIP a IGRP, sledují stáří jednotlivých řádek směrovací tabulky a proto je nutné pravidelně posílat aktualizace směrovací tabulky.

EIGRP používá aktualizační algoritmus **DUAL (*Diffusing Update Algorithm*)**. Ačkoli je EIGRP stále ještě směrovacím protokolem typu vektor vzdálenosti, implementuje s algoritmem DUAL funkce, které nejsou v tradičních směrovacích protokolech typu vektor vzdálenosti. EIGRP neposílá pravidelné aktualizace a nesleduje stáří řádky tras ve směrovací tabulce. Místo toho EIGRP používá jednoduchý protokol Hello pro monitorování stavu spojení se svými sousedy. Pouze změny ve směrovací informace, jako je nová linka nebo že se linka stala nedostupnou, způsobí, že nastane aktualizace. Směrovací aktualizace EIGRP jsou stále vektory vzdáleností předávané přímo připojeným sousedům.

Stanovení cesty

Tradiční směrovací protokoly typu vektor vzdálenosti jako RIP a IGRP sledují pouze preferované trasy, nejlepší cestu k cílové síti. Pokud přestane být tato trasa k dispozici, směrovač čeká na další směrovací aktualizaci s cestou k této vzdálené síti.

Algoritmus DUAL v EIGRP udržuje odděleně od směrovací tabulky tabulku topologie, která obsahuje jak nejlepší cestu k cílové síti tak všechny záložní cesty, které DUAL určil jako neobsahující smyčky (*loop-free*). *Loop-free* znamená, že soused nemá cestu do cílové sítě, která prochází přes tento router.

Později v této kapitole uvidíte, že trasa, která bude algoritmem DUAL považována za platnou záložní cestu bez smyček, musí splňovat požadavek známý jako podmínka proveditelnosti. Jakákoli záložní cesta, která splňuje tuto podmínku má zaručeno, že je bez smyček (*loop-free*). Vzhledem k tomu, že EIGRP je směrovací protokol typu vektor vzdálenosti, je možné, že mohou existovat záložní cesty k cílové síti neobsahující smyčky, které nesplňují podmínku proveditelnosti. Tyto cesty proto nejsou zahrnuty v tabulce topologie jako platná záložní cesta bez smyček určená algoritmem DUAL.

Jestliže se trasa stane nedostupnou, bude DUAL hledat ve své topologické tabulce platnou záložní cestu. Pokud existuje, tak se tato trasa okamžitě zapíše do směrovací tabulky. V případě že neexistuje, DUAL provádí proces zjišťování sítí, zda tam náhodou není záložní cesta, která nesplňují požadavek podmínky proveditelnosti. Tento proces se diskutuje důkladněji později v této kapitole.

Konvergence

Tradiční směrovací protokoly typu vektor vzdálenosti jako RIP a IGRP používají periodické aktualizace. Vzhledem k nespolehlivé povaze periodických aktualizací, jsou tradiční směrovací protokoly typu vektor vzdálenosti náchylné ke směrovacím smyčkám a počítání do nekonečna. RIP a IGRP využívají několik mechanismů, které pomáhají vyhnout se těmto problémům, včetně zadržovacích časovačů, které způsobují dlouhé doby konvergence.

EIGRP nepoužívá zadržovací časovače. Místo toho je cest bez smyček dosaženo prostřednictvím systému výpočtů trasy (rozptylové výpočty), které jsou vykonávány koordinovaným způsobem mezi směrovači. Detail toho, jak se to provádí, je nad rámec tohoto kurzu, ale výsledkem je rychlejší konvergence než u tradičních směrovacích protokolů typu vektor vzdálenosti.

Formát zprávy EIGRP

Poznámka: V následující diskusi zpráv EIGRP je mnoho políček jdoucích nad rámec tohoto kurzu. Jsou zobrazena všechna pole, aby se poskytl přesný obraz formátu zprávy EIGRP. Avšak jsou diskutována pouze pole relevantní pro uchazeče CCNA.

Každá zpráva EIGRP obsahuje záhlaví. Důležitá pro naši diskusi jsou políčko Opcode a políčko číslo autonomního systému. Opcode specifikuje typ paketu EIGRP:

- Aktualizace
- Dotaz
- Odpověď na dotaz
- Kontaktní paket

Číslo Autonomního systému (AS) určuje proces směrování EIGRP. Na rozdíl od RIP mohou směrovače Cisco provozovat více instancí EIGRP. Číslo AS slouží k odlišení vícero instancí EIGRP od sebe.

Zapouzdření zprávy protokolu EIGRP

<table>
<tr><td>Záhlaví linkové vrstvy</td><td>Záhlaví paketu IP</td><td>Záhlaví paketu EIGRP</td><td>Typy Type/Length/Value (TLV)</td></tr>
<tr><td colspan="4">Rámec linkové vrstvy
Zdrojová MAC adresa = adresa vysílajícího rozhraní
Cílová MAC adresa = Multicast: 01-00-5E-00-00-0A</td></tr>
<tr><td></td><td colspan="3">Paket IP
Zdrojová IP adresa = adresa vysílajícího rozhraní
Cílová IP adresa = Multicast: 224.0.0.10
Protokol = 88 pro EIGRP</td></tr>
<tr><td colspan="2"></td><td colspan="2">Záhlaví paketu EIGRP
Opcode pro typ paketu EIGRP
Číslo Autonomního systému (AS)</td></tr>
<tr><td colspan="3"></td><td>Typy TVL (pouze výběrový seznam):
0x0001 - Parametry EIGRP
0x0102 - IP trasy interní
0x0103 - IP trasy externí</td></tr>
</table>

EIGRP obsahuje několik funkcí, které běžně nejsou k nalezení u jiných směrovacích protokolů **typu vektor vzdálenosti** jako je RIP (RIPv1 a RIPv2) a IGRP. Tyto funkce zahrnují:

- Spolehlivý transportní protokol (L4) *Reliable Transport Protocol (RTP)* – potvrzovaná i nepotvrzovaná (datagramová) služba na transportní vrstvě.
- Částečné omezené aktualizace (*Partial Bounded Updates)* – aktualizace obsahují pouze změny topologie a jsou zasílány pouze směrovačům, kterých se týkají,
- Difuzní algoritmus aktualizací - *Diffusing Update Algorithm* (DUAL) – umožňuje mít připravenou předem vypočtenou záložní cestu při výpadku linky bez čekání na další aktualizaci => **rychlá konvergence (= synchronizace směrovacích tabulek do konzistentního stavu)**
- Vytváření vztahů sousedství (*Establishing Adjacencies*) mezi přilehlými směrovači ve stejné směrovací doméně (AS).
- Tabulka sousedů a tabulka topologie (*Neighbor and Topology Tables).* Tabulka topologie obsahuje tzv. přípustné následníky *(feasible successors)* = záložní cesty. Tabulka sousedů obsahuje přilehlé směrovače ve stavu sousedství.

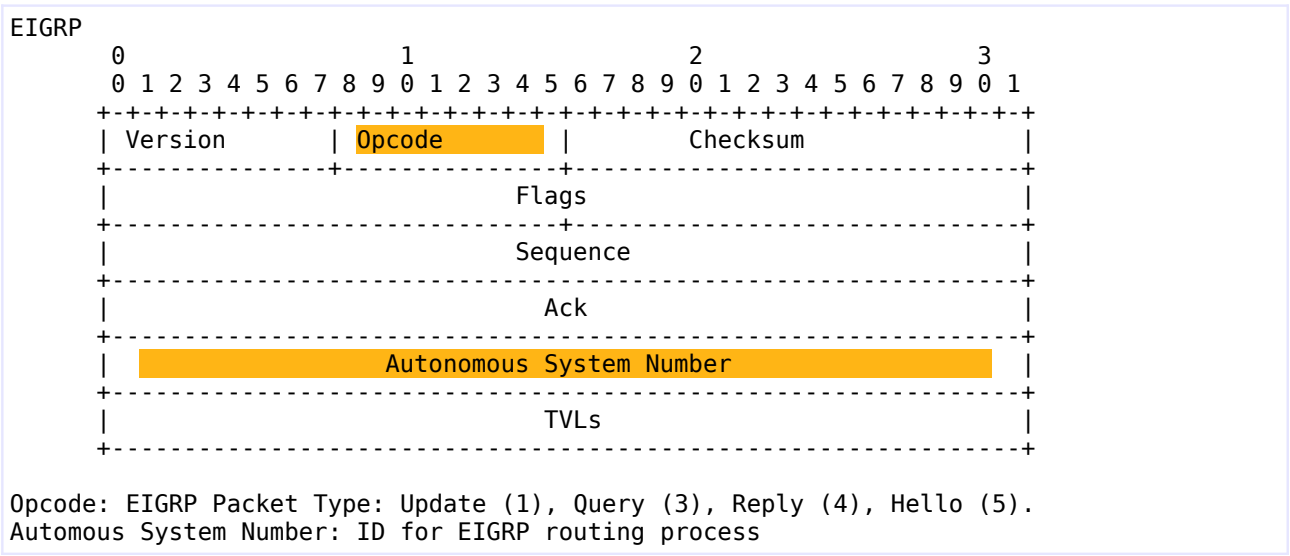
Přestože EIGRP může působit dojmem jako směrovací protokol typ stav linky, je to stále směrovací protokol typu vektor vzdálenosti.

Poznámka: Pro definici EIGRP býval někdy použit termín hybridní směrovací protokol. Nicméně tento termín je matoucí, protože EIGRP je výhradně protokol typu vektor vzdálenosti. Z tohoto důvodu Cisco již nadále nepoužívá tento termín v odkazu na EIGRP.

EIGRP používá a udržuje pro svoji činnost **3 tabulky**:

- **směrovací** (*routing*) – obsahuje pouze nejlepší cesty (*successor*) (jednu nebo několik se stejnou nejnižší metrikou (*feasible distance*)) do cílové sítě použité pro směrování => v algoritmu DUAL jsou tzv. *Successor route* - primární cesty (*primary route*) vybrané pomocí DUAL pro směrování – zařazení do směrovací tabulky. Její obsah je určen pomocí DUAL z následujících dvou tabulek:
- **topologie** (*topology*) – obsahuje všechny zjištěné (naučené) směry (nejlepší směr (*successor route*), záložní směr (*feasible successor route*) i všechny ostatní) do všech cílových sít (obsahuje tedy celou topologie sítě ve stejné směrovací doméně),
- **sousedů** (*neighbor*) – obsahuje sousední směrovače, kteří si vzájemně vyměňují aktualizace v EIGRP (směrovače jsou **ve vztahu přilehlého sousedství** (*adjacent routers*) na přímo připojené (přilehlé) síti ve stejném autonomním systému (AS)). (Směrovač má své informace včetně hodnoty metriky trasy pouze od přilehlých sousedů, proto je to směrovací protokolu používající algoritmus vektoru vzdálenosti.)

Přesný formát těchto tabulek je závislý na **směrovaném protokolu** a je včetně jejich obsahu veden odděleně (pro směrované protokoly L3: IP, IPX, AppleTalk) = tzv. modul závislý na protokolu (*Protocol Dependent Module, PDM*).



Typy paketů EIGRP (typ je určen hodnotou pole *Opcode*):

- Aktualizace (*Update)* – obsahují pouze změny, nejsou periodické, vysílané unicast/multicast (podle počtu adresátů), potvrzované. Aktualizace jsou:
  - vázané, omezené (*bounded*) – aktualizace jsou posílány (propagovány) pouze na směrovače, na které má tato změna vliv,
  - částečné (*partial*) – aktualizace obsahují pouze změny topologie (týká se to též změny metriky).
- Dotaz (*Query*) - hledání sítí, další úkoly, unicast nebo multicast, potvrzovaná,
- Odpověď na dotaz (*Reply*) - odpověď, vždy unicast, potvrzovaná,
- Kontaktní paket (*Hello)* - hledání, identifikace a verifikace sousedních směrovačů (EIGRP ve stejném autonomním systému), multicast, datagram (periodické – 5 sekund u Ethernetu). (Protože aktualizace nejsou úplné (= nikoliv celá směrovací tabulka) a neposílají se všem směrovačům, musí být pro kontrolu toho že všechny směrovače jsou „naživu“ vytvořen a udržován **vztah sousedství mezi směrovači** *(adjacency)*, které si vyměňují informace, vztah sousedství se vytváří a udržuje právě pomocí těchto kontaktních paketů.)

Administrativní vzdálenosti

- Interní EIGRP = 90,
- EIGRP agregovaný směr (*summary route*) = 5,
- External EIGRP (redistribuce z jiných směrovacích protokolů nebo z EIGRP v jiném autonomním systému) = 170.

Metrika

Kompozitní (*composite*), složená, metrika u EIGRP:

Default metric = [K1*bandwidth[20] + K3*delay] (implicitní formule)

Metric = [K1*bandwidth + (K2*bandwidth)/(256 – load) + **K3*delay]** * [K5/(reliability + K4)] (kompletní formule)

Při vypočtu číselné hodnoty metriky se použijí následující hodnoty:

referenční šířka pásma = bandwidth = 256 * (10 000 000 /nejnižší šířka pásma na trase do cíle),

delay = 256 * (součet zpoždění na cestě do cíle) / 10.

Nejlepší cesta (s nejmenší metrikou, *feasible distance*) je ta s největší šířkou pásma a s nejmenším zpožděním.

Implicitní hodnoty K:

1. K1 (*bandwidth*) = 1
2. K2 (*load*) = 0
3. K3 (*delay*) = 1
4. K4 (*reliability*) = 0
5. K5 (*reliability*) = 0

Aktuální hodnoty K zobrazí příkaz:

show ip protocols

Změna hodnot K:

```
Router(config-router)#metric weights tos k1 k2 k3 k4 k5
```

tos = *type of service* je vždy nastavena na 1.

```
R2#show ip protocols

Routing Protocol is "eigrp  100 "
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100[21]
```

20 Do vzorce se automaticky použije relativní referenční šířka pásma nejpomalejší linky na trase do cílové sítě. Bandwidth = 256* 10 000 000 / bandwidth. (Nejlepší cesta je cesta s nejmenší hodnotou metriky.)

21 Implicitní maximální počet přeskoků v EIGRP je roven 100 a lze ho nastavit až na maximálně 220 přeskoků.

Aktuální hodnoty vah metrik EIGRP na konkrétním rozhraní zobrazí:

show interface

```
R2#sh int fa0/0
FastEthernet0/0 is up, line protocol is up (connected)
  Hardware is Lance, address is 0060.2f37.725b (bia 0060.2f37.725b)
  Internet address is 192.168.2.254/24
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec, rely 255/255, load 1/255
  Encapsulation ARPA, loopback not set
```

- Metrika **přenosová rychlost (přenosová kapacita)** (*bandwidth*) je zobrazena v Kbit (kilobitech). *Většina sériových rozhraní používá implicitní hodnotu 1,544,000 bps, což je hodnota pro připojení typu T1.* Nastavená hodnota může a také nemusí odrážet skutečnou přenosovou rychlost rozhraní. Můžete ji nastavit v konfiguračním režimu rozhraní.
- **Zpoždění** (*delay*) je měřítkem doby potřebné pro cestu paketu přes daný směr (*route*). Je to statická hodnota vyjádřená v mikrosekundách (μsec ve výpisech usec). Pro FastEtherenet je to 100 μsec. Pro T1 je to 20 000 μsec.
- **Spolehlivost** (*reliability, rely*) je měřítkem pravděpodobnosti (*probability*), že linka selže, nebo jak často se na lince vyskytují chyby. Na rozdíl od zpoždění je spolehlivost měřena dynamicky s hodnotou mezi 0 a 255, kde 1 je minimálně spolehlivá linka a 255 je 100% spolehlivá. Je počítána jako průměr za 5 minut, aby se předešlo vlivům náhlých změn četnosti chyb.
- **Zatížení** (*load*) odráží využití linky síťovým provozem. Zatížení je měřeno dynamicky s hodnotami mezi 0 a 255. Je žádanější nižší hodnota, která indikuje méně zatíženou linku.

Konvergenční algoritmus DUAL

Koncepce algoritmu DUAL

DUAL (*Diffusing Update Algorithm*) je algoritmus používaný EIGRP pro dosažení (primární) nejlepší cesty neobsahující smyčky a dalších záložních cest neobsahujících smyčky (*the best loop-free path and loop-free backup paths*), má rychlou konvergenci – protože záložní cesty má napočítány dopředu a potřebuje malou šířku pásma – používá omezené a částečné aktualizace.

DUAL používá několik termínů:

- Následník (*Successor*) - sousední směrovač na cestě, přes který bude přeposílán (*forward*) paket (nejnižší metrika)
- Přípustná vzdálenost (*Feasible Distance (FD*) - nejnižší metrika do cílové sítě (je ve směrovací tabulce aktuálního směrovače i v tabulce síťové topologie)
- Přípustný následník (*Feasible Successor (FS)*) - soused, který má cestu k cíli neobsahující smyčky (*loop-free*), musí splnit podmínku přípustnosti (*feasibility condition),*

- Inzerovaná vzdálenost - *Reported Distance (RD)* neboli *Advertised Distance (AD)* - vzdálenost souseda k cíli, kterou hlásí soused aktuálnímu směrovači
- Podmínka přípustnosti (*Feasible Condition* neboli *Feasibility Condition (FC)*) - je splněna, pokud sousedova *reported distance* (tj. vzdálenost souseda k cíli, kterou mi hlásí) je menší než moje *Feasible Distance*. Pokud není k dispozici *Feasible Successor* (nesplňuje podmínku přípustnosti), musí se přepočítat celý DUAL

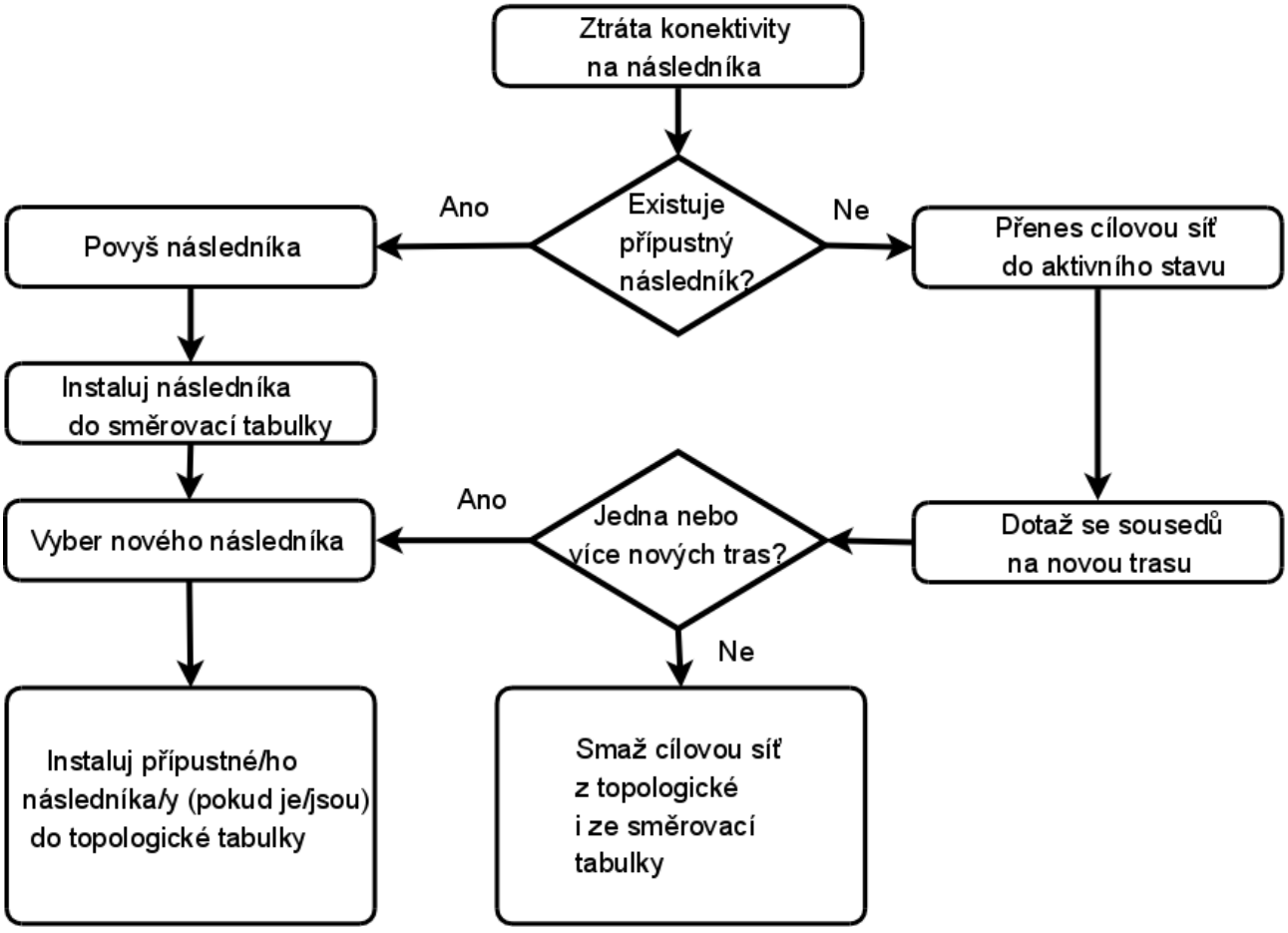
Tyto termíny a koncepty jsou centrem mechanismu předcházení směrovacím smyčkám.

Konečný automat

Konečný automat (*Finite State Machine*, *FSM*) – je abstraktní automat, nikoliv mechanické zařízení s pohyblivými součástmi. Konečný automat (*FSM*) definuje množinu možných stavů, které někdy mohou nastat, a jaké události jsou příčinou těchto stavů a jaké události jsou důsledkem těchto stavů. (Na rozdíl od logického obvodu, kde výstupní stav záleží na okamžitém vstupním stavu, výstup konečného automatu závisí na celé posloupnosti vstupních stavů.)

Vývojáři používají konečné automaty k popisu jak budou zařízení, počítačové programy nebo směrovací algoritmy, reagovat na určitou konkrétní sadu vstupních událostí.

Konečný automat algoritmu DUAL (DUAL Finite State Machine)



=> U EIGRP (a stejně potom i u OSPF) stav směrování závisí i na postupně provedených změnách nastavení (protože výpočetní algoritmus je konečný automat (FSM). Někdy je tedy nutné, po změnách konfigurace, vymazat tabulky ukládající průběžné stavy. (Resetovat procesy příslušného směrovacího protokolu nebo restartovat směrovače.)

Autonomní systém

Autonomní systém (*Autonomous System*, *AS*) neboli směrovací doména je oblast ve které jsou nastaveny stejné zásady směrování do Internetu. Je mu přiděleno 16-ti bitové číslo (0 – 65535). Při konfiguraci EIGRP musí být číslo AS zadáno. Pokud mají směrovací procesy EIGRP jiné číslo AS nekomunikují spolu (pokud není mezi nimi nastavena redistribuce cest). Z tohoto pohledu je tedy vlastně AS číslo, identifikátor, procesu (*process ID*).

Příkazy pro kapitolu 9, EIGRP

Konfigurace EIGRP

Router(config)#router eigrp 100	Zapne proces EIGRP. 100 je číslo autonomního systému, což může být číslo mezi 1 a 65 535.
	Všechny směrovače v tom samém autonomním systému musí používat stejné číslo autonomního systému.
Router(config-router)#network 10.0.0.0	Specifikuje, která síť je inzerována pomocí EIGRP.
Router(config-if)#bandwidth x	Nastaví šířku pásma (přenosovou rychlost, kapacitu) tohoto rozhraní na x kilobitů, což EIGRP umožní lepší kalkulaci metriky.
	TIP: Příkaz bandwidth je použit pouze pro výpočet metriky. Nemění skutečný výkon rozhraní.
Router(config-router)#no network 10.0.0.0	Vymaže zadanou síť ze zpracování EIGRP.
Router(config)#no router eigrp 100	Vypne směrovací proces 100.
Router(config-router)#network 10.0.0.0 0.255.255.255	Identifikuje, která rozhraní nebo sítě jsou zahrnuty do EIGRP. Rozhraní musí být nakonfigurována a adresami, které spadají do rozsahu určeného pseudomaskou v příkazu network. Maska sítě zde lze také použít.
Router(config-router)#metric weights tos k1 k2 k3 k4 k5	Změní implicitní hodnoty *k*, použité při výpočtu metriky. Toto jsou implicitní hodnoty: tos=0,

	k1=1, k2=0, k3=1, k4=0, k5=0

POZNÁMKA: Klíčové slovo tos (*type of service*) je odkaz na původní protokol IGRP, zamýšlející směrování podle typu služby. Protože to ale nebylo nikdy zavedeno do praxe je pole tos v tomto příkaze **vždy** nastaveno na nulu (0).

POZNÁMKA: S implicitním nastavením je metrika EIGRP redukována na nejpomalejší šířku pásma plus součet všech zpoždění odchozích rozhraní z lokálního směrovače do cílové sítě.

TIP: Aby mohly dva směrovače zformovat vztah sousedství v EIGRP, musí jim vzájemně souhlasit hodnoty k.

UPOZORNĚNÍ: Bez toho aniž byste byli opravdu velmi dobře obeznámeni s tím, co se děje ve vaší síti, doporučuje se neměnit hodnoty k.

Automatická a manuální sumarizace v EIGRP

Router(config-router)#auto-summary	Zapne automatickou sumarizaci v EIGRP. POZNÁMKA: Implicitní chování automatické sumarizace je změněno z povoleno na nepovoleno od verze IOS 12.2(8)T.
Router(config-router)#no auto- summary	Vypne automatickou sumarizaci.
	POZNÁMKA: Chování automatické sumarizace je implicitně vypnuto, počínaje od IOS 12.2.(8)T. To znamená, že IOS nyní posílá směrovací informace o podsítích i mimo hranice sítě v plné třídě (nadsítě).
Router(config)#interface fastethernet 0/0	Vstup do konfiguračního režimu rozhraní.
Router(config-if)#ip summary-address eigrp 100 10.10.0.0 255.255.0.0 75	Zapne manuální sumarizaci pro autonomní systém 100 v EIGRP na tomto konkrétním rozhraní pro zadanou síť a masku. Administrativní vzdálenost pro tento sumarizovaný směr je nastavena na 75.
	POZNÁMKA: Argument administrativní vzdálenost je v tomto případě nepovinný. Bez něho je na sumarizovaný směr automaticky použita hodnota 5.

VAROVÁNÍ: EIGRP automaticky sumarizuje sítě na hranicích plné třídy. Špatně navržená síť s nesouvislými podsítěmi může mít problémy s konektivitou, jestliže je funkce sumarizace ponechána zapnutá. Například: jestliže by dva směrovače inzerovaly stejnou síť 172.16.0.0/16, když by ve skutečnosti bylo třeba, aby inzerovaly dvě různé sítě 172.16.10.0/24 a 172.16.20.0/24. Doporučená praxe je, abyste vypnuli automatickou sumarizaci a použili příkaz ip summary-address a sumarizovali manuálně to, co je potřeba.

Vyvažovaní zátěže: variance (variace)

Router(config)#router eigrp 100	Vytvoří směrovací proces 100.
Router(config-router)#network 10.0.0.0	Určuje, která síť je inzerována v EIGRP.
Router(config-router)#variance n	Dá pokyn směrovači, aby zahrnul směry s metrikou menší nebo rovnou n-krát minimální metrice směru pro daný cíl. N je číslo specifikované pomocí příkazu variance.

POZNÁMKA: Jestliže cesta není přípustný následník (*feasible successor*), není použita ve vyvažování zátěže.

POZNÁMKA: EIGRP podporuje vyvažování zátěže až šesti cest s nestejnou cenou (metrikou).

Použití příkazu Bandwidth

Router(config)#interface serial 0/0	Vstup do konfiguračního režimu rozhraní.
Router(config-if)#bandwidth 256	Nastaví šířku pásma, přenosovou kapacitu (*bandwidth*) na 256 kilobitů, aby tak umožnilo lepší kalkulaci metriky v EIGRP.
Router(config-if)#ip bandwidth-percent eigrp 50 100	Nastaví procento přenosové kapacity - šířky pásma (*bandwidth*), které může být EIGRP použito na tomto rozhraní pro výměnu směrovacích informací. 50 je číslo autonomního systému EIGRP. 100 je hodnota procenta. 100% * 256 = 256 kb/s.

POZNÁMKA: Implicitně je EIGRP nastaveno pouze na 50 procent šířky pásma rozhraní pro výměnu směrovacích informací. Mohou být nastaveny větší hodnoty než je 100 procent. Takové nastavení může být užitečné jestliže je *bandwidth* z jiných důvodů nastaven uměle nízký (jako je manipulace se směrovací metrikou).

POZNÁMKA: Příkaz *ip bandwidth-percent* se spoléhá na hodnotu nastavenou příkazem *bandwidth*.

Autentizace

Router(config)#interface serial 0/0	Vstup do konfiguračního režimu rozhraní.
Router(config-if)#ip authentication mode eigrp 100 md5	Zapne na tomto rozhraní v autentizaci EIGRP paketů hašovací algoritmus MD5 (*Message Digest 5*).
Router(config-if)#ip authentication key-chaine-	Zapne na tomto rozhraní autentizaci EIGRP pa-

igrp 100 romeo	ketů. Romeo je je jméno pojmenované skupiny klíčů (*key chain*).
Router(config-if)#exit	Návrat do globálního konfiguračního režimu.
Router(config)#key chain romeo	Určuje pojmenovanou skupinu klíčů (*key chain*). Jméno musí souhlasit se jménem nastaveným ve výše uvedené konfiguraci rozhraní.
Router(config-keychain)#key 1	Určuje číslo klíče.
	POZNÁMKA: Rozsah klíčů je od 0 do 2147483647. Identifikační čísla klíčů nemusí být na sebe navazující. V řetězci musí být definován nejméně jeden klíč.
Router(config-keychain-key)#key-string shake-speare	Určuje heslo klíče (*key string*).
	POZNÁMKA: Řetězec klíče (heslo) může obsahovat od 1 do 80 alfanumerických znaků (malá i velká písmena), s výjimkou prvního znaku, který nemůže být číslice.
Router(config-keychain-key)#accept-lifetime start-time {infinite | end-time | duration seconds}	Volitelně (nepovinně) určuje periodu, během které mohou být klíče přijímány.
	POZNÁMKA: Implicitní počátek periody a nejranější akceptovatelný datum je 1.1.1993. Implicitní konec periody je nekonečno.
Router(config-keychain-key)#send-lifetime start-time {infinite | end-time | duration seconds}	Volitelně (nepovinně) určuje periodu, během které mohou být klíče vysílány.
	POZNÁMKA: Implicitní počátek periody a nejranější akceptovatelný datum je 1.1.1993. Implicitní konec periody je nekonečno.

POZNÁMKA: Pro zajištění relevantních údajů pro počátek a konec periody se ujistěte, že má směrovač nastavený správný čas. Doporučovaná praxe je spustit protokol NTP (*Network Time Protocol*) nebo použít jinou metodu pro synchronizaci času, pokud zamýšlíte použít nastavení životnosti klíčů.

Verifikace, ověření funkce EIGRP

Router#show ip eigrp neighbors	Zobrazí tabulku sousedů.

Router#show ip eigrp neighbors detail	Zobrazí tabulku sousedů detailně.
	TIP: Příkaz *show ip eigrp neighbors detail* ověřuje zde je soused nastaven jako hraniční směrovač (*stub router*).
Router#show ip eigrp interfaces	Zobrazí informace pro každé rozhraní.
Router#show ip eigrp interfaces serial 0/0	Zobrazí informace pro konkrétní rozhraní.
Router#show ip eigrp interfaces 100	Zobrazí informace pro rozhraní, na kterém běží proces 100.
Router#show ip eigrp topology	Zobrazí tabulku topologie.
	TIP: Příkaz *show ip eigrp topology* zobrazuje, kde jsou Vaši přípustní následníci (*feasible successors*).
Router#show ip eigrp traffic	Zobrazí počet a typ vyslaných a přijatých paketů.
Router#show ip route eigrp	Zobrazí směrovací tabulku pouze s řádky od EIGRP.

Odstraňování závad EIGRP

Router#debug eigrp fsm	Zobrazí události/akce související s EIGRP metrikou přípustných následníků (*feasible successor metrics (FSM)*).
Router#debug eigrp packet	Zobrazí události/akce související s EIGRP pakety.
Router#debug eigrp neighbor	Zobrazí události/akce související s Vašimi EIGRP sousedy.
Router#debug ip eigrp neighbor	Zobrazí události/akce související s Vašimi EIGRP sousedy (pro protokol IP).
Router#debug ip eigrp notifications	Zobrazí oznámení událostí EIGRP.

Příkazy pro kapitolu 9, EIGRP

Příkaz (Command)	***Popis (Description)***
Router(config)# router eigrp 100	Zapíná EIGRP. 100 je číslo autonomního systému (*autonomous system AS*), které může být

	mezi 1 a 65535. všechny směrovače ve stejném AS musí mít stejné číslo AS.
Router(config-router)# network 192.168.1.32 0.0.0.31	Umožňuje směrování pro podsíť 192.168.1.32/27. (V případě, že jde o podsíť (= není použita implicitní (= třídní) maska pro danou třídu sítě), je nutné v klauzuli **network** použít pseudomasku, zástupnou masku (*wild-card mask*). Pokud by byla uvedena agregovaná třídní adresa, nemusí se žádná pseudomaska používat.).
Router# show ip eigrp neighbors	Zobrazí tabulku sousedů.
Router# show interface serial 0/0/0	Lze ověřit aktuální metriku použitou EIGRP pro rozhraní Serial 0/0/0.
Router(config-if)# bandwidth 128	Mění přenosovou rychlost (*bandwidth*) rozhraní na 128 kb/s.
Router# show ip eigrp topology	Zobrazí tabulku topologie. Tento příkaz Vám ukáže kdo jsou Vaši přípustní následníci (= zástupci) (*feasible successors*), splňují podmínku přípustnosti.
Router# show ip eigrp topology all-links	Zobrazí tabulku topologie včetně cest, které nesplňují podmínku přípustnosti (*feasibility condition*). Zobrazuje všechny možné cesty do cílové sítě.
Router# debug eigrp fsm	Zobrazí události/akce vztahující se k algoritmu DUAL FSM.
Router(config)# ip classless	Umožní beztřídní směrování. (V IOS od Release 11.3 výše je implicitně zapnuto.)
Router(config-router)# no auto-summary	Vypne automatickou sumarizaci sítí na hranicích plné třídy.
Router(config-router)# eigrp log-neighbor-changes	Loguje všechny změny ve vztazích přilehlého sousedství EIGRP (*neighbor adjacency*).
Router(config-if)# ip summary-address eigrp 100 10.10.0.0 255.255.0.0	Umožní manuální sumarizaci na tomto určitém rozhraní pro zadaný adresní prostor 10.10.0.0/16.
Router(config-route)# redistribute static metric	Nastaví EIGRP tak, aby zahrnoval ve svých aktualizacích statické cesty. Je třeba nastavit

	hodnoty EIGRP metrik.

Komplexní praktické laboratorní cvičení – EIGRP

Použijte příklad pro RIPv2.

1. Vypněte RIP (*no router rip*).
2. Směrujte pomocí EIGRP v **autonomním systému** 100. Správně určete pseudomasky pro podsítě. Privátní sítě ve třídě C lze vložit do konfigurace EIGRP bez pseudomasky (= je použita implicitní maska třídy C.)
3. Zakažte propagaci EIGRP **do sítí obsahujících pouze koncová zařízení (zde netranzitních (*stub*) sítí)** (*passive-inteface*).
4. Na hraničním směrovači plné třídy vypněte automatickou sumarizaci.
5. Změňte přenosovou rychlost (*bandwidth*) na jednotlivých rozhraních. Nastavte na obou koncích jednoho média stejnou hodnotu. => Chování se změní. Od původního, kdy se chovalo stejně jako RIP, tzn. Nejlepší je nejkratší cesta (s nejmenším počtem skoků), nyní je délka ovlivněna i přenosovou rychlostí. (Nastavíme na lince mezi R1 a R3 hodnotu 1 000.). Potom do sítě 192.168.2.0 se dostaneme pouze spodní cestou (původně tam byly dvě cesty se stejnou cenou / metrikou).
6. Dále nastavte na směrovači R4 statickou (implicitní) cestu pro síť 10.2.2.0/24 na virtuální rozhraní typu loopback a redistribujte ji na ostatní směrovače. Vypněte automatickou sumarizaci.
7. Zobrazte si na R3:
   - směrovací tabulku (sh ip route),
   - tabulku sousedů (sh ip eigrp neighbors),
   - tabulku topologie (bez a včetně cest, které nesplňují podmínku přípustnosti): sh ip eigrp topology, sh ip topology all-links,
   - ladicí výpis algoritmu DUAL FCM: debug eigrp fsm.

Konfigurace EIGRP na R2:

```
!
router eigrp 100
 passive-interface FastEthernet0/0
 network 172.16.2.0 0.0.0.127
 network 172.16.1.0 0.0.0.255
 network 192.168.2.0
 auto-summary
!
ip classless
!
```

Redistribuce statické cesty na R4:

```
<vynecháno>
interface Loopback0
 ip address 10.1.1.1 255.255.255.0
!
<vynecháno>
!
router eigrp 100
 redistribute static metric 100 10 255 255 1500
 passive-interface FastEthernet0/1
 network 192.168.4.0
 network 192.168.3.0
 no auto-summary
!
ip classless
ip route 10.2.2.0 255.255.255.0 Loopback0
!
<vynecháno>!
```

Směrovač R3:

Směrovací tabulka R3:

```
     10.0.0.0/24 is subnetted, 1 subnets
D EX    10.2.2.0 [170/25605120] via 192.168.3.253, 00:03:32, FastEthernet0/0
     172.16.0.0/16 is variably subnetted, 5 subnets, 3 masks
D       172.16.0.0/16 is a summary, 00:12:04, Null0
D       172.16.1.0/24 [90/33280] via 172.16.2.253, 00:19:46, FastEthernet1/1
D       172.16.2.0/25 [90/30720] via 172.16.2.253, 00:19:46, FastEthernet1/1
C       172.16.2.128/25 is directly connected, FastEthernet1/1
C       172.16.3.0/24 is directly connected, FastEthernet1/0
D    192.168.1.0/24 [90/35840] via 172.16.2.253, 00:12:05, FastEthernet1/1
D    192.168.2.0/24 [90/33280] via 172.16.2.253, 00:19:46, FastEthernet1/1
C    192.168.3.0/24 is directly connected, FastEthernet0/0
D    192.168.4.0/24 [90/30720] via 192.168.3.253, 00:03:30, FastEthernet0/0
D    192.168.5.0/24 [90/284160] via 172.16.2.253, 00:19:46, FastEthernet1/1
```

Tabulka sousedů R3:

```
R3#sh ip eigrp nei
```

```
IP-EIGRP neighbors for process 100
H   Address         Interface      Hold Uptime    SRTT   RTO   Q   Seq
                                   (sec)          (ms)        Cnt  Num
0   172.16.2.253    Fa1/1          11   00:21:49  40     1000  0   74
1   172.16.3.253    Fa1/0          14   00:14:07  40     1000  0   96
2   192.168.3.253   Fa0/0          14   00:05:38  40     1000  0   35
```

Tabulka topologie (bez cest, které nesplňují podmínku přípustnosti) R3:

```
R3#sh ip eigrp topology
IP-EIGRP Topology Table for AS 100

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - Reply status

P 172.16.3.0/24, 1 successors, FD is 2562560
         via Connected, FastEthernet1/0
P 192.168.3.0/24, 1 successors, FD is 28160
         via Connected, FastEthernet0/0
P 172.16.0.0/16, 1 successors, FD is 28160
         via Summary (28160/0), Null0
P 172.16.2.128/25, 1 successors, FD is 28160
         via Connected, FastEthernet1/1
P 172.16.2.0/25, 1 successors, FD is 30720
         via 172.16.2.253 (30720/28160), FastEthernet1/1
         via 172.16.3.253 (2567680/30720), FastEthernet1/0
P 192.168.5.0/24, 1 successors, FD is 284160
         via 172.16.2.253 (284160/281600), FastEthernet1/1
P 192.168.4.0/24, 1 successors, FD is 30720
         via 192.168.3.253 (30720/30720), FastEthernet0/0
P 172.16.1.0/24, 1 successors, FD is 33280
         via 172.16.2.253 (33280/30720), FastEthernet1/1
         via 172.16.3.253 (2565120/28160), FastEthernet1/0
P 192.168.1.0/24, 1 successors, FD is 35840
         via 172.16.2.253 (35840/35840), FastEthernet1/1
         via 172.16.3.253 (2565120/28160), FastEthernet1/0
P 192.168.2.0/24, 1 successors, FD is 33280
         via 172.16.2.253 (33280/30720), FastEthernet1/1
```

```
         via 172.16.3.253 (2567680/30720), FastEthernet1/0
P 10.2.2.0/24, 1 successors, FD is 25605120
         via 192.168.3.253 (25605120/25602560), FastEthernet0/0
R3#
```

Poznámka:

1. **P**- tento směr je v **pasivním stavu** (*passive state*). Když algoritmus DUAL neprovádí svůj výpočet k určení cesty do sítě, směr, cesta je ve stabilním režimu (*stable mode*), který je známý jako pasivní stav (*passive state*). Jestliže DUAL přepočítává nebo hledá novou cestu, směr, cesta je v aktivním stavu (*active state*). Všechna směrovače v topologické tabulce by měly být ve stabilním stavu pro stabilní směrovací doménu. DUAL zobrazí stav A, jestliže je směrovač „*Stuck in Active*“ (= uvázlý, přilepený v aktivním stavu), což je problém pro výuku hledání a odstraňování chyb na úrovni kurzu CCNP.
2. **30720** – inzerovaná vzdálenost záložní cesty (*reported distance of feasible successor*).

Tabulka topologie (včetně cest, které nesplňují podmínku přípustnosti, tzn. všechny cesty) R3:

Kontrolní opakovací otázky a odpovědi (kvíz):

1) Co je účelem EIGRP PDM (= *Protocol Dependent Module,* modul závislý na protokolu)?
   a) PDM poskytuje modulární podporu pro L3 protokoly.
2) Spárujte termíny EIGRP a jejich popisy:
   a) obsahuje směry EIGRP určené pro přeposílání paketů = směrovací tabulka,
   b) primární směr, který má být použit, vybraný algoritmem DUAL = následník (*successor route*)
   c) nejdůležitější datový zdroj EIGRP, obsahuje seznam směrovačů s vytvořeným sousedstvím (*adjacency*) = tabulka sousedů (*neighbor table*)
   d) záložní cesta do cílové sítě = přípustný následník (*feasible successor route*)
   e) obsahuje všechny naučené (zjištěné) směry do všech cílových sítí = topologická tabulka (*topology table*)
3) Který typ paketů EIGRP je použit pro objevování, verifikaci a znovu objevování sousedních směrovačů?
   a) Kontaktní paket hello
4) Jestliže směr EIGRP spadne a v topologické tabulce není pověřený následník (= záložní směr), jakým návěstím (flag) DUAL označí tento směr, který selhal?
   a) Aktivní
5) Které tři tabulky EIGRP spravuje (= udržuje)?
   a) Směrovací

b) topologická

c) sousedů

6) Jaký je účel tabulky sousedů a topologické tabulky u EIGRP?

a) Jsou použity algoritmem DUAL pro vytvoření (naplnění) směrovací tabulky.

7) Co znamená číslo 255/255 ve následujícím výpisu?

```
R1#sh int fa1/0
FastEthernet1/0 is up, line protocol is up (connected)
  Hardware is Lance, address is 0030.a309.4001 (bia 0030.a309.4001)
  Internet address is 172.16.1.253/24
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec, rely 255/255, load 1/255
  Encapsulation ARPA, loopback not set
```

a) Pravděpodobnost, že linka bude dále funkční (= spolehlivost).

8) Spárujte termíny DUAL s jejich popisy:

a) funkční záložní cesta do cíle = přípustný náasledník (*feasible successor*)

b) směr, který je použit pro přeposílání paketů do cíle a zároveň směr s nejmenšími náklady = následník (*successor*)

c) nejnižší vypočtená metrika pro dosažení cílové sítě = přípustná vzdálenost (*feasible distance*)

d) tabulka, která obsahuje následníky i přípustné následníky = topologická tabulka

e) tabulka, která obsahuje pouze následníky = směrovací tabulka

9) Administrátor hledá a odstraňuje závady směrování EIGRP. Který příkaz vypíše všechny možné cesty do cíle?

a) show ip eigrp topology all-link

10) Jaká je inzerovaná (oznamovaná) vzdálenost (*reported distance*) v inzerovaném přípustném následníkovi do sítě 172.16.2.128/25?

```
R1#show ip eigrp topology
IP-EIGRP Topology Table for AS 100

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - Reply status

P 192.168.1.0/24, 1 successors, FD is 28160
         via Connected, FastEthernet0/0
P 172.16.2.128/25, 1 successors, FD is 33280
         via 172.16.1.254 (33280/30720), FastEthernet1/0
         via 172.16.3.254 (4294967295/28160), FastEthernet1/1
```

a) 28160

Kapitola 10 - Směrovací protokoly typu stav linky (Link-State)

V této kapitole se naučíme:

- Popsat základní koncepty a funkce směrovacích protokolů používajících algoritmus stavu linky
- Popsat výhody a požadavky kladené na protokoly typu stav linky

Směrování typu stav linky

Směrovací protokoly typu vektor vzdálenosti (*distance vector*) si můžeme představit jako směrové dopravní značky na silnici (*road signs*), protože směrovače musí rozhodnout o preferovaném směru na základě vzdálenosti neboli metriky do cílových sítí. Právě tak jako cestovatel důvěřuje dopravnímu značení, že ukazuje správnou vzdálenost do dalšího města, směrovače s vektorem vzdálenosti důvěřují, že ostatní směrovače inzerují pravdivou vzdálenost do cílové sítě.

Směrovací protokoly typu stav linky (*link-state*) volí jiný přístup. Směrovací protokoly typu stav linky (*link-state*) jsou, pro představu, spíše jako silniční mapy, protože vytvářejí mapu topologie sítě a každý směrovač tuto mapu používá k určení nejkratší cesty do každé sítě. Tak, jako se vy podíváte do mapy, abyste našli směr do jiného města, směrovače se stavem linky používají topologickou mapu k určení preferované cesty k dosažení dalšího cíle.

Směrovací protokoly typu **stav linky** (*Link-State Routing Protocols*) jsou také známy jako **protokoly typu nejkratší cesta jako první** (*shortest path first protocols, SPF*) a jsou postaveny na **algoritmu SPF Dijkstra**[22].

Pro IP jsou nejznámějšími protokoly stavu linky:

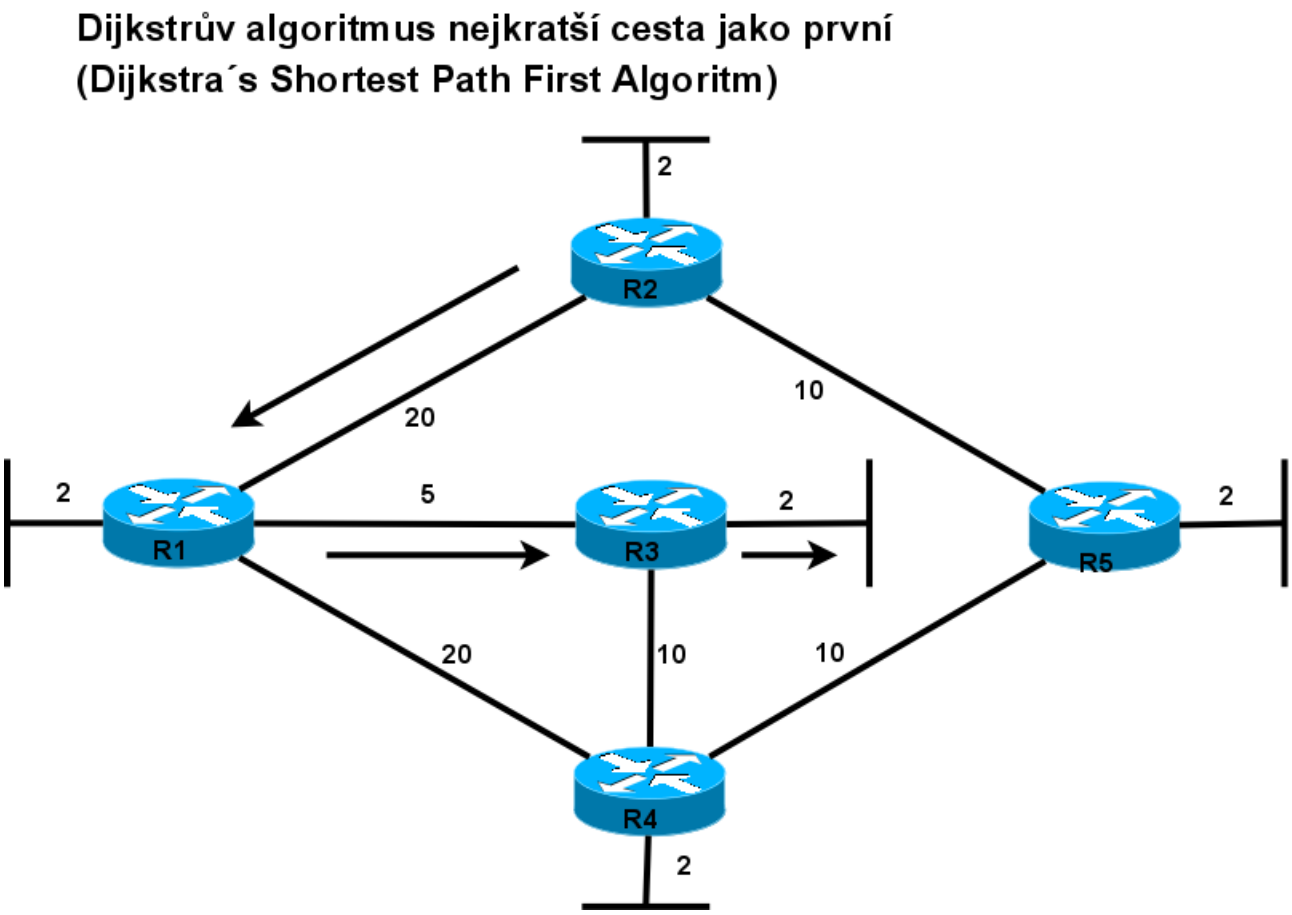
- Open Shortest Path First (OSPF)
- Intermediate System–to–Intermediate System (IS-IS)

Poznámka: OSPF bude diskutován v kapitole 11 a IS-IS v kurzu CCNP. Existují také směrovací protokoly typu stav linky pro sítě nepoužívající protokol IP. Například DNA Phase V od firmy DEC, Netware Link Service Protocol (NLSP) od firmy Novell, ty se nebudou probírat ani v CCNA ani v CCNP.

22 Edsger Wybe Dijkstra (1930 – 2002) byl holandský vědec v oboru počítačů.

Úvod do algoritmu SPF

Algoritmus „nejkratší cesta první“ (*Shortest Path First, SPF*) akumuluje ceny (*costs*) podél každé cesty od zdroje do cíle. Každý směrovač vypočítává algoritmus SPF a určuje metriku = **cenu** (*cost*) ze své vlastní perspektivy (sčítá jednotlivé ceny jednotlivých segmentů sítě (linek) podél každé možné cesty do cíle včetně ceny segmentu cílové sítě (ze směrovače do cílového hostitele) a s výjimkou ceny segmentu zdrojové sítě (od zdrojového hostitele do bránového směrovače zdrojové sítě)). Přestože algoritmus Dijkstra je znám jako algoritmus nejkratší cesta první, je to ve skutečnosti smysl každého směrovacího protokolu.[23]



Postup zpracování algoritmu SPF na směrovači:

1. Každý směrovač se dozví o každé k sobě přímo připojené síti,
2. Každý směrovač je zodpovědný, že řekne „*hello*“ (= pošle kontaktní pakety *hello*) každému sousedovi v přímo připojené síti
   - podobně, jako v EIGRP, se tak vytvoří vztah přilehlosti, sousedství (*adjacency*) (v dané oblasti),

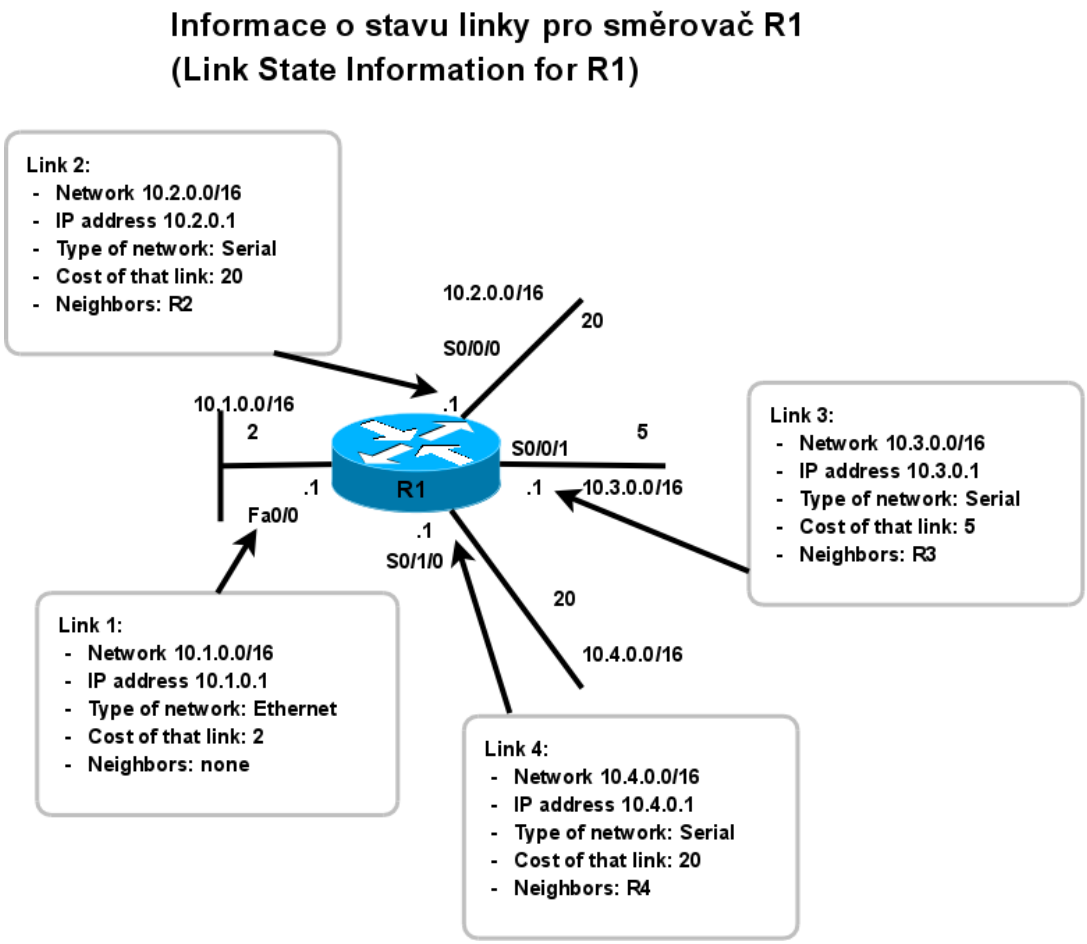
23 Ve směrovací tabulce je vždy pouze „nejlepší“ tj. nejkratší, nejrychlejší cesta (směr) do cíle.

3. Každý směrovač sestavuje **pakety stavu linky** (*Link-State Packet*, *LSP*), které obsahují stavy každé přilehlé (přímo připojené) linky
   - LSP obsahuje:
     - údaje o lince mezi dvěma směrovači: směrovač 1 – směrovač 2, ID souseda, typ linky, adresa sítě, maska, přenosová kapacita, cena,
     - nebo informace o netranzitní síti.
4. Kdykoliv při změně topologie, zapnutí/vypnutí linky, nebo zapnutí směrovače nebo směrovacího protokolu (vytvoření vztahu sousedství), každý směrovač zaplavuje (*flood)* pakety stavu linky (*Link-State Packet*, *LSP*) všechny sousedy v přímo připojených, přilehlých, sítích ve směrovací oblasti, kteří potom ukládají všechny přijaté pakety stavu linky (LSP) do své **databáze stavu linky** (*link-state database*, *LSDB*).
   - Nezapomeňte: LSP není posílán periodicky!
   - Každý směrovač **ve směrovací oblasti** (*area*) bude mít LSP ze všech směrovačů v této oblasti,
5. Jednotlivé směrovače si vytvářejí úplnou a synchronizovanou **mapu topologie sítě** a nezávisle počítají **nejlepší cestu do každé cílové sítě** (s celkovou nejnižší cenou celé trasy).
   - Vytváří si **strom sítě** (*Link State Tree*) – mapu neobsahující smyčky.

Informace o stavu linky

Informace o stavu linek směrovače je známa jako stavy linky (*Link States*). Obsahuje:

- IP adresu sítě a masku podsítě přilehlé sítě,
- IP adresu rozhraní směrovače,
- typ sítě (Ethernet (*broadcast*) nebo sériové dvoubodové připojení (*point-to-point link*)),
- cenu této linky,
- všechny sousedící (přilehlé) směrovače této linky.



Výhody algoritmu Link-State

Následuje několik výhod směrovacích protokolů typu stav linky proti protokolům typu vektor vzdálenosti:

- Každý **směrovač si vytváří vlastní topologickou mapu** neboli **strom SPF** síťové topologie, ze kterého si sám počítá nejkratší cestu.
- Bezprostředním zaplavováním (*flooding*) sousedů pakety LSP se dosáhne **rychlá konvergence.**
- **LSP jsou posílány pouze při změně topologie** a obsahují pouze informace týkající se této změny – automaticky spouštěné aktualizace (*triggered update*).
- **Hierarchický návrh**, při použití více oblastí (*area*).

Systémové požadavky

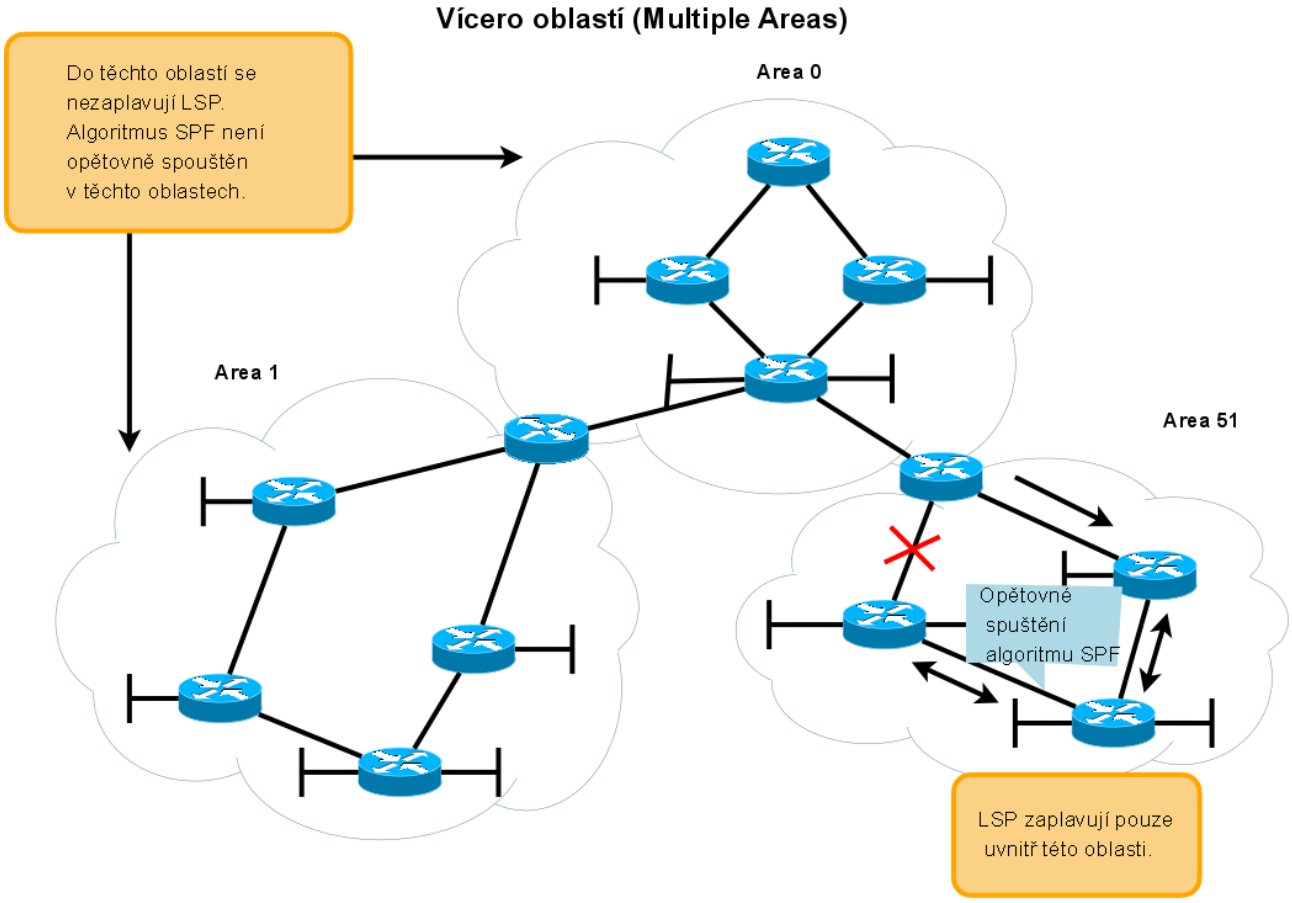
Systémové požadavky na směrovač s protokolem typu stav linky jsou proti protokolům typu vektor vzdálenosti **zvýšené** o:

- Operační paměť pro databázi link-state.

- Procesorový čas pro výpočet algoritmu SPF.
- Přenosová kapacita (šířka pásma) pro záplavy paketů LSP (ta je ale čerpána převážně při startu směrovače, později obvykle nastávají již pouze malé změny topologie).

Vícero oblastí

Aby se zmenšila zátěž procesoru směrovače a požadavky na jeho paměť, je topologie pro směrování typu stav linky rozdělena do malých **oblastí** (*area*). Procesor je nejvíce zatížen při počáteční záplavě (*flood*) paketů stavu linky (*Link State Packet, LSP*), poté už přicházejí pouze změny topologie (tím je pro aktualizace potřeba nižší šířka pásma). Konvergenci sítě urychlují aktualizace spouštěné změnami v síti (*triggered updates*).



Kontrolní opakovací otázky a odpovědi (kvíz):

1) Které tři mechanismy používají směrovací algoritmy typu stav linky (*link state*) k vytvoření a ke správě směrovacích tabulek?
   a) Kontaktní pakety hello,
   b) Oznamovače stavu linky LSA,
   c) Algoritmus SPF.

2) Porovnání vlastností směrovacích algoritmů:
   a) stav linky (*link-state*):
      i. používají algoritmus Dijkstra,
      ii. vytvářejí kompletní topologii na každém směrovači,
      iii. rychlá konvergence (= hlavní výhoda proti vektoru vzdálenosti),
      iv. větší zatížení a požadavky na HW (= hlavní nevýhoda proti vektoru vzdálenosti).
   b) Vektor vzdálenosti (*distance vector*):
      i. používají algoritmus Bellman-Ford,
      ii. závislé na cestách zjištěných od souseda,
      iii. cesty jsou tedy známé „z doslechu“ (*by „rumor“*),
      iv. používají periodicky se opakují aktualizace.
3) Co je obsaženo v LSP posílaných směrovači typu stav linky (*link-state*) na jejich sousedy?
   a) Stav přímo připojených linek
4) Potom, co si dva směrovače OSPF vymění kontaktní pakety *hello* a vytvoří vztah sousedství (*adjacency*), je další krok?
   a) Začnou si navzájem posílat pakety LSP.
5) Jak se směrovač dozví o přímo připojené síti?
   a) Když administrátor přiřadí k rozhraní IP adresu a masku podsítě.

Kapitola 11 - Protokol OSPF

V této kapitole se naučíme:

- Popsat východiska a základní funkce OSPF
- Popsat a použít základní konfigurační příkazy OSPF
- Popsat, vypočítat a modifikovat metriku používanou OSPF
- Popsat proces volby pověřeného směrovače/záložního pověřeného směrovače (*Designated Router / Backup Designated Router - DR/BDR*) v síti s více přístupy (s více branami)
- Využít příkazu „***default-information originate***“ ke konfiguraci a k propagaci implicitní cesty v OSPF

Úvod do OSPF

Open Shortest Path First (OSPF) je **veřejný směrovací protokol typu stav linky**, který byl vyvinut jako náhrada směrovacího protokolu typu vektor vzdálenosti RIP. RIP byl přijatelný v počátcích sítí a Internetu, ale spoléhání se na počet přeskoků jako na jediný způsob určení nejlepší trasy rychle přestalo být ve velkých sítích akceptovatelné. OSPF je **beztřídní směrovací protokol**, který používá pro svoji rozšiřitelnost **koncept oblastí** (*area*). Metrika je definována jako libovolná hodnota nazývaná **cena** (*cost*) podle RFC 2328.

Hlavní výhodou OSPF proti RIP je jeho **rychlá konvergence** a jeho roz**šířitelnost na mnohem větší sítě**. V této závěrečné kapitole tohoto kurzu se naučíte implementaci a konfiguraci OSPF v jedné oblasti. Komplexnější konfigurace jsou v kurzu CCNP.

Historické pozadí

Počáteční vývoj OSPF začala pracovní skupina OSPF při Internet Engineering Task Force (IETF) v roce 1987. V té době byl Internet převážně v akademických a výzkumných sítích financovaných vládou USA.

V roce 1989 byla publikována specifikace OSPFv1 v RFC 1131. OSPFv1 byl experimentální směrovací protokol, který nebyl nikdy nasazen.

V roce 1991 uveden OSPFv2 v RFC 1247 (napsal John Moy). Ve stejné době pracovala ISO na svém vlastním směrovacím protokolu typu stav linky *Intermediate System-to-Intermediate System* (IS-IS). IETF doporučila OSPF jako vnitřní směrovací protokol IGP (Interior Gateway Protocol).

V roce 1998 byla specifikace OSPFv2 aktualizována nyní platnou RFC 2328.

Poznámka: V roce 1999 byl publikován OSPFv3 pro IPv6 v RFC 2740 (napsali John Moy, Rob Coltun a Dennis Ferguson). OSPFv3 je probírána v CCNP.

Linky:

OSPFv2 http://www.ietf.org/rfc/rfc2328.txt

Zjednodušená činnost OSPF

1. Směrovač vysílá přes svá rozhraní kontaktní pakety (*Hello packet*). Pokud se dva navzájem propojené routery pomocí těchto paketů dohodnou na určitých společných parametrech, stávají se sousedy (*neighbors*)
2. Mezi některými ze sousedů se vytvářejí užší vazby sousedství. Tyto směrovače se pak označují jako přilehlé (*adjacent*).
3. Přilehlé směrovače si vzájemně vyměňují aktualizační pakety (*Link-State Update*, *LSU*) obsahující oznamovače LSA (*Link-State Advertisement*). Informace v oznamovačích popisují stav rozhraní směrovače nebo seznam směrovačů připojených k dané síti.
4. Všechny směrovače si ukládají přijaté LSA do své lokální topologické databáze (LSDB) a zároveň je přeposílají na ostatní přilehlé směrovače. Tím se informace postupně záplavově (*flood*) rozšíří mezi všechny směrovače v síti. Výsledkem bude shodná topologická databáze na všech směrovačích.
5. Po naplnění databáze (*Link-State DataBase, LSDB*) každý směrovač samostatně provede výpočet pomocí SPF (Dijkstrova) algoritmu. Jeho výsledkem bude nalezení nejkratší cesty do každé známé sítě v podobě stromu a tím odstranění smyček v topologii sítě.
6. Na základě vypočtených dat ve stromu SPF (*SPF tree*) je možné naplnit směrovací tabulku směrovač nejlepšími cestami do cílových sítí.
7. Pokud dojde ke změně topologie sítě, směrovač na kterém ke změně došlo odešle přilehlým směrovačům informaci v podobě datových položek LSA v LSU paketu. Ty se postupně rozšíří po celé síti a každý směrovač upraví svou topologickou databázi a provede nový výpočet SPF algoritmu.

Zapouzdření zprávy protokolu OSPF

<table>
<tr><td>Záhlaví linkové vrstvy</td><td>Záhlaví paketu IP</td><td>Záhlaví paketu OSPF</td><td>Data specifická dle typu paketu OSPF</td></tr>
<tr><td colspan="4">Rámec linkové vrstvy
Zdrojová MAC adresa = adresa vysílajícího rozhraní
Cílová MAC adresa = Multicast: 01-00-5E-00-00-05 nebo 01-00-5E-00-00-06</td></tr>
<tr><td></td><td colspan="3">Paket IP
Zdrojová IP adresa = adresa vysílajícího rozhraní
Cílová IP adresa = Multicast: 224.0.0.5 nebo 224.0.0.6
Protokol = 89 pro OSPF</td></tr>
<tr><td colspan="2"></td><td colspan="2">Záhlaví paketu OSPF
Kód typu pro Typ paketu OSPF
ID směrovače a ID oblasti</td></tr>
<tr><td colspan="3"></td><td>Typ paketu OSPF
0x01 Hello
0x02 Database Description
0x03 Link State Request
0x04 Link State Update
0x05 Link State Acknowledgment</td></tr>
</table>

Typy paketů OSPF

1. **Hello** – kontaktní pakety *hello* objevují sousedy (*neighbor*) OSPF, vytvářejí a udržují vztah přilehlého sousedství (*adjacency*) s ostatními směrovači OSPF.
2. **DBD - The Database Description** – zkrácený výpis link-state databáze vysílajícího směrovače, určen k ověření a synchronizaci lokální databáze link-state na přijímajícím směrovači.
3. **LSR - Link-State Request** – žádost o další informace pro řádku DBD.
4. **LSU - Link-State Update** – odpověď na LSR, který žádal nové informace. LSU může obsahovat až 11 (7) různých typů oznamovačů *Link-State Advertisements* (LSA) (někdy se jako synonym pro LSA používá termín *Link-State Update* (LSU), ve skutečnosti LSU obsahuje jeden nebo více LSA). Jednotlivé LSA obsahují směrovací informace do cílové sítě.
   Typy LSA:
   4.1. Směrovač – Router,
   4.2. Síť – Network,
   4.3. Agregace – Summary,
   4.4. Agregace – Summary,
   4.5. Externí autonomní systém,
   4.6. Multicast OSPF
   4.7. Definované pro tranzitní oblasti (not-so-stubby areas),
   4.8. Externí atributy pro protokol BGP,
   4.9. nejasný LSA,
   4.10. nejasný LSA,
   4.11. nejasný LSA.
5. **LSAck - Link-State Acknowledgement (LSAck)** – potvrzení přijetí LSU.

Kontaktní pakety Hello

- Objevují sousedy (*neighbor*) OSPF, vytvářejí a udržují vztah přilehlosti, sousedství (*adjacency*) s ostatními směrovači OSPF.
- Inzerují parametry, na kterých se dva směrovače musí shodnout, aby vytvořily vztah sousedství.
- Volí pověřený směrovač (*Designated Router (DR)*) a záložní pověřený směrovač (*Designated Router (BDR)*) v sítích s více přístupy (s více branami) (*multiaccess networks)* jako jsou Ethernet nebo Frame Relay.
- Nejčastěji je zasílán na skupinovou adresu *ALLSPFRouters* 224.0.0.5.

Aby bylo možné vytvořit vztah přilehlého sousedství mezi dvěma směrovači, musí mít rozhraní těchto směrovačů stejné hodnoty pro následující proměnné:

- **Hello interval** – indikuje jak často směrovač vysílá hello pakety (v sekundách)
  - 10 sekund - implicitně v segmentech broadcastových sítí s vícenásobnými přístupy (*broadcast multiaccess (BMA))* (Ethernet) a dvoubodová spojení (point-to-point).
  - 30 sekund v segmentech *non-broadcast multiaccess (NBMA)* sítí (Frame Relay, X.25,

ATM).

- **Dead interval** – perioda v sekundách, po kterou bude směrovač čekat na příjem hello paketu, než označí sousedství za „mrtvé“ a zruší ho. Jestliže vyprší dead interval před tím, než směrovač přijme hello paket, OSPF smaže souseda ze své link-state databáze LSDB. Směrovač zaplaví (*floods*) informacemi, že sousedství je vypnuté, všechna rozhraní, na kterých je spuštěný OSPF. Je obvykle nastaven na čtyřnásobek intervalu hello.
  - 40 sekund - segmenty multiaccess a point-to-point,
  - 120 sekund – sítě NBMA.
- **Network type** – typ sítě:
  - Broadcast sítě - ty sítě, které jsou schopny vzájemně propojit více než dva počítače a navíc zajišťují, že jeden vyslaný paket mohou přijmout současně všechny počítače. Typickými představiteli broadcast sítí jsou sítě typu Ethernet nebo FDDI.
  - Point to point sítě (dvoubodové spoje) - sítě spojující pouze dva směrovače. Jejich typickým příkladem jsou sériové linky. Na těchto sítích se nevolí DR/BDR a směrovače na point to point sítích se vždy stávají přilehlými. Pro komunikaci mezi nimi se používá pouze multicast adresa 224.0.0.5.
  - NBMA sítě - *Non Broadcast Multi Access*. Síť tohoto typu může propojit více než dva směrovače, není však schopna posílat broadcasty. Není tedy možné vyslat paket, který by byl přijat všemi směrovači současně. Jako příklad NBMA sítě můžeme uvést síť Frame Relay, ATM nebo X.25. Na NBMA síti se volí DR a BDR a veškerá komunikace probíhá pomocí unicastů.

Volba DR a BDR

Aby zmenšil objem provozu OSPF **v sítích s více přístupy**, s více branami (*multiaccess network*), OSPF volí pověřený směrovač (*Designated Router (DR)*) a záložní pověřený směrovač (*Backup Designated Router (BDR)*). DR (směrovač s nejvyšší prioritou) je zodpovědný za aktualizace všech ostatních směrovačů OSPF (nazývaných DROther), když nastane změna topologie v síti s více přístupy (*multiaccess network*). BDR monitoruje DB a převezme funkci DR, pokud aktuální DR selže.

Jak je volen DR a BDR?

1. DR: směrovač s nejvyšší prioritou OSPF rozhraní
2. BDR: směrovač s druhou nejvyšší prioritou OSPF rozhraní
3. Jestliže jsou OSPF priority shodné, rozetne nerozhodný výsledek nejvyšší ID směrovače.

Konfigurace priority rozhraní:

Router(config-if)#ip ospf priority 255

(tímto se nastaví rozhraní nejvyšší možná priorita => bude zvoleno DR)

Zobrazení aktuální priority daného rozhraní a ID routeru:

Router# show ip ospf interface jméno_rozhraní

Implicitní priorita pro rozhraní směrovače je jednička (1). **Pokud mají všechny směrovače nastavenou implicitní prioritu rozhraní, bude jako DR zvolen směrovač s nejvyšším identifikáto-**

rem směrovače (Router ID, RID).

Jednotlivé DROther (= jiné směrovače než DR nebo BDR (*DR other*)) budou formovat sousedství typu **FULL** pouze s DR a BDR, ale budou stále formovat přilehlé sousedství s jakýmkoliv jiným směrovačem DROther, který je připojený v síti. To znamená, že všechny směrovače DROther v síti s více přístupy (*multiaccess*) stále přijímají kontaktní pakety hello ze všech ostatních směrovačů DROther. Tímto způsobem jsou si vědomy všech směrovačů v síti. Když dva směrovač typu DROther zformují přilehlé sousedství, je stav sousedství zobrazen jako typ **2WAY**. Další stavy sousedství jsou diskutovány v kurzu CCNP.

Algoritmus OSPF

Každý OSPF směrovač spravuje svoji databázi stavů linek (*link-state database*), která obsahuje jednotlivé LSA přijaté ze všech ostatních směrovačů. Jakmile směrovač přijal všechny oznamovače v aktualizaci a sestavil svoji lokální databázi, OSPF použije Dijsktrův algoritmus SPF k vytvoření stromu SPF (*SPF tree*). SPF strom je potom použit k naplnění směrovací tabulky nejlepšími směry do každé sítě.

Autentizace

OSPF pakety jsou šifrované a autentizované.

Je dobrou praxí autentizovat přenášené směrovací informace. RIPv2, EIGRP, OSPF, IS-IS a BGP mohou všechny být nakonfigurované, aby šifrovaly a autentizovaly jejich směrovací informace (aktualizace, nikoliv směrovací tabulky). Tato praxe zajišťuje, že směrovače akceptují pouze ty směrovací informace z druhých směrovačů, které byly nastavené se stejným heslem nebo autentizační informací.

Poznámka: Autentizace nešifruje směrovací tabulku.

Identifikátor směrovače

Identifikátor směrovače (Router ID) je unikátní identifikace směrovače v OSPF doméně. Router ID je jednoduše IP adresa. Směrovače Cisco odvozují hodnotu Router ID na základě tří kritérií a následující nadřazenosti:

1. směrovač použije IP adresu nastavenou příkazem „**router-id**“,
2. jestliže není nastaven příkaz „**router-id**“, směrovač si zvolí **nejvyšší adresu** ze **všech** svých rozhraní typu **loopback**,
3. jestliže nejsou nastavená žádná rozhraní typu loopback, směrovač si vybere **nejvyšší aktivní adresu** ze **všech svých fyzických rozhraní**.

Verifikace identifikátoru směrovače (*Router ID*): **show ip protocols**. Pokud některá verze IOS nevrací Router ID v tomto příkazu, použijte: **show ip ospf** nebo **show ip ospf interface**.

Rozhraní typu loopback:

```
Router(config)#interface loopback number
Router(config-if)#ip address ip-address subnet-mask
```

Nastavení příkazem router-id:

```
Router(config)#router ospf process-id
Router(config-router)#router-id ip-address
```

Při dodatečných změnách v příkazu *network* nebo *router-id* je vhodné restartovat směrovač (Router#reload) nebo vymazat proces OSPF:

```
Router#clear ip ospf process
```

Duplikace identifikátorů směrovače:

IOS duplicitu detekuje a oznámí:

```
%OSPF-4-DUP_RTRID1: Detected router with duplicate router ID
```

Ověření funkčnosti OSPF

Ověření vztahu přilehlosti

`show ip ospf neighbor`

Výstupy příkazu show ip ospf neighbor:

- **Neighbor ID** – identifikátor sousedícího směrovače.
- **Pri** - OSPF priorita rozhraní.
- **State** – stav rozhraní. Stav FULL znamená, že směrovač a jeho soused mají identické databáze LSDB. Stavy (*state*) jsou podrobněji diskutovány v kurzu CCNP.
- **Dead Time** – zbývající čas, který bude směrovač čekat na přijmutí kontaktního paketu hello od souseda před tím, než prohlásí sousedství za zrušené (mrtvé). Tato hodnota je resetována, když rozhraní přijme kontaktní *hello* paket.
- **Address** - IP adresa sousedova rozhraní, kterému je tento směrovač přímo připojen.
- **Interface** – rozhraní, na kterém tento směrovač zformoval sousedství / přilehlost se sousedem.

Poznámka: na multiaccess sítích[24] - sítích s vícenásobným přístupem, jako je Ethernet, mohou mít dva přilehlé směrovače zobrazen jejich stav jako 2WAY. Viz Volba DR a BDR.

```
R3#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
10.1.1.1          0   FULL/  -        00:00:36    192.168.10.5    Serial0/1/0
10.2.2.2          0   FULL/  -        00:00:36    192.168.10.9    Serial0/1/1
R3#
```

Dva směrovače nemusejí mít vytvořený vztah přilehlosti, sousedství, jestliže:

- Vzájemně nesouhlasí (*do not match*) masky podsítě, to má za příčinu, že směrovače jsou v různých sítích,

24 Sítě s více branami.

- Vzájemně nesouhlasí OSPF Hello nebo Dead intervaly,
- Vzájemně nesouhlasí OSPF Network Type.
- Chybějící nebo nesprávný OSPF příkaz **network** (například různá oblast (*area*)).

Nastavení rozhraní, časovačů, typ sítě, cenu linky a vznik sousedství v příslušném směru na konkrétním rozhraní ověříte pomocí *show ip ospf interface*:

```
R3#show ip ospf interface serial 0/1/1
Serial0/1/1 is up, line protocol is up
  Internet address is 192.168.10.10/30, Area 0
  Process ID 1, Router ID 10.3.3.3, Network Type POINT-TO-POINT, Cost: 64
  Transmit Delay is 1 sec, State POINT-TO-POINT, Priority 0
  No designated router on this network
  No backup designated router on this network
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:00
  Index 3/3, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1 , Adjacent neighbor count is 1
    Adjacent with neighbor 10.2.2.2
  Suppress hello for 0 neighbor(s)
R3#
```

V případě, že není sousedství vytvořeno je možno nesoulad nastavených hodnot časovačů zjistit též z výpisu ladicího příkazu: *debug ip ospf events*

```
R3#
00:04:06: OSPF: Rcv hello from 10.2.2.2 area 0 from Serial0/1/1 192.168.10.9
00:04:06: OSPF: Mismatched hello parameters from 192.168.10.9
00:04:06: OSPF: Dead R 40 C 40 Hello R 10 C 50 Mask R 255.255.255.252 C
255.255.255.252
R3#
```

Standardní ověřovací příkazy pro OSPF

- **show ip protocols**

- **show ip ospf**
- **show ip ospf interface ...**
- **show ip route**
- **debug ip ospf events**

Administrativní vzdálenost

Viz tabulka *Administrativní vzdálenosti pro jednotlivé směrovací protokoly* v kapitole 3.

Pro OSPF je implicitní administrativní vzdálenost (*distance*, AD) = 110.

Zjistíte ji příkazem *show ip protocols*

```
R3#sh ip protocols


Routing Protocol is "ospf 1"
<vynecháno>
  Distance: (default is 110)


R3#
```

Metrika OSPF

Metrika OSPF se nazývá cena (*cost*). Citát z RFC 2328:

„Cena je přiřazena k odchozí straně každého rozhraní směrovače. Tato cena je nastavitelná systémovým administrátorem. Čím nižší cena, tím více žádoucí je toto rozhraní pro přeposlání datového provozu.“

Pamatujte, že RFC 2328 nespecifikuje, jaké hodnoty by měly být použity k určení ceny. My se budeme zabývat dvěma metodami nastavení cen na směrovačích Cisco.

Cost = 10^8/bandwidth (v bps, b/s)

Příklad:

Jaká je OSPF cena linky FastEthernet? 10^8/100 000 000 b/s = 1
Jaká je OSPF cena linky T1 ? 10^8/1 544 000 b/s = 64.7, která je systémem IOS zaokrouhlená na cenu 64.
Jaká je OSPF cena vytáčené linky 56K (*dial up*)? 10^8/56000 = 1785.71, která je systémem IOS zaokrouhlená na cenu 1785.

1) Konfigurace metriky pomocí šířky pásma (rychlosti) (*bandwidth*):

```
R2(config)# interface serial0/0/0
R2(config-if)# bandwidth 64
```

2) Konfigurace metriky přímo nastavením ceny (*cost*):

```
R3(config)# interface serial0/0/0
R3(config-if)# ip ospf cost 390
```

OSPF 2 (aktualizace 1998) RFC: http://www.ietf.org/rfc/rfc2328.txt

=> U OSPF (a stejně tak i u EIGRP) stav směrování závisí i na postupně provedených změnách nastavení (algoritmus je konečný automat (FSM)). Někdy je tedy nutné, po změnách konfigurace, vymazat tabulky ukládající průběžné stavy. Tzn. resetovat procesy příslušného směrovacího protokolu (Router# clear ip ospf process) nebo restartovat směrovače.

Příkazy pro kapitolu 11, OSPF

Konfigurace OSPF: Mandatorní (povinné) příkazy

Router(config)#router ospf 123	Nastartuje proces OSPF 123. Identifikátor (ID) procesu je jakékoliv kladné celé číslo mezi 1 a 65 535. Identifikátor procesu se nevztahuje k oblasti OSPF (*OSPF area*). Identifikátor procesu nesouvisí s oblastí OSPF. Identifikátor procesu pouze odlišuje jeden proces od jiného na jednom zařízení.
Router(config-router)#network 172.16.10.0 0.0.0.255 area 0	OSPF inzeruje rozhraní nikoliv sítě. Používá pseudomasku (*wildcard mask*) k určení, která rozhraní inzerovat. Tato příkazová řádka říká: „Všechna rozhraní s adresou 172.16.10.x mají být vložena do OSPF oblasti 0“.
	POZNÁMKA: Číslo identifikátoru procesu na jednom směrovači nemusí souhlasit s číslem identifikátoru na jakémkoliv jiném směrovači. Na rozdíl od EIGRP, rovnost tohoto čísla na všech směrovačích nezajistí, že se vytvoří sousedství.
Router(config-router)#log-adjacency-changes detail	Nastaví směrovač tak, aby posílal systémové logovací zprávy (*syslog message*), když nastane změna stavu mezi OSPF sousedy.
	TIP: Ačkoliv je příkaz *log-adjacency- changes* implicitně zapnutý, je bez použití klíčového slova *detail* oznamována (report) pouze událost

	zapnuto/vypnuto.

Použití pseudomasky v oblastech OSPF

Router(config-router)#network 172.16.10.1 0.0.0.0 area 0	Tuto řádku čtěte jako: „Každé rozhraní s přesnou adresou 172.16.10.1 má být dáno do oblasti 0.“
Router(config-router)#network 172.16.10.0 0.0.255.255 area 0	Tuto řádku čtěte jako: „Každé rozhraní s přesnou adresou 172.16.X.X má být dáno do oblasti 0.“
Router(config-router)#network 0.0.0.0 255.255.255.255 area 0	Tuto řádku čtěte jako: „Každé rozhraní s jakoukoliv adresou má být dáno do oblasti 0.“

Konfigurace OSPF: Nepovinné (volitelné) příkazy

Virtuální rozhraní zpětná smyčka (Loopback)

Router(config)#interface loopback 0	Vytvoří virtuální rozhraní pojmenované loopback 0 a potom směrovač přepne do konfiguračního režimu rozhraní.
Router(config-if)#ip address 192.168.100.1 255.255.255.255	Přiřadí k rozhraní IP adresu. (Všimněte si zadané masky. Kolik je v této síti adres?[25])
	POZNÁMKA: rozhraní typu zpětná smyčka má vždy stav „*up and up*“ = rozhraní administrativně zapnuté a běžící protokol linkové vrstvy. Nevypne se bez toho aniž by se ručně zadal příkaz *shutdown*. To je výborné pro použití rozhraní *loopback* jako identifikátorů směrovačů v OSPF (OSPF router ID).

Router ID

Router(config)#router ospf 1	Spustí proces číslo 1 v OSPF.
Router(config-router)#router-id 10.1.1.1	Nastaví identifikátor směrovače (Router ID) na 10.1.1.1. Jestliže je tento příkaz použit na OSPF proces, který je již aktivní (má sousedy), je nový identifikátor směrovače použit až po příštím znovuzavedení systému (*reload*) nebo po manuálním restartu procesu OSPF.

25 Je tam právě jenom jedna adresa a to adresa sítě.

Router(config-router)#no router-id 10.1.1.1	Odstraní z konfigurace statický identifikátor směrovače. Jestliže je tento příkaz použit na OSPF proces, který je již aktivní (má sousedy), je staré chování ID směrovače použito až při příštím znovuzavedení systému (*reload*) nebo při manuálním restartu procesu OSPF.

Volby pověřeného a záložního pověřeného směrovače (DR/BDR)

Router(config)#interface serial 0/0	Změní režim směrovače na režim konfigurace rozhraní.
Router(config-if)#ip ospf priority 50	Změní OSPF prioritu rozhraní na 50.
	POZNÁMKA: Přiřazená priorita může být mezi 0 a 255. Priorita 0 činí tento směrovač nezpůsobilý stát se pověřeným směrovačem (*designated router (DR)*) nebo záložním pověřeným směrovačem (*backup designated router (BDR)*). Nejvyšší priorita vyhrává volbu. Priorita 255 zaručuje nerozhodný výsledek volby. Jestliže mají všechny směrovače stejnou prioritu, bez ohledu na číslo priority, je výsledek volby nerozhodný. Nerozhodnost je prolomena nejvyšším ID směrovače.

Modifikace ceny metriky (cost)

Router(config)#interface serial 0/0	Změní režim směrovače na režim konfigurace rozhraní.
Router(config-if)#bandwidth 128	Pokud změníte *bandwidth*, OSPF přepočte cenu (*cost*) linky.
Nebo	
Router(config-if)#ip ospf cost 1564	Změní cenu na hodnotu 1564.
	POZNÁMKA: Cena linky je určena vydělením referenční šířky pásma (*reference bandwidth*) šířkou pásma tohoto rozhraní. Šířka pásma rozhraní je číslo mezi 1 a 10 000 000. Měrná jednotka (*unit of measurement*) je kilobit (kb). Cena je číslo mezi 1 a 65 535. Cena nemá

	měrnou jednotku - je to jenom číslo.

Autentizace: jednoduchá

Router(config)#router ospf 1	Spustí OSPF proces 1.
Router(config-router)#area 0 authentication	Umožní jednoduchou autentizaci, heslo bude posíláno v čistém textu.
Router(config-router)#exit	Návrat do globálního konfiguračního režimu.
Router(config)#interface fastethernet 0/0	Přesun do konfiguračního režimu rozhraní.
Router(config-if)#ip ospf authentication-key fred	Nastaví klíč (*key*) tj. heslo (*password*) na hodnotu *fred*.
	POZNÁMKA: Heslo může být libovolný řetězec znaků vložených z klávesnice do délky 8 bajtů. Aby byly schopny si vyměňovat OSPF informace, musí mít všechny sousedící směrovače ve stejné síti stejné heslo.

Autentizace: použití šifrování MD5

Router(config)#router ospf 1	Spustí OSPF proces 1.
Router(config-router)#area 0 authentication message-digest	Umožní autentizaci, kdy heslo bude zašifrováno MD5.
Router(config-router)#exit	Návrat do globálního konfiguračního režimu.
Router(config)#interface fastethernet 0/0	Přesun do konfiguračního režimu rozhraní.
Router(config-if)#ip ospf message- digest-key 1 md5 fred	1 je identifikátor klíče (*key-id*). Tato hodnota musí být stejná jako na sousedícím směrovači. Md5 indikuje použití hašovacího algoritmu MD5. *fred* je klíč (heslo) a musí být stejné jako na sousedícím směrovači.
	POZNÁMKA: Jestliže není použit příkaz *service password- encryption*, když je implementována MD5 OSPF autentizace, je tajné heslo MD5 uloženo v konfiguraci v NVRAM jako čistý text (*plain text*).

Časovače

Router(config-if)#ip ospf hello-interval timer 20	Změní časovač *Hello Interval* na 20 sekund.
Router(config-if)#ip ospf dead-interval 80	Změní časovač *Dead Interval* na 80 sekund.
	POZNÁMKA: Časovače Hello a Dead Interval musí být na směrovačích stejné, aby se mohly stát sousedy.

Propagace implicitní cesty

Router(config)#ip route 0.0.0.0 0.0.0.0 s0/0	Vytvoří implicitní cestu.
Router(config)#router ospf 1	Spustí OSPF proces 1.
Router(config-router)#default-information originate	Nastaví, aby byla implicitní cesta propagována na všechny směrovače OSPF.
Router(config-router)#default-information originate always	Volba *always* (= vždy) propaguje implicitní „čtyr-nulovou“ cestu i když na tomto směrovači žádná není nastavená.
	POZNÁMKA: Příkazy *default-information originate* nebo *default-information originate always* jsou konfigurovány obvykle pouze na „vstupním“ nebo „bránovém“ směrovači, tzn. na směrovači, který propojuje Vaši síť s vnějším světem - *Autonomous System Boundary Router (ASBR)*.

Ověření konfigurace OSPF

Router#show ip protocol	Zobrazí parametry všech směrovacích protokolů běžících na tomto směrovači.
Router#show ip route	Zobrazí kompletní směrovací tabulku IP.
Router#show ip ospf	Zobrazí základní informace o směrovacích procesech OSPF.
Router#show ip ospf interface	Zobrazí informace o OSPF, vztahující se ke všem rozhraním.
Router#show ip ospf interface fastethernet 0/0	Zobrazí informace o OSPF, vztahující se k rozhraní fastethernet 0/0.
Router#show ip ospf border-routers	Zobrazí informace o hraničních a okrajových

	směrovačích.
Router#show ip ospf neighbor	Vypíše všechny OSPF sousedy a jejich stavy.
Router#show ip ospf neighbor detail	Zobrazí detailní výpis sousedů.
Router#show ip ospf database	Zobrazí obsah OSPF databáze.
Router#show ip ospf database nssa-external	Zobrazí stavy externích linek do oblastí *Not-So-Stubby Area (NSSA)*[26].

Odstraňování závad OSPF

Router#clear ip route *	Vymaže obsah směrovací tabulky a vynutí si její znovu naplnění.
Router#clear ip route a.b.c.d	Smaže cestu do konkrétní sítě a.b.c.d
Router#clear ip opsf counters	Vynuluje čítače OSPF.
Router#clear ip ospf process	Vynuluje celý proces OSPF, vynutí si znovuvytvoření sousedství, databáze a směrovací tabulky.
Router#debug ip ospf events	Zobrazí všechny události OSPF.
Router#debug ip ospf adjacency	Zobrazí jednotlivé stavy OSPF a volby DR/BDR mezi směrovači ve vztahu sousedství.
Router#debug ip ospf packets	Zobrazí pakety OSPF.

Přehled základních příkazů pro OSPF

Příkaz (Command)	***Popis (Description)***
Router(config)# router ospf 123	Zapne OSPF s číslem procesu (*process number*) 123. Identifikátor (ID) procesu je číslo s jakoukoliv hodnotou mezi 1 a 65 535. Číslo procesu není stejné jako oblast OSPF (OSPF area).
Router(config-router)# network 172.16.10.0 0.0.0.255 area 0	OSPF inzeruje (*advertise*) rozhraní, nikoli sítě. Používá pseudomasku (*wildcard mask)* k určení, která rozhraní se mají inzerovat. Zobrazený příkaz je třeba číst takto: kterékoliv rozhraní s adresou 172.16.10.x má být vloženo do oblasti

26 Netranzitní sítě, do kterých se propagují směrovací informace.

	0. (*OSPF area 0*)
Router(config-if)# ip ospf priority 50	Mění prioritu (*priority*) OSPF rozhraní na 50.
Router(config-if)# bandwidth 128	Mění šířku pásma, přenosovou kapacitu (*bandwidth*) rozhraní na 128 kbps.
Router(config-if)# ip ospf cost 1564	Mění cenu (*cost*) na hodnotu 1564.
Router(config-if)# ip ospf hello-interval 20	Mění časovač intervalu rozesílání paketů Hello (*Hello interval timer*) na 20 sekund.
Router(config-if)# ip ospf dead-interval 80	Mění nastavení, jak dlouho se bude čekat na paket Hello před prohlášením, že linka je shozena (*Dead interval timer*), na 80 sekund.
Router(config)# ip route 0.0.0.0 0.0.0.0 s0/0/0	Vytváří statickou implicitní cestu směřující ven z rozhraní Serial 0/0/0. Tato cesta bude mít administrativní vzdálenost 0.
Router(config-router)# default-information originate	Nastaví, že je implicitní cesta propagována na všechny směrovače OSPF.
Router# show ip protocols	Zobrazí parametry pro všechny, na směrovači běžící, směrovací protokoly.
Router# show ip route	Zobrazí kompletní směrovací tabulku.
Router# show ip ospf	Zobrazí základní informace pro všechny procesy OSPF běžící na směrovači.
Router# show ip ospf interface	Zobrazí informace o OSPF jak jsou vztažené ke všem rozhraním.
Router# show ip ospf neighbor	Zobrazí všechny OSPF sousedy a jejich stavy.
Router# show ip ospf neighbor detail	Zobrazí detailní výpis sousedů.

Cvičení

Základní konfigurace OSPF podle Lab 11.6.1 scénář A, scénář B.

Kontrolní opakovací otázky a odpovědi (kvíz):

1) Tři pravdivá tvrzení ohledně směrovacích protokolů typu stav linky:

a) jsou všeobecně známé jako protokoly SPF (*Shortest Path First*, nejkratší cesta jako první),
b) udržují komplexní databázi síťové topologie,
c) jsou založené na algoritmu Dijkstra.

2) Spárování termínů a jejich popisů:
   a) kontaktní pakety hello = vytvářejí a udržují vztah sousedství směrovačů (*adjacency*),
   b) výměna LSA = je spuštěna, když nastane změna topologie,
   c) stav linky = popis rozhraní a jeho vztahu s jiným směrovačem,
   d) algoritmus SPF = vypočítává nejlepší (nejkratší) cestu do cílové sítě.
3) Z jakých dvou důvodů by měl administrátor při konfiguraci OSPF používat rozhraní typu zpětná smyčka (*loopback*)?
   a) Zpětné smyčky jsou logická virtuální rozhraní, která nelze vypnout (shodit, *do not go down*)
   b) Adresa zpětné smyčky bude použitá jako ID směrovače a „přebije“ IP adresu lokálního rozhraní na směrovači.
4) U kterých dvou typů sítí nemůže být zvolen pověřený směrovač OSPF?
   a) Point-to-point,
   b) point-to-multipoint.
5) Administrátor vložil příkaz „router ospf 100“, jaký je význam čísla 100?
   a) číslo (identifikátor) procesu OSPF (process ID).
6) Smysl příkazu „bandwidth 56“ vloženého na sériovém rozhraní směrovače OSPF?
   a) Změní hodnotu ceny (*cost*) linky.
7) Který faktor bere v úvahu Cisco implementace OSPF při výpočtu ceny linky?
   a) Bandwidth.
8) Propagace implictní cesty v OSPF:
   a) default-information originate
9) Pokud mají participující směrovače stejnou prioritu, co bude vzato v potaz při volbě DR/BDR v OSPF?
   a) Router ID.
10) Podtrhněte vadný typ paketu pro OSPF: hello, LRU, LSR, LSAck, DBD.

Přílohy

Opakování - příklady na adresaci IPv4

IPv4 má 32 bitovou adresu. Toto binární číslo se zapisuje v dekadickém tečkovém (= kanonickém) zápise po celých oktetech (bajtech) oddělených tečkou.
Například 10101100.00010000.00000001.00000010 zapíšeme jako 172.16.1.2.
Uvědomte si, že tento kanonický zápis je jakoby v číselné soustavě o základu 256.
(Platí například: 0.0.2.0 – 0.0.0.1 = 0.0.1.255).

Celkový počet všech možných IP adres je 2^{32} = 4 294 967 296. Maximální rozsah IP adres (všechny možné adresy v IPv4) je 0.0.0.0 až 255.255.255.255. Jiný způsob zápisu tohoto **adresního bloku** je 0.0.0.0/0. To znamená **adresa sítě** a **/prefix (délka prefixu)**. Prefix (lomítkový tvar) je jiný způsob zápisu masky podsítě. (Takovéto názvosloví je používané v novém adresním schématu VLSM, ve starším adresním schématu CIDR se naopak prefixem myslí síťová část adresy a o masce v lomítkovém tvaru se mluví jako o délce prefixu.) Prefix vyjadřuje kolik je v masce zleva binárně jedniček. (Například: / 19 = 255.255.224.0.). IP adresa se skládá ze dvou částí: **síťové části** a **hostitelské části**. Počet bitů v síťové části adresy je daný počtem jedničkových bitů zleva v masce sítě (podsítě).
Masku může zapsat také v kanonickém tvaru. Například /19 = 255.255.224.0.

Krajní **meze adresního bloku** se nazývají:

- **adresa sítě** (*network address*) – v hostitelské části adresy jsou binárně samé nuly (např. 172.16.16.16/28)
- **adresa všeměrového vysílání** (*broadcast address*) – v hostitelské části adresy jsou binárně samé jedničky (např. 172.16.16.31/28)

Tyto dvě adresy nelze použít pro adresaci fyzického zařízení (portu).

Počet adres v jedné síti je určen počtem bitů v **hostitelské části** adresy = $2^{(32\text{-prefix})}$
Počet všech možných sítí se stejnou maskou je určen počtem bitů v **síťové části** adresy = $2^{(\text{prefix})}$
Počet stejně velkých podsítí k jedné výchozí síti je určen **počtem vypůjčených bitů** = $2^{(\text{počet vypůjčených bitů})}$
Počet vypůjčených (*borrowed*) bitů (rozdíl mezi počtem bitů v nové masce podsítě a počtem bitů v masce výchozí sítě) = prefix nové podsítě – prefix výchozí sítě. Ve vypůjčených bitech je přímo obsaženo pořadové číslo podsítě vzhledem k výchozí síti.

Jaké procento obsadí všechny adresy ve třídě A, B, C, D a E vzhledem ke všem možným adresám v protokolu IPv4?

Třída	***Počet všech adres ve všech sítích jedné třídy = počet všech TŘÍDNÍCH SÍTÍ (v jedné konkrétní třídě) * ADRES v jedné nich***	***% vzhledem ke všem možným IPv4 adresám***
Celý IPv4	2^{32}	100,00%
A	$2^{7}*2^{24}=2^{31}$ (1. bit v prvním bitu je fixně 0, zbývá 7 bitů v síťové části IP adresy a je 24 bitů v hostitelské části)	50,00%
B	$2^{14}*2^{16}=2^{30}$	25,00%
C	$2^{21}*2^{8}=2^{29}$	12,50%
D	$2^{4}*2^{24}=2^{28}$	6,25%
E	$2^{4}*2^{24}=2^{28}$	6,25%

Z toho vidíme, že třídní adresa neekonomicky plýtvá s dostupnými adresami. Tomu se zabraňuje:

1. Zavedením privátních adres v privátních sítích. (Na hraničním směrovači mezi neveřejnou a veřejnou sítí potom musí být NAT).
2. Beztřídní adresací – tvorbou menších adresních bloků – podsítí (*subnets, subnetting*).

Pro zadanou adresu hostitele (10.65.10.10) a různé masky (/9, /10, /11, /12, /13, /14, 15) vypočtěte adresu sítě, ve které příslušná adresa leží, a adresu všesměrového vysílání téže sítě:
Vypočteme pomocí binárního tvaru IP adresy. Maska leží vždy ve druhém bajtu zleva a ten v tomto případě je $(65)_{10} = (0100\ 0001)_2$
10.65.10.10/9: 10.0.0.0 – 10.127.255.255
10.65.10.10/10: 10.64.0.0 – 10.127.255.255
10.65.10.10/11: 10.64.0.0 – 10.95.255.255
10.65.10.10/12: 10.64.0.0 – 10.79.255.255
10.65.10.10/13: 10.64.0.0 – 10.71.255.255
10.65.10.10/14: 10.64.0.0 – 10.67.255.255
10.65.10.10/15: 10.64.0.0 – 10.65.255.255

Máte IP adresu hostitelského počítače a prefix (masku v lomítkovém (*slash*) tvaru) 172.16.61.210/20. Určete:

- masku v kanonickém tvaru: 255.255.240.0
- velikost bloku adres v kanonickém tvaru: počet bitů v hostitelské části adresy je 32 – 20 = 12 bitů => $2^{12} = 2^{4}*2^{8} = 0.0.\mathbf{16}.0$
- adresa sítě (odmaskováním): 172.16.61.210 AND 255.255.240.0 = 172.16.**48**.0/20
- adresa všesměrového vysílání je adresa následující sítě zmenšená o 0.0.0.1: 172.16.48.0 + 0.0.16.0 – 0.0.0.1 = 172.16.63.255/20
- Jiný způsob výpočtu adresy sítě (pomocí velikosti bloku): hranice masky leží ve 3. bajtu a adresa sítě musí být celočíselný násobek velikosti bloku. Nejbližší nižší násobek 16 k 61 je **48** a tedy 172.16.48.0/20.

Máte zadánu adresu sítě 10.60.0.0 a adresu všesměrového vysílání 10.63.255.255 jedné sítě (jednoho adresního bloku). Určete masku a velikost adresního bloku.

- Rozdíl krajních adres sítě 10.63.255.255 – 10.60.0.0 = 0.3.255.255. To je číslo, které má binárně samé jednotky v hostitelské části a nazývá se pseudomaska (zástupná maska). Maska

je tedy dvojkový doplněk této inverzní masky (pseudomasky). 255.255.255.255 – 0.3.255.255 = 255.252.0.0 => /14. Velikost bloku je rovna pseudomasce plus 0.0.0.1. 0.3.255.255 + 0.0.0.1 = 0.4.0.0. (Jinak řečeno hostitelská část má 32-14=18 bitů a velikost bloku 2^{18} = 0.4.0.0.)

Máte zadánu síť 172.16.48.0/20. Kolikátá podsíť to je při zadané adresa a masce vzhledem k plné třídě.

- První bajt je 172, jde tedy o třídu B s implicitní maskou /16. K implicitní masce máme vypůjčeny 4 bity (17. až 20.).
- Třetí bajt je binárně 00110000. Ve vypůjčené části je binárně 0011 což jsou dekadicky 3. Jde tedy o třetí podsíť.
- Jiný postup řešení: velikost bloku při masce /20 je 2^4*2^8 = 0.0.16.0 a 172.16.48.0 – 172.16.0.0 = 0.0.48.0 = 3*(0.0.16.0). Jde tedy o třetí podsíť.

Máte zadánu síť 172.16.48.0/20. Rozdělte ji alespoň na 3 nové stejně velké podsítě. Určete druhou podsíť z nich. (Číslovat začínáme vždy od 0.) Vejde se do ní 1000 klientů?

- Nová maska: Máme vytvořit alespoň 3 nové sítě. Počet vytvořených podsítí je vždy mocninou základu 2. Takže, nejbližší vyšší mocnina dvou ke třem jsou 4 = 2^2 => k původní masce si musíme vypůjčit 2 bity a nová maska podsítě je tedy /20+2=/22
- Velikost bloku: 32-22=10 bitů může adresovat 2^{10} = 2^2*2^8 = 0.0.4.0 (vejde se do ní 1022 klientů + dvě rezervované adresy).
  Adresy sítí a všesměrového vysílání u čtyř vytvořených podsítí budou:

Č.	Rozsah adres	Binární tvar: síťová část (vyp.bity) + **hostitelská část**
0.	172.16.48.0 – 172.16.51.255	10101100.00010000.001100**00.00000000** 10101100.00010000.001100**11.11111111**
1	172.16.52.0 – 172.16.55.255	10101100.00010000.001101**00.00000000** 10101100.00010000.001101**11.11111111**
2	172.16.56.0 – 172.16.59.255	10101100.00010000.001110**00.00000000** 10101100.00010000.001110**11.11111111**
3	172.16.60.0 – 172.16.63.255	10101100.00010000.001111**00.00000000** 10101100.00010000.001111**11.11111111**

- Druhá podsíť je 172.16.48.0 + 2*(0.0.4.0) = 172.16.56.0. Adresy klientů této sítě leží v rozsahu: 172.16.56.1 až 172.16.59.254.
- Ve vypůjčených dvou bitech (21. a 22. bit zleva) je hodnota $(10)_2$ = $(2)_{10}$, což je přímo pořadové číslo vytvořené podsítě. (Obsah 3. bajtu zleva je $(56)_{10}$ = $(0011\ 1000)_2$.)

Máte zadánu adresu 172.16.25.100/21 v kolikáté podsíti vzhledem k výchozí třídní síti tato adresa leží? (Předpokládáme adresní schéma CIDR – je podsíťována třídní adresa a všechny podsítě jsou stejné.)

- výchozí třídní síť je ve třídě B a má implicitní masku /16,
- obsah 3. bajtu je $(25)_{10}$ = $(0001\ 1001)_2$, ve vypůjčených bitech je $(00011)_2$ = $(3)_{10}$, tedy leží ve třetí podsíti,
- 172.16.25.100 – 172.16.0.0 = 0.0.25.100, velikost bloku je 0.0.8.0, adresa leží ve 3. podsíti

172.16.24.0/21 – 172.16.31.255/21.

Máte zadánu výchozí síť 172.16.16.0/20 (jde už o podsíť plné třídy). Začínající administrátor učeň ji podsíťoval (na stejně velké podsítě) maskou /25. Na portu směrovače zkouší nastavit adresu 172.16.31.127/25. V kolikáté podsíti tato adresa leží? A proč se mu nedaří nastavit tuto adresu?

- V binárním tvaru zobrazíme v posledních 2B vypůjčené bity: 0001 1111. 0111 1111. $(11110)_2=(30)_{10}$ Jde tedy o 30. podsíť.
- Velikost bloku je pro prefix /25 rovna 0.0.0.128. Adresa sítě je 172.16.31.127 AND 255.255.255.128 = 172.16.31.0. Následující síť je 172.16.31.0+0.0.0.128 = 172.16.31.128. Z toho vidíme, že adresa 172.16.31.127 je všesměrová adresa.

Máte zadány dvě adresy 172.16.2.3/22 a 172.16.3.2/24. Můžete je použít na síťových rozhraních jednoho směrovače, to znamená nepřekrývají se sítě, ve kterých ty dvě adresy leží?

- Rozsahy adres sítí, ve kterých zadané adresy leží, jsou 172.16.0.0-172.16.3.255/22 a 172.16.3.0-172.16.3.255/24, sítě se překrývají a nelze je proto použít na jednom směrovači najednou.

Máte zadány dvě adresy, které chce vložit jako adresy síťových rozhraní na jednom směrovači: 172.16.1.100/23 a 172.16.3.10/22. Překrývají se sítě, ve kterých leží?
172.16.0.0 – 172.16.1.255 velikost bloku je 0.0.2.0
172.16.0.0 – 172.16.3.255 velikost bloku je 0.0.4.0
Sítě se překrývají a nelze je proto použít na jednom směrovači. Tuto chybu odhalí operační systém směrovače. Ale POZOR adresy nelze použít ani v jedné skupině sítí (směrovací doméně). Tato chyba je ale zákeřnější, protože ji na rozdíl od předchozí operační systém směrovače neodhalí a zapojení „z neznámých příčin“ nefunguje.

Je adresa 172.32.1.1 neveřejná (privátní)?

- Není. Nejbližší adresní blok neveřejných adres je 172.16.0.0-172.31.255.255 to jest 172.16.0.0/12. Zadaná adresa leží mimo tento rozsah. Všimněte si, že rozsah privátních adres je tvořen 16 třídními bloky ve třídě B. Představuje tedy jednu nadsíť (*supernet*) pro 16 třídních bloků 172.16.0.0/16 až 172.31.0.0/16.

Vztah délky prefixu a velikosti bloku

Prefix	Maska (binárně)	Maska	Inverzní maska (binárně)	Inverzní maska	Počet bitů hostitele	Velikost bloku	Velikost bloku (binárně)	Velikost bloku (kanonicky)
/8	11111111000000000000000000000000	255.0.0.0	00000000111111111111111111111111	0.255.255.255	24	16777216	00000001000000000000000000000000	1.0.0.0
/9	11111111100000000000000000000000	255.128.0.0	00000000011111111111111111111111	0.127.255.255	23	8388608	00000000100000000000000000000000	0.128.0.0
/10	11111111110000000000000000000000	255.192.0.0	00000000001111111111111111111111	0.63.255.255	22	4194304	00000000010000000000000000000000	0.64.0.0
/11	11111111111000000000000000000000	255.224.0.0	00000000000111111111111111111111	0.31.255.255	21	2097152	00000000001000000000000000000000	0.32.0.0
/12	11111111111100000000000000000000	255.240.0.0	00000000000011111111111111111111	0.15.255.255	20	1048576	00000000000100000000000000000000	0.16.0.0
/13	11111111111110000000000000000000	255.248.0.0	00000000000001111111111111111111	0.7.255.255	19	524288	00000000000010000000000000000000	0.8.0.0
/14	11111111111111000000000000000000	255.252.0.0	00000000000000111111111111111111	0.3.255.255	18	262144	00000000000001000000000000000000	0.4.0.0
/15	11111111111111100000000000000000	255.254.0.0	00000000000000011111111111111111	0.1.255.255	17	131072	00000000000000100000000000000000	0.2.0.0
/16	11111111111111110000000000000000	255.255.0.0	00000000000000001111111111111111	0.0.255.255	16	65536	00000000000000010000000000000000	0.1.0.0
/17	11111111111111111000000000000000	255.255.128.0	00000000000000000111111111111111	0.0.127.255	15	32768	00000000000000001000000000000000	0.0.128.0
/18	11111111111111111100000000000000	255.255.192.0	00000000000000000011111111111111	0.0.63.255	14	16384	00000000000000000100000000000000	0.0.64.0
/19	11111111111111111110000000000000	255.255.224.0	00000000000000000001111111111111	0.0.31.255	13	8192	00000000000000000010000000000000	0.0.32.0
/20	11111111111111111111000000000000	255.255.240.0	00000000000000000000111111111111	0.0.15.255	12	4096	00000000000000000001000000000000	0.0.16.0
/21	11111111111111111111100000000000	255.255.248.0	00000000000000000000011111111111	0.0.7.255	11	2048	00000000000000000000100000000000	0.0.8.0
/22	11111111111111111111110000000000	255.255.252.0	00000000000000000000001111111111	0.0.3.255	10	1024	00000000000000000000010000000000	0.0.4.0
/23	11111111111111111111111000000000	255.255.254.0	00000000000000000000000111111111	0.0.1.255	9	512	00000000000000000000001000000000	0.0.2.0
/24	11111111111111111111111100000000	255.255.255.0	00000000000000000000000011111111	0.0.0.255	8	256	00000000000000000000000100000000	0.0.1.0
/25	11111111111111111111111110000000	255.255.255.128	00000000000000000000000001111111	0.0.0.127	7	128	00000000000000000000000010000000	0.0.0.128
/26	11111111111111111111111111000000	255.255.255.192	00000000000000000000000000111111	0.0.0.63	6	64	00000000000000000000000001000000	0.0.0.64
/27	11111111111111111111111111100000	255.255.255.224	00000000000000000000000000011111	0.0.0.31	5	32	00000000000000000000000000100000	0.0.0.32
/28	11111111111111111111111111110000	255.255.255.240	00000000000000000000000000001111	0.0.0.15	4	16	00000000000000000000000000010000	0.0.0.16
/29	11111111111111111111111111111000	255.255.255.248	00000000000000000000000000000111	0.0.0.7	3	8	00000000000000000000000000001000	0.0.0.8
/30	11111111111111111111111111111100	255.255.255.252	00000000000000000000000000000011	0.0.0.3	2	4	00000000000000000000000000000100	0.0.0.4
/31	11111111111111111111111111111110	255.255.255.254	00000000000000000000000000000001	0.0.0.1	1	2	00000000000000000000000000000010	0.0.0.2

Ověřte si pochopení látky:

1. Které z následujících jsou adresy sítě? (Vyberte dvě.)
   a) 64.104.3.7/28
   b) 192.168.12.64/26
   c) 192.135.12.191/26
   d) 198.18.12.16/28
   e) 209.165.200.254/27
   f) 220.12.12.33/27
2. Administrátor sítě vytváří síť pro malou firmu, která má 22 hostitelských počítačů. ISP přiřadil pouze jednu IP adresu směrovatelnou do Internetu. Který adresní blok může administrátor použít pro adresaci této sítě?
   a) 10.11.12.16/28
   b) 172.31.255.128/27
   c) 192.168.1.0/28
   d) 209.165.202.128/27
3. Která maska podsítě může být použita na hostitelské počítači v síti 128.107.176.0/22?
   a) 255.0.0.0
   b) 255.248.0.0
   c) 255.255.252.0
   d) 255.255.255.0
   e) 255.255.255.252
4. Pro vytvoření dvoubodového WAN spojení (*point-to-point*) vám by vám přidělen adresní blok 10.255.255.224/28. Kolik takových sítí WAN může být v tomto bloku adres?
   a) 1
   b) 4
   c) 7
   d) 14
5. Co definuje jednu logickou IP síť?
6. Pojmenujte a popište účel tří typů adres IPv4:
7. Administrátor sítě potřebuje vytvořit novou síť, která má 14 počítačů a dvě síťová rozhraní na směrovači. Která maska podsítě poskytne odpovídající počet adres s minimálním plýtváním adresami.
   a) 255.255.255.128

b) 255.255.255.192

c) 255.255.255.224

d) 255.255.255.240

e) 255.255.255.248

f) 255.255.255.252

8. Co rozlišuje každý ze tří typů adres IPv4?
9. Napište seznam tří forem komunikace IPv4.
10. Napište důvod proč jsou definovány specifické rozsahy IPv4 adres pro veřejné a pro privátní použití.
11. Hostitelský počítač z jižní pobočky firmy nemůže přistupovat k serveru s adresou 192.168.254.222/24. Během prozkoumávání hostitelského počítače jste zjistili, že má IPv4 adresu 169.254.11.15/16. Co je očividný problém?

    a) hostitelský počítač používá adresu lokální linky (local-link)

    b) server používá vadnou masku podsítě

    c) hostitelský počítač má přiřazenou adresu všesměrového vysílání

    d) server si myslí, že hostitelský počítač je v jedné logické síti s tímto serverem.

12. Vypište tři důvody pro plánování a dokumentaci adres v síti.
13. Uveďte příklady zařízení, kde by měl administrátor přiřazovat IPv4 adresy staticky.
14. Co je primární motivací pro vývoj a zavádění protokolu IPv6.
15. Jaký je účel masky podsítě v adresaci IPv4?
16. Vypište faktory, které by se měli vzít v úvahu při plánování adresního schéma IPv4.
17. Které jsou tři testy pomocí služebního programu (utility) ping pro ověření funkčnosti hostitelského počítače v síti?
18. Které jsou to rezervované a speciální IPv4 adresy a jak se používají?
19. Proč je protokol ICMPv4 důležitý ve vztahu k činnosti IPv4? Jaké jsou typy zpráv ICMP?

Zabezpečení sítě pomocí přístupových seznamů IP

(Pro ty, kdo chtějí vědět víc. Zde uvádím pouze základní nastavení a podrobněji bude probráno ve čtvrtém semestru CCNA Exploration.)

V této kapitole se naučíme:

- Základní informace o přístupových seznamech *(Access Control List, ACL)*
- Čísla přístupových seznamů ACL (*Access Control List, ACL*)
- Použití zástupných (pseudo)masek
- Klíčová slova pro ACL
- Vytvoření standardního přístupového seznamu
- Aplikace standardního přístupového seznamu na rozhraní
- Ověření funkčnosti ACL
- Odstranění ACL
- Vytvoření rozšířeného ACL
- Aplikace rozšířeného ACL na rozhraní
- Klíčové slovo „**established**“ (nepovinné)
- Vytvoření pojmenovaného ACL
- Pořadová čísla řádků v pojmenovaném ACL
- Odstranění konkrétních řádků z pojmenovaného ACL
- Tipy pro číslování řádků
- Komentáře k řádkům ACL
- Omezení přístupu k virtuálnímu terminálu

Základní informace o přístupových seznamech

Přístupové seznamy (*Access Control List, ACL*) se používají k zabezpečení sítí a řízení provozu do a ze sítě. Přístupové seznamy (ACL) filtrují provoz na základě pravidel, které můžete nastavit v příkazech (jednotlivých řádcích) svého ACL. Tato pravidla určují, zda pakety jsou povoleny nebo zakázány, jaké služby mají možnost používat, a kdo s kým může komunikovat. Příkladem toho je například, zda má hostitel povolen přístup k Internetu nebo má přístup k určitému serveru v síti.

Přístup ke službám je filtrován na základě čísel portů. Porty 0 až 1023 se nazývají dobře známé porty. Patří mezi ně běžné služby, jako Telnet s portem 23 a HTTP, který používá port 80. Firmy vyvíjející SW mohou požádat organizaci IANA o přidělení čísla portu k identifikaci konkrétní aplikace v rozmezí čísla portu 1024 až 49151. Například: Shockwave používá číslo portu 1626. Porty 49 152 až 65 535 jsou přiřazována dynamicky koncovým zařízením a jsou dočasné, tj. trvají pouze po dobu trvání spojení.

Když je nakonfigurován, změní ACL router na firewall a testuje veškerý provoz proti každé řádce seznamu před tím, než mohou být předány do jejich místa určení. Tento proces řídí síťový provoz a pomáhá chránit vaši síť, ale rozhodně přidává latenci. Pakety jsou kontrolovány proti řádkům - příkazům ACL v pořadí, ve kterém jsou nakonfigurovány, od shora dolů, jednotlivý příkaz (řádek) najednou. Při prvním výskytu shody se zadanou podmínkou, podle toho, zda je provoz povolen (*permit*) či zakázán (*deny*), je příslušná akce provedena. Jestliže je každý příkaz akce povolení (permit), je na konci seznamu příkazů implicitní „zákaz všeho“ ("deny any"), který ale není zobrazován a není ho ani třeba nakonfigurovat. Jakýkoliv paket, který neodpovídá žádnému příkazu s povolením provozu, je potom automaticky odmítnut. Proto, pokud jsou všechny příkazy akce odmítnutí

(deny), musí být vložen jako poslední příkaz „povolení všeho“ ("permit any"), jinak je veškerý provoz zakázán! To je velmi častý omyl, který dělají správci sítě - nováčci.

Standardní ACL jsou jednoduché příkazy, které provoz povolují nebo zakazují na základě zdrojové IP adresy. Měly by být nastaveny na routeru tak blízko k cíli, jak jen to je možné.

Rozšířenými ACL lze filtrovat provoz pomocí více proměnných, jako jsou protokol, zdrojová a cílová IP adresa a číslo portu, na základě čeho je příslušná služba nebo aplikace filtrována. Protože tyto rozšířené ACL jsou přesné, jsou nakonfigurovány na routeru co nejblíže ke zdroji který je filtrován. Toto zabraňuje odepření provozu z důvodu spotřeby přenosové kapacity.

Standardní a rozšířené seznamy ACL mohou být nakonfigurovány buď jako **pojmenované** nebo jako **číslované**. ACL obecně mají dáno číslo identifikující jejich typ - 1 až 99 pro standardní IP a 100 až 199 pro rozšířené IP ACL. Pojmenované ACL nemají žádná omezení, ale co je důležitější, mohou být snadno změněny bez nutnosti začínat konfiguraci znovu od samého začátku. Když chcete přidat příkaz do prostředka takovéhoto seznamu, lze použít Pořadová čísla řádek, aniž by bylo třeba začít celou konfiguraci znovu od začátku. Jak již bylo uvedeno, pakety jsou vyhodnocovány proti řádkům přístupovému seznamu v pořadí, ve kterém byly řádky vytvořeny. To znamená, že pokud uděláte chybu a dáte jako první příkaz, který by měl být jako poslední, nelze ho jenom jednoduše odstranit, ale musíte začít konfigurovat od začátku. To je důvod, proč je doporučeno napsat si svůj ACL v textovém editoru a nechat ho někým zkontrolovat ještě před tím, než ho vložíte do vaší konfigurace. Používáte-li pojmenovaného ACL, nejste omezeni počtem příkazů, které můžete vytvářet, a také vám to umožní vyladit konfigurace ACL bez nutnosti celého jeho odstranění a začínání znovu.

Po vytvoření přístupového seznamu, který slouží svému účelu, je dalším a posledním krokem jeho aplikace na rozhraní. Pro to, aby ACL pracoval, musíte ho aplikovat na rozhraní a to buď v příchozím (in) nebo odchozím (out) směru. Bez toho je ACL k ničemu a je to totéž jako nemít vůbec žádné zabezpečení.

Pro jeden směrovaný protokol lze mít aplikován **jeden ACL na jednom rozhraní a jednom směru**.

Čísla ACL

1–99 nebo 1300–1999	Standardní IP (Standard IP)
100–199 nebo 2000–2699	Rozšířený IP (Extended IP)
600–699	AppleTalk
800–899	IPX
900–999	Rozšířený IPX (Extended IPX)
1000–1099	IPX Service Advertising Protocol

Zástupné masky

Zástupná maska, pseudomaska (*wildcard mask*) určuje, které části IP adresy se při rozhodování musí shodovat, aby se na ně aplikovalo pravidlo **permit (povolit)** nebo **deny (zakázat)** v jednom příkazu (jedné řádce) přístupového seznamu (ACL):

- **0 (nula)** v zástupné masce znamená, že odpovídající bit v adrese je kontrolován a že se musí přesně shodovat.
- **1 (jednička)** v zástupné masce znamená, že odpovídající bit v adrese je ignorován a může být 1 nebo 0.

Příklad 1: 172.16.0.0 0.0.255.255

```
172.16.0.0  = 10101100.00010000.00000000.00000000
0.0.255.255 = 00000000.00000000.11111111.11111111
              -----------------------------------
výsledek =    10101100.00010000.xxxxxxxx.xxxxxxxx
172.16.x.x (Cokoliv mezi 172.16.0.0 a 172.16.255.255
bude odpovídat uvedenému příkazu v příkladu.)
```

TIP: Oktet složený ze samých nul v masce znamená, že se musí oktet v adrese přesně shodovat. Oktet složený ze samých jedniček v masce znamená, že příslušný oktet v adrese může být celý ignorován.

Příklad 2: 172.16.8.0 0.0.7.255

```
172.168.8.0 = 10101100.00010000.00001000.00000000
0.0.0.7.255 = 00000000.00000000.00000111.11111111
              -----------------------------------
výsledek =      10101100.00010000.00001xxx.xxxxxxxx
00001xxx = 00001000 až 00001111 = 8-15
xxxxxxxx = 00000000 až 11111111 = 0-255
```

Cokoliv mezi 172.16.8.0 až 172.16.15.255 bude vyhovovat uvedenému příkazu.

Klíčová slova pro ACL

any	Používá se místo výrazu 0.0.0.0 255.255.255.255, při porovnání vyhovuje jakékoliv IP adrese
host	Používá se v zástupné masce místo výrazu 0.0.0.0, při porovnání tedy vyhoví pouze jedna konkrétní adresa.

Vytvoření standardního ACL

Router(config)#access-list 10 permit 172.16.0.0 0.0.255.255	Čteme jako: „Všechny pakety se zdrojovou IP adresou 172.16.x.x budou mít povolen další průchod sítí.“
access-list	Příkaz pro vytvoření ACL
10	Libovolné číslo mezi 1 až 99, nebo 1300 až 1999, definuje příslušný ACL jako standardní IP ACL
permit	Pakety které vyhovují tomuto příkazu mají povoleno pokračovat v průchodu.
172.16.0.0	Zdrojová IP adresa, která bude porovnávána
0.0.255.255	Zástupná maska

Router(config)#access-list 10 deny host 172.17.0.1	Čteme jako: „Všechny pakety se zdrojovou IP adresou 172.17.0.1 budou vyřazeny a zahozeny.“
access-list	Příkaz pro vytvoření ACL
10	Libovolné číslo mezi 1 až 99, nebo 1300 až 1999, definuje příslušný ACL jako standardní IP ACL
deny	Pakety které vyhovují tomuto příkazu budou budou vyřazeny a zahozeny.
Host	Klíčové slovo
172.17.0.1	Adresa konkrétního hostitele

Router(config)#access-list 10 permit any	Čteme jako: „Všechny pakety s jakoukoliv zdrojovou IP adresou budou mít povolen další průchod sítí.“
access-list	Příkaz pro vytvoření ACL
10	Libovolné číslo mezi 1 až 99, nebo 1300 až 1999, definuje příslušný ACL jako standardní IP ACL
permit	Pakety které vyhovují tomuto příkazu mají povoleno pokračovat v průchodu.
Any	Klíčové slovo, které znamená jakákoliv IP adresa

- **TIP:** Na konci každého ACL je pevně zakódován implicitní příkaz **deny**. Nevidíte ho sice, ale přikazuje „zakaž vše co dosud nebylo povoleno“. Je to vždy poslední řádka každého ACL. Pokud tomuto implicitnímu zákazu chcete zabránit, vložte na poslední řádku standardního ACL příkaz **permit any** nebo v případě rozšířeného ACL vložte příkaz **permit ip any any**.

Aplikace standardního ACL na rozhraní

Router(config)#interface fastethernet 0/0	Přejde do konfiguračního režimu rozhraní
Router(config-if)#ip access-group 10 in	Vezme všechny řádky ACL, které jsou definovány jako části skupiny 10 a aplikuje na na příchozí směr. Pakety přicházející na směrovač na rozhraní fastethernet 0/0 budou zkontrolovány.

- **TIP:** ACL mohou být aplikovány buď na příchozí směr (klíčové slovo **in**) nebo na odchozí směr (klíčové slovo **out**).
- **TIP:** Pro jeden protokol (IP), na jednom rozhraní a na jeden směr (dovnitř/ven) je možné aplikovat pouze jeden ACL.
- **TIP:** Standardní ACL aplikujte co nejblíže je to možné k cílové síti nebo zařízení.

Kontrola ACL

Router#show ip interface	Zobrazí všechny ACL aplikované na zadané rozhraní
Router#show access-lists	Zobrazí obsah všech ACL na směrovači
Router#show access-list *access-list-number*	Zobrazí obsah ACL se zadaným číslem
Router#show access-list *name*	Zobrazí obsah ACL se zadaným jménem
Router#show run	Zobrazí všechny ACL a jejich přiřazení k rozhraním

Odstranění ACL

Router(config)#no access-list 10	Odstraní všechny ACL s číslem 10

Vytvoření rozšířeného ACL

Router(config)#access-list 110 permit tcp 172.16.0.0 0.0.0.255 192.168.100.0 0.0.0.255 eq 80	Čteme jako: „HTTP pakety se zdrojovou IP adresou 172.16.0.x budou mít povolen další průchod do cílové adresy 192.168.100.x.”
access-list	Příkaz pro vytvoření ACL
110	Libovolné číslo mezi 100 až 199, nebo 2000 až 2699, definuje příslušný ACL jako rozšířený IP ACL.
permit	Pakety které vyhovují tomuto příkazu mají povoleno pokračovat v průchodu.
tcp	Protokol musí být TCP.
172.16.0.0	Zdrojová IP adresa, která bude porovnávána.
0.0.0.255	Zástupná maska pro zdrojovou IP adresu.
192.168.100.0	Cílová IP adresa, která bude porovnávána.
0.0.0.255	Zástupná maska pro cílovou IP adresu.
eq	Operátor, který znamená „rovná se“
80	Port 80, indikující provoz HTTP.

Router(config)#access-list 110 deny tcp any 192.168.100.7 0.0.0.0 eq 23	Čteme jako: „Telnet pakety s jakoukoliv zdrojovou IP adresou budou vyřazeny, jestliže jsou adresovány do konkrétního hostitele 192.168.100.7.”
access-list	Příkaz pro vytvoření ACL
110	Libovolné číslo mezi 100 až 199, nebo 2000 až 2699, definuje příslušný ACL jako rozšířený IP

	ACL.
deny	Pakety které vyhovují tomuto příkazu budou vyřazeny a zahozeny.
tcp	Protokol musí být TCP.
any	Jakákoliv zdrojová IP adresa.
192.168.100.7	Cílová IP adresa, která bude porovnávána.
0.0.0.0	Zástupná maska; adresa musí přesně souhlasit.
eq	Operátor znamenající „rovná se“.
23	Port 23, indikující provoz Telnet.

Aplikace rozšířeného ACL na rozhraní

Router(config)#interface fastethernet 0/0 Router(config-if)#ip access-group 110 out	Přepne do konfiguračního režimu rozhraní a vezme všechny řádky ACL, které jsou definovány jako část skupiny 110 a aplikuje je na odchozí směr. Pakety opouštějící rozhraní fastethernet 0/0 budou zkontrolovány.

- **TIP:** ACL mohou být aplikovány buď na příchozí směr (klíčové slovo **in**) nebo na odchozí směr (klíčové slovo **out**).
- **TIP:** Pro jeden protokol (IP), na jednom rozhraní a na jeden směr (dovnitř/ven) je možné aplikovat pouze jeden ACL.
- **TIP:** Rozšířený ACL aplikujte co nejblíže je to možné ke zdrojové síti nebo zařízení.

Klíčové slovo „established“ (nepovinné)

Router(config)#access-list 110 permit tcp 172.16.0.0 0.0.0.255 192.168.100.0 0.0.0.255 eq 80 established	Indikuje již navázané (*established*) spojení.

- **POZNÁMKA:** Spárování (splnění podmínky) nyní nastane pouze pokud má datagram TCP nastavený bit ACK nebo RST.
- **TIP:** Klíčové slovo „**established**“ bude funkční (a má smysl) pouze pro protokol TCP a nikoliv pro UDP (který je nespojovaný).
- **TIP:** Uvažte následující situaci: chcete zabránit hackerům zneužít port 80 pro vniknutí do vaší sítě. Protože neprovozujete žádný Web server zablokovat příchozí provoz na portu 80 ovšem s výjimkou kdy vnitřní uživatelé potřebují přístup na Web. Při jejich požadavku na Web je nutné povolit návratový provoz na port 80. Řešením je použití příkazu **established**. ACL nyní povolí vstup odpovědi do vaší sítě, protože bude mít nastavený ACK bit jako výsledek prvotního požadavku zevnitř vaší sítě. Požadavky zvenčí budou blokovány, protože ACK bit nebude nastaven, ale odpovědím bude povolen průchod.

Vytvoření pojmenovaného (named) ACL

Router(config)#ip access-list extended serveraccess	Vytvoří pojmenovaný rozšířený ACL s názvem *serveraccess* a přejde do konfiguračního režimu pojmenovaného ACL.
Router(config-ext-nacl)#permit tcp any host 131.108.101.99 eq smtp	Povolí průchod paketů poštovních paketů SMTP z libovolného zdroje do hostitele 131.108.101.99.
Router(config-ext-nacl)#permit udp any host 131.108.101.99 eq domain	Povolí průchod paketů DNS z libovolného zdroje do hostitele 131.108.101.99.
Router(config-ext-nacl)#deny ip any any log	Zamítne všechny ostatní pakety jdoucí kamkoliv. Jestliže bude paket zamítnut, bude zaprotokolován (log) pro pozdější prohlídku.
Router(config-ext-nacl)#exit	Návrat do globálního konfiguračního režimu.
Router(config)#interface fastethernet 0/0 Router(config-if)#ip access-group serveraccess out	Přejde do konfiguračního režimu rozhraní a aplikuje tento ACL na rozhraní fastethernet 0/0 v odchozím směru.

Použití pořadového čísla řádky v pojmenovaném ACL

Router(config)#ip access-list extended serveraccess2	Vytvoří pojmenovaný rozšířený ACL s názvem *serveraccess2.*
Router(config-ext-nacl)#10 permit tcp any host 131.108.101.99 eq smtp	Pro tuto řádku použije pořadové číslo 10.
Router(config-ext-nacl)#20 permit udp any host 131.108.101.99 eq domain	Řádek s pořadovým číslem 20 se zařadí za řádku 10.
Router(config-ext-nacl)#30 deny ip any any log	Řádek s pořadovým číslem 30 se zařadí za řádku 20.
Router(config-ext-nacl)#exit	Návrat do globálního konfiguračního režimu.
Router(config)#interface fastethernet 0/0	Přejde do konfiguračního režimu rozhraní.
Router(config-if)#ip access-group serveraccess2 out	Aplikuje ACL na odchozí směr tohoto rozhraní.
Router(config-if)#exit	Návrat do globálního konfiguračního režimu.
Router(config)#ip access-list extended serveraccess2	Přejde do konfiguračního režimu pojmenovaného ACL se jménem serveraccess2.
Router(config-ext-nacl)#25 permit tcp any host 131.108.101.99 eq ftp	Pořadové číslo 25 zařadí tuto řádku mezi řádky 20 a 30.
Router(config-ext-nacl)#exit	Návrat do globálního konfiguračního režimu.

- **TIP:** Pořadová čísla se používají, aby umožnila snazší editaci přístupového seznamu. V předchozím příkladu byla na řádcích ACL použita pořadová čísla 10, 20 a 30. Pokud byste

chtěli přidat další řádku, přidala by se za poslední řádku číslo 30. Pokud byste chtěli jít více nahoru, museli byste smazat celý ACL a potom znovu použít čísla řádek ve správném pořadí. Nyní můžete vložit novou řádku s pořadovým číslem přímo na správné místo.

- **POZNÁMKA:** Argument *sequence-number* byl přidán v Cisco IOS Release 12.2(14)S. Byl plně integrován do Cisco IOS Release 12.2(15)T.

Odstranění řádky v pojmenovaném ACL s použitím čísla řádky

Router(config)#ip access-list extended serveraccess2	Přejde do režimu konfigurace pojmenovaného ACL s názvem *serveraccess2*
Router(config-ext-nacl)#no 20	Smaže řádku 20 ze seznamu.
Router(config-ext-nacl)#exit	Návrat do globálního konfiguračního režimu.

Tipy pro číslování řádek

- Pořadová čísla začněte od 10 a přidávejte na každé další řádce číslo o 10 více.
- Pokud zapomenete připsat pořadové číslo, je řádka přidána na konec seznamu.
- Při restartu směrovače se pořadová čísla přečíslují, aby odpovídala zásadám ikrementace po 10 (tip 1). Pokud jste v ACL měli čísla 10, 20, 30, 32, 40, 50 a 60 , po restartu tato čísla budou 10, 20, 30, 40, 50, 60, 70.
- Ve výstupech příkazů Router#show running-config nebo Router#show startup-config se pořadová čísla řádků nezobrazují. Vypsat čísla řádků ACL lze následujícími příkazy:
  - Router#show access-lists
  - Router#show access-lists list name
  - Router#show ip access-list
  - Router#show ip access-list list name

Komentáře k řádkům v ACL

Router(config)#access-list 10 remark only Jones has access	Příkaz **remark** dovolí vložení komentáře (omezeného na 100 znaků).
Router(config)#access-list 10 permit 172.16.100.119	Tuto řádku čteme jako: „Hostitel 172.16.100.119 má povolen průchod sítí“.
Router(config)#ip access-list extended telnetaccess	Vytvoří pojmenovaný ACL s názvem telnetaccess a přejde do režimu konfigurace pojmenovaného ACL.
Router(config-ext-nacl)#remark do not let Smith have telnet	Příkaz **remark** dovolí vložení komentáře (omezeného na 100 znaků).
Router(config-ext-nacl)#deny tcp host 172.16.100.153 any eq telnet	Tuto řádku čteme jako: „Tento konkrétní hostitel 172.16.100.153 má zakázán přístup Telnetem do libovolného místa v síti“.

- **TIP:** Příkaz **remark** můžete použít do libovolného číslovaného standardního IP, číslovaného rozšířeného nebo pojmenovaného IP ACL.
- **TIP:** Příkaz **remark** můžete použít buď před nebo po příkazové sekvenci permit nebo deny. Proto buďte konzistetní v umisťování, abyste předešli zmatku, ke které řádce se ten který

komentář vztahuje.

Omezení přístupu k virtuálnímu terminálu

Router(config)#access-list 2 permit host 172.16.10.2	Hostiteli 172.16.10.2 povolí přístup k tomuto směrovači Telnetem v závislosti na tom, kde je tento ACL aplikován.
Router(config)#access-list 2 permit 172.16.20.0 0.0.0.255	Povolí komukoliv z rozsahu adres 172.16.20.x přístup k tomuto směrovači Telnetem v závislosti na tom, kde je tento ACL aplikován.
	Implicitní příkazová řádka **deny any** zakáže komukoliv dalšímu přístup Telnetem.
Router(config)#line vty 0 4	Přejde do režimu konfigurace linky vty.
Router(config-line)access-class 2 in	Aplikuje tento ACL na všech 5 virtuálních rozhraní v příchozím směru.

- **TIP:** Pro omezení přístupu virtuálním rozhraním vty Telnet používejte příkaz **access-class** místo příkazu **access-group**, který slouží k aplikaci ACL na fyzické rozhraní.

Použitá literatura

- Kolektiv: Online kurikulum CCNA Exploration – Routing Protocols and Concepts verze 4.0 (aktuální verze pro registrované uživatele je dostupná na portálu cisco.netacad.net)
- Kolektiv: Course Booklet CCNA Exploration – Routing Protocols and Concepts verze 4.0, Cisco Press 2009
- Prezentace PowerPoint k jednotlivým kapitolám kurikula (pro registrované instruktory jsou dostupné na portálu cisco.netacad.net)
- GRAZANI, Rick a JOHNSON, Allan: CCNA Exploration Companion Guide – Routing Protocols and Concepts, Cisco Press 2008
- JOHNSON, Allan: CCNA Exploration Labs and Study Guide – Routing Protocols and Concepts, Cisco Press 2008
- SCOTT, Empson: CCNA Portable Command Guide, Cisco Press 2007 (v roce 2009 vyšel český překlad v nakladatelství Computer Press)
- Kolektiv: jednotlivá RFC ke zmiňovaným protokolům: http://www.ietf.org/rfc.html .

Obsah