Vybrané statě z oblasti problematiky počítačových sítí na bázi zařízení Cisco

doc. Ing. Jaroslav Dočkal, CSc.
Ing. Josef Kaderka, Ph.D.

Masarykova univerzita
Brno 2011

Recenzovali:

Název: Vybrané statě z oblasti problematiky počítačových sítí na bázi zařízení Cisco
Autoři: Ing. Josef Kaderka, PhD., doc. Ing. Jaroslav Dočkal, CSc.
Typ publikace: studijní texty
Vydavatel: Fakulta informatiky Masarykovy univerzity
Tisk:
Vydáno v roce: 2011
Vydání: první
ISBN: 978-80-904257-2-9

OBSAH

1. Všeobecný popis operačního systému IOS

Zařízení firmy Cisco Systems, typicky směrovače, přepínače, přístupové body apod. jsou zpravidla řízeny operačním systémem IOS (Internetwork Operating System). V některých případech se lze setkat i s jinými operačními systémy, zpravidla se jedná o původní řešení firem, které byly firmou Cisco převzaty (např. CatOS u starších přepínačů, resp. PIX OS u firewallů apod.). Tyto operační systémy bývají časem nahrazeny příslušnou verzí IOSu. Operační systém IOS je dodáván ve velmi široké škále verzí a mutací, podle potřeb zákazníka. Pro výběr existují navigační pomůcky, které umožňují zvolit přesnou verzi IOSu podle zadaných kritérií (např. typu hardware, požadovaných vlastností, kapacit pamětí atd.). Je třeba říci, že IOS nelze stáhnout volně, nýbrž že se jedná o plně komerční produkt. V případě požadavku na nové funkce apod. je třeba jednat s příslušným dodavatelem.

Z hlediska architektury je IOS operačním systémem unixového typu (včetně systému souborů) a má tedy monolitickou architekturu. Všechny procesy sdílejí stejný paměťový prostor, a tudíž neexistuje žádná ochrana oblasti paměti užívané jedním procesem před modifikací jiným procesem. To znamená, že chyby v kódu IOSu mohou potenciálně poškodit data, používaná jinými procesy. Klasický IOS není preemptivní, jeho plánovač procesů patří do kategorie „run to completion“. To znamená, že jádro nemůže přerušit běžící proces – aby mohl být spuštěn jiný proces, musí aktuálně běžící proces sám uvolnit procesor, tj. zavolat příslušnou službu kernelu.

U produktů jako je Cisco CRS-1, od kterých je vyžadována velmi vysoká dostupnost, není toto akceptovatelné (navíc mladší konkurenční operační systémy takové omezení nemají). Firma Cisco proto vyvinula novou verzi Cisco IOS s názvem IOS XR, která nabízí modularita a ochranu paměti mezi procesy, odlehčená vlákna, preemptivní plánování a schopnost samostatného restartu havarovaných procesů. IOS XR pracuje v reálném čase zejména díky využití mikrokernelu (od firmy QNX) a většina kódu současného IOS byla přepsána tak, aby mohla využít funkce, které nové jádro nabízí. Architektura mikrokernelu odstraňuje z jádra všechny procesy, které v něm nejsou absolutně nezbytně a vykonává je jako každý jiný aplikační proces. Prostřednictvím této metody je IOS XR schopen dosáhnout vysoké dostupnosti požadované pro nové, výkonné páteřní přepínače a směrovače.

Uživatel komunikuje s operačním systémem IOS prostřednictvím specializovaného shellu, tj. řádkových příkazů. Shell je vybaven účinnou, kontextově orientovanou nápovědou, která umožňuje zadávání příkazů, jejich voleb či argumentů i bez detailní znalosti příkazů samotných či jejich syntaxe. V kterémkoliv okamžiku lze vložit znak otazník a shell zobrazí možné pokračování. Veškeré příkazy, volby či argumenty lze zkrátit na tolik znaků, aby byla vložená sekvence jednoznačná. Klávesou tabelátor lze vyžádat doplnění zadaného řetězce znaků buď až do konce příkazu nebo do hranice jednoznačnosti.

Příklad – souhrnný výpis seznamu rozhraní a jejich stavů

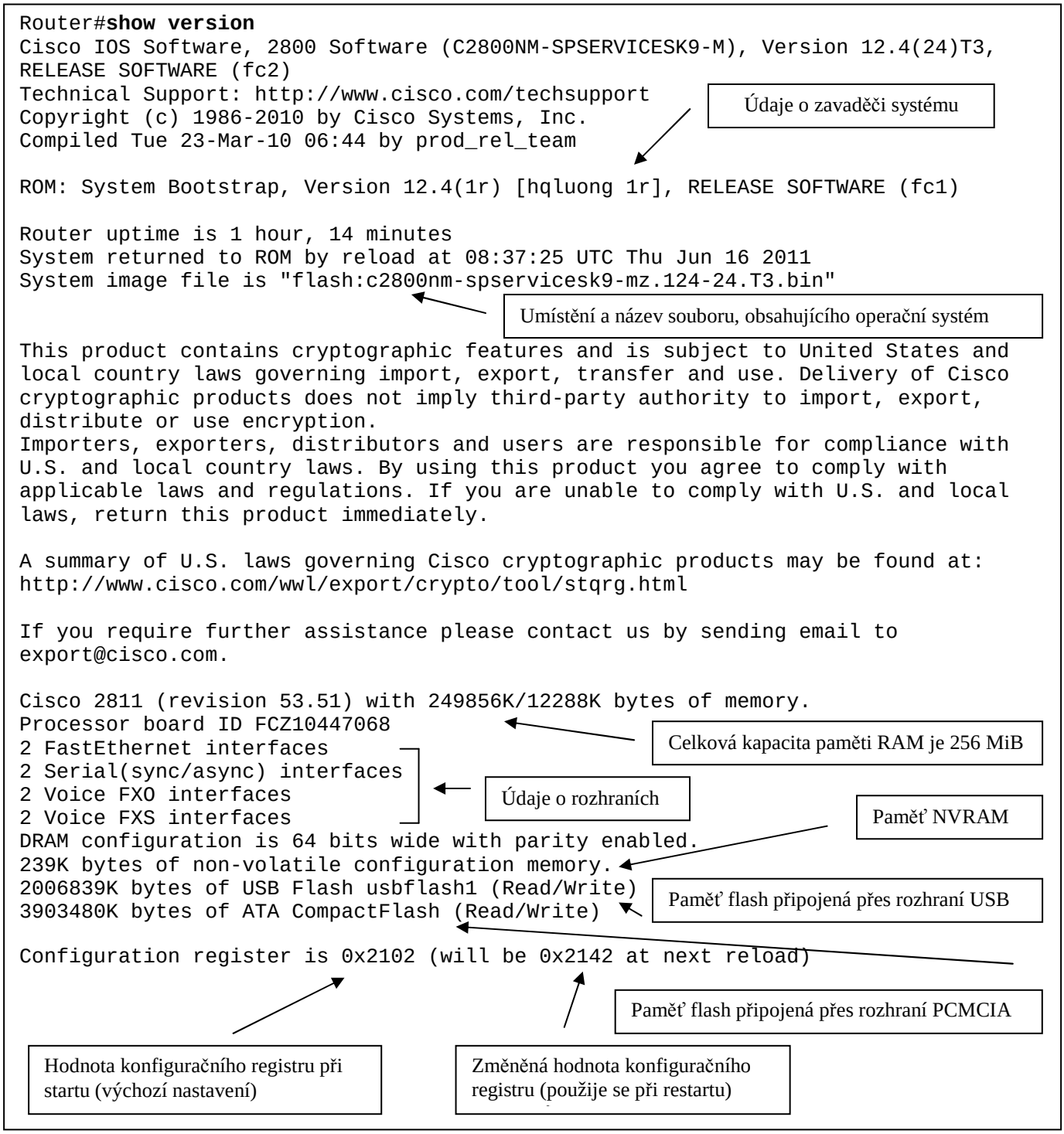
```
show ip interface brief
```

Zkrácená varianta téhož

```
sh ip int b
```

2. **Základní přehled o zařízení**

K získání základního přehledu o zařízení poslouží příkaz `show version` . Pomocí něj lze získat údaje o zavaděči systému, zdroji, z něhož byl zaveden operační systém, operační paměti, rozhraních, paměti NVRAM, paměti flash (interní, připojené přes rozhraní USB nebo PCMCIA/Compact Flash). O pamětech podrobněji viz dále.



3. **Paměti**

U zařízení firmy Cisco se lze setkat s celkem čtyřmi či pěti druhy pamětí; ne každé zařízení používá všechny:

- paměť typu ROM – slouží pro základní start zařízení. V minulosti bývala uložena v patici, umožňující její výměnu (ta byla u směrovačů řady 2500 nutná při použití nových typů pamětí flash). Paměť typu ROM obsahuje:
  - program pro základní diagnostiku hardware (POST – Power On Self Test),
  - zavaděč systému (Bootstrap)
  - ROM Monitor – nástroj obsahující omezenou sadu funkcí (příkazů), umožňující nejzákladnější manipulaci se zařízením. Aktivuje se při startu zařízení buď ručně (stiskem klávesy Break nebo kombinace Ctrl+Break) či automaticky při zjištění havarijního stavu, např. při poškození operačního systému,
  - Rxboot – jen u některých typů zařízení, obsahuje tzv. minimální IOS, což je omezená verze IOSu, umožňující částečnou provozuchopnost směrovače i v případě, nelze-li zavést IOS z regulérního zdroje.
  - údaje o paměti ROM získat příkazem `show version`
- paměť typu Flash (původně EPROM) – typicky obsahuje IOS a jeho doprovodné soubory; případně i jiné obecné soubory, uložené např. administrátorem. Příkladem mohou být různé experimentální varianty konfiguračních souborů (u přepínačů se zde nalézá standardní startovací konfigurační soubor, který je u směrovačů umístěn v NV RAM, viz dále). Část této paměti bývá pevnou součástí základní desky zařízení, zbývající část jako výměnný modul. Ten se buď umisťuje do slotů na základní desce (což vyžaduje demontáž skříně zařízení), nebo u některých zařízení do slotu PCMCIA; novější zařízení bývají vybavena rozhraním USB. Má-li být provozován na funkce bohatý IOS, je třeba pořídit paměť flash (a RAM) o potřebné kapacitě.
  - Údaje o paměti flash lze získat příkazy `show flash:, dir flash:, dir usbflash1:`
- paměť NVRAM (Non-Volatile Random Access Memory), tj. trvale udržující data; i bez napájení. Vyskytuje se u směrovačů, uchovává startovací konfigurační soubor. Původně byla realizována samostatným integrovaným obvodem, který je nyní nejčastěji emulován,
  - údaje o paměti NVRAM lze získat příkazy `dir nvram:`
- paměť typu RAM – operační systém ji dělí do dvou základních částí. První slouží bezprostředně jemu samotnému, dále je zde uložena aktuální konfigurace a některé tabulky (směrovací, ARP aj.). Druhá část je použita jako vyrovnávací paměť pro příjem paketů. Obdobně jako u paměti flash je část paměti RAM na základní desce, zbytek pak realizován formou modulů. Náročnější verze IOSu vyžadují větší kapacitu paměti RAM,
  - údaje o paměti RAM lze získat příkazy `show memory`
- za poslední typ paměti lze do jisté míry považovat externí zařízení, přístupné pomocí protokolu tftp (či ftp nebo i jinak). Z tftp serveru lze zavádět jak operační systém, tak i konfiguraci.

4. **Konfigurační registr**

Konfigurační registr o délce 16 bitů slouží v dále popisovaném smyslu využíván u směrovačů. Pomocí něj lze ovlivnit např. průběh startu směrovače, tj. rozhodnout zda, odkud a jaký operační systém zavést, zda a odkud načíst konfigurační soubor, jaké mají být komunikační parametry konsolového rozhraní atd. Pozor – nastavit lze jakoukoliv kombinaci bitů, ale ne všechny mají význam; doporučuje se používat jen ty, které význam mají. Co přesněji konfigurační registr umožňuje a jaké důvody mohou vést ke změně jeho hodnoty:

- vynutit start systém do prostředí do ROM monitoru nebo boot ROM,
- vybrat zdroj zavedení IOSu a implicitní název souboru, který jej obsahuje,
- povolit nebo zakázat funkci klávesy Break za chodu operačního systému,
- definovat adresu pro rozhlašování,
- nastavit přenosovou rychlost konzoly (terminálu),
- načíst IOS z paměti flash,
- povolit zavádění IOSu po sítí z TFTP serveru (Trivial File Transfer Protocol),
- překonat neznámé heslo,
- ručně nastartovat systém pomocí příkazu `boot` z prostředí ROM monitoru,
- vynutit automatické zavedení IOSu (boot image) z ROM nebo flash a načíst konfiguraci uloženou v konfiguračním souboru v NVRAM.

S obsahem konfiguračního registru lze manipulovat dvěma cestami:

- na úrovni ROM monitoru – například při nemožnosti zavést operační systém, při obnově hesla po jeho ztrátě apod.,
  - příkaz `confreg 0xABCD`, kde `ABCD` jsou hexadecimální číslice, typické hodnoty jsou:
    - `0x2102` (zavést IOS z flash, načíst konfiguraci z NVRAM)
    - `0x2142` (zavést IOS z flash, nenačíst žádnou konfiguraci)
- v globálním konfiguračním módu
  - `Router(config)#config-register 0x2102`

Bit	**Význam** (hodnota bitu = 1)
15	Povolit diagnostická hlášení a ignorovat obsah NVRAM
14	IP rozhlašování neobsahují čísla sítí
13	Nelze-li zavést software ze sítě, pak jej zavést z flash
12	Nastavení rychlosti konsoly (*1)
11	
10	IP protokol používá v rozhlašovací adrese bity 0
9	Použít sekundární zavaděč
8	Ignorovat stisk klávesy Break při běhu IOSu (při opačném nastavení by stiskem této klávesy byl IOS okamžitě ukončen!)
7	Povolen OEM bit
6	Systémový software ignoruje obsah NVRAM
5	Rychlost konsoly je vyšší než 9600 Bd (*2)
4	Rychlý start (jen u řady 7000)
3	Chování se směrovače po startu, např. zdroj zavádění operačního systému (*3)

2	
1	
0	

Poznámky

(*1), (*2)

Nastavení přenosové rychlosti konsolové linky			
Bit 5	Bit 12	Bit 11	Rychlost (Bd)
0	1	0	1200
0	1	1	2400
0	0	1	4800
0	**0**	**0**	**9600**
1	0	0	19200
1	0	1	38400
1	1	0	57600
1	1	1	115200

(*2)
Vyšší přenosové rychlosti až 115 200 Bd lze použít jen u novějších zařízení. Přenosová rychlost se nastavuje se buď v prostředí ROM monitoru příkazem `confreg` (dále je nutno projít dialogem) nebo v specifickém módu konfigurace konsolové linky příkazem `speed`. Příklady jsou uvedeny níže.

```
rommon 1 > confreg

    Configuration Summary
enabled are:
load rom after netboot fails
console baud: 9600
boot: image specified by the boot system commands
      or default to: cisco2-C3600

do you wish to change the configuration? y/n  [n]:  y
enable  "diagnostic mode"? y/n  [n]:

!--- Pressing "Enter" accepts the default (value between the brackets).

enable  "use net in IP bcast address"? y/n  [n]:
disable  "load rom after netboot fails"? y/n  [n]:
enable  "use all zero broadcast"? y/n  [n]:
enable  "break/abort has effect"? y/n  [n]:
enable  "ignore system config info"? y/n  [n]:
change console baud rate? y/n  [n]:  y
enter rate: 0 = 9600,  1 = 4800,  2 = 1200,  3 = 2400
            4 = 19200, 5 = 38400, 6 = 57600, 7 = 115200  [7]:  7
change the boot characteristics? y/n  [n]:

    Configuration Summary
enabled are:
load rom after netboot fails
console baud: 115200
boot: image specified by the boot system commands
      or default to: cisco2-C3600

do you wish to change the configuration? y/n  [n]:

You must reset or power cycle for new config to take effect
```

```
Router(config)#line con 0
Router(config-line)#speed 115200
```

Změna přenosové rychlosti konsolové linky má smysl pouze v případě nezbytnosti. Příkladem může být situace, kdy zařízení po zapnutí opakovaně přechází do režimu ROM monitor a je zjevné, že došlo k poškození či smazání IOSu. Tento je třeba obnovit, což může být významný problém u směrovače, který nemá ethernetové rozhraní nebo u jakéhokoliv přepínače. Pokud by totiž zařízení nemělo ani rozhraní USB nebo PMCIA, pak by nezbývalo než nakopírovat IOS do paměti flash přes konsolovou linku (protokolem X-modem), což by při rychlosti 9600 Bd trvalo i řadu hodin. Poslední možnost by spočívala v demontáži krytu zařízení a vložení modulu paměti flash s IOSem; tento modul by bylo třeba připravit externě. Po úkonu je třeba nezapomenout vrátit přenosovou rychlost konsolového portu zpět na výchozí hodnotu 9600 Bd, neboť by se mohlo stát, že by jiná obsluha o změně rychlosti nevěděla a pokoušela by se pracovat se standardním nastavení terminálu či emulačního programu. V takovém případě by zařízení nereagovalo a mohl by vzniknout dojem jeho úplné nefunkčnosti. Některá zařízení sice mívají možnost resetu nastavení konsolové linky, to ale vyžaduje demontáž krytu a přesun zkratovací spojky (jumperu), což zpravidla přesahuje znalosti a možnosti běžné obsluhy.

(*3)

Význam bitů 0-3 konfiguračního registru	
Hodnoty (hexa)	Význam
0	Po zapnutí přechod do ROM monitoru
1	Aktivace boot helperu, resp. přechod do režimu ROM monitoru – umožňuje zavedení IOSu pomocí příkaz prostřednictvím sítě (jen některá zařízení).
2-F	Použít příkaz `boot` (a jeho parametry), uložený v konfiguračním souboru; není-li tam uveden, pak zavést IOS z paměti flash; je-li jich tam více, pak první.

Vybrané varianty hodnot konfiguračního registru	
Hodnota	Chování směrovače
0x102	Ignorovat stisk klávesy Break, rychlost konsolové linky 9600 Bd
0x1202	Rychlost konsolové linky 1200 Bd
0x2101	Aktivovat zavaděč, ignorovat stisk klávesy Break, přejít do ROM monitoru pokud nelze zavést IOS, rychlost konsolové linky 9600 Bd
0x2102	Ignorovat stisk klávesy Break, přejít do ROM monitoru pokud nelze zavést IOS, rychlost konsolové linky 9600 Bd; **výchozí nastavení pro většinu zařízení**
0x2120	Přejít do ROM monitoru, rychlost konsolové linky 19200 Bd
0x2122	Ignorovat stisk klávesy Break, přejít do ROM monitoru pokud nelze zavést IOS, rychlost konsolové linky 19200 Bd
0x2124	Zavést IOS prostřednictvím sítě (tftp), ignorovat stisk klávesy Break, přejít do ROM monitoru pokud nelze zavést IOS, rychlost konsolové linky 19200 Bd
0x2142	Ignorovat stisk klávesy Break, přejít do ROM monitoru pokud nelze zavést IOS, rychlost konsolové linky 9600 Bd, ignorovat obsah NVRAM; **při obnově hesla aj.**
0x2902	Ignorovat stisk klávesy Break, přejít do ROM monitoru pokud nelze zavést IOS, rychlost konsolové linky 4800 Bd
0x2922	Ignorovat stisk klávesy Break, přejít do ROM monitoru pokud nelze zavést IOS, rychlost konsolové linky 38400 Bd
0x3122	Ignorovat stisk klávesy Break, přejít do ROM monitoru pokud nelze zavést IOS, rychlost konsolové linky 57600 Bd
0x3902	Ignorovat stisk klávesy Break, přejít do ROM monitoru pokud nelze zavést IOS, rychlost konsolové linky 2400 Bd
0x3922	Ignorovat stisk klávesy Break, přejít do ROM monitoru pokud nelze zavést IOS, rychlost konsolové linky 115200 Bd

5. **Některé problémové situace a jejich řešení**

Neznámá nebo zapomenutá hesla (Password Recovery)
Neznámé či zapomenuté heslo pro přístup na konsolu nebo pro přechod do privilegovaného režimu představuje problém, neboť zařízení bez jeho znalosti nelze spravovat. Tuto situaci lze v naprosté většině případů snadno vyřešit, vyžaduje se však fyzický přístup k zařízení (a samozřejmě konsola – dnes typicky emulovaný terminál, tj. PC se sériovým portem RS232, zpravidla vyhoví i převodník USB-RS232). Příslušný postup se označuje jako „Pasword Recovery“, je principiálně stejný jak u směrovačů, tak přepínačů, byť se v detailech může lišit (zejména u přepínačů), a je detailně popsán v dokumentace či na webových stránkách firmy Cisco.
Pozor – u některých zařízení lze nastavit vyšší míru bezpečnosti, při ní nelze (snadno) přejít do ROM monitoru, což situaci výrazně komplikuje. Tuto vyšší míru bezpečnosti je vhodné volit jen tehdy, jsou-li pro to opravdu mimořádné důvody.

Neznámé nebo zapomenutá heslo u směrovače	
1. Přerušit start pomocí konsoly (vyžaduje se fyzické přístup)	**<Ctrl><Break>**
2. Zavést IOS z flash, nenačítat konfigurační soubor z NVRAM	**confreg 0x2142**
2a. Jiná syntaxe platná u starších zařízení	**o/r 0x2142**
3. Restart operačního systému	**Reset**
4. Přejít do privilegovaného módu; nenačtením konfigurační souboru lze provést bez hesla	**Enable**
5. Nyní v privilegovaném módu překopírovat konfigurační soubor z NVRAM do RAM	**copy startup-config running-config**
6. Změnit enable hesla na "NoveHeslo" (případně provést další operace)	**enable password NoveHeslo**
7. Uložit aktuální konfiguraci do NVRAM (tj. s novým heslem)	**copy running-config startup-config**
8. Příští start směrovače nechť proběhne normálně (IOS z flash, konfigurační soubor z NVRAM)	**config-reg 0x2102**

Neznámé nebo zapomenutá heslo u přepínače – varianta	
1. Vypnout napájení přepínače	
2. Stisknout a držet tlačítko "Mode" na předním panelu přepínače	**<mode>**
3. Zapnout napájení přepínače	
4. Po zhasnutí STAT LED uvolnit tlačítko "Mode"	
5. Vyčkat ukončení výpisu a na přechod do ROM monitoru	
6. Zadat sekvenci příkazů	**flash_init** **load_helper**
7. Přejmenovat konfigurační soubor (je uložen ve flash, ne v NVRAM)	**rename flash:config.text flash:config.old**
8. Zavést operační systém přepínače	**Boot**
9. Přeskočit konfigurační dialog, přejít do privilegovaného módu	**Enable**
10. Obnovit konfigurační soubor	**rename flash:config.old flash:config.text**

11. Načíst uloženou konfiguraci, tj. se starým heslem	**copy startup-config running-config**
12. Nastavit nové heslo pro přechod do privilegovaného módu	**enable secret NoveHeslo**
13. Uložit aktuální konfiguraci, tj. s novým heslem	**copy running-config startup-config**

Chybějící nebo poškozený operační systém

Pokud zařízení po zapnutí nabíhá do ROM monitoru, je pravděpodobně smazán nebo poškozen operační systém IOS (není-li ovšem toto chování způsobeno nastavením konfiguračního registru).

Pro jeho obnovu třeba mít jeho záložní kopii (nutno pořídit předem!), jinak nezbývá než obstarat jiný, stejného typu a verse, například od dodavatele (IOS nelze volně stáhnout). V nouzi vyhoví IOS z jiného směrovače téhož typu.

Dojde-li ke smazání IOSu z paměti flash, ale zařízení dosud běží (tj. IOS je aktivní v paměti RAM), je nejlépe zachovat klid a postupovat standardně, tedy překopírovat záložní IOS běžným postupem z tftp serveru – **copy tftp flash**,

U zařízení s výměnnou pamětí flash nebo USB portem lze IOS zapsat na dané médium například v osobním počítači, což je pravděpodobně nejjednodušší a nejrychlejší postup.

Obnova chybějícího nebo poškozeného operačního systému IOS u směrovače	
Ověřit (ne)přítomnost IOSu v paměti flash	**rommon 1 > dir flash:** **device does not contain a valid magic number** **dir: cannot open device "flash:"**
Připojit ethernetové rozhraní s nejnižším ID (např. fa0/0); u směrovače bez Ethernetu je nutno použít konsolový port a X-modem. Ověřit nastavení potřebných proměnných (viz příklad). Neexistují-li nebo jejich hodnoty nevyhovují, změnit je	**rommon 2 > set** **IP_ADDRESS=172.18.16.76** **IP_SUBNET_MASK=255.255.255.192** **DEFAULT_GATEWAY=172.18.16.65** **TFTP_SERVER=172.18.16.2** **TFTP_FILE=c2600-ik9o3s3-mz.123-13.bin**
Příklad nastavení/změny hodnoty proměnné	**TFTP_SERVER=172.18.16.88**
Spustit stahování a instalaci IOSu	**tftpdnld**
Restartovat směrovač	**reset**

Obnova chybějícího nebo poškozeného operačního systému IOS u přepínače	
K připojení ke konsolovému rozhraní použít emulátor terminálu, který podporuje protokol X-modem (Hyperterminal, Teraterm – nikoliv putty)	
Volitelné – nastavit vyšší rychlost konsolového rozhraní (pak i emulátoru terminálu) pro urychlení přenosu	**switch: confreg** (vyvolá se dialog), nebo **switch: set BAUD 115200**
Aktivovat systém souborů. Aktivovat pomocný zavaděč	**switch: flash_init** **switch: load_helper**

Ověřit (ne)přítomnost IOSu v paměti flash; zde IOS chybí	**Switch: dir flash:** **2 −rwx 556 Mar 01 1993 00:00:35 vlan.dat** **4 −rwx 1595 Mar 01 1993 19:52:41 config.text**
Spustit instalaci na straně přepínače	**copy xmodem: flash:c3550-ipservicesk9-mz.122-25.see.bin**
Spustit instalaci na straně terminálu (Hyperterminal) prostřednictvím výběru z položek menu Pozor, vzhledem k nízké přenosové rychlosti konsoly může operace trvat i hodiny!	**Transfer** **Send File** **Browse** (nalézt soubor, např. c3550-ipservicesk9-mz.122-25. bin) **Send**
Spustit operační systém	**boot flash: c3550-ipservicesk9-mz.122-25.bin**
Vrátit výchozí přenosovou rychlost konsolového portu i terminálu	**set BAUD 9600** nebo **unset BAUD**

Případová studie
Přepínač po zapnutí nabíhal do ROM monitoru s tím, že vypisoval uvedené chybové hlášení.

```
...done initializing flash.
Boot Sector Filesystem (bs:) installed, fsid: 3
Parameter Block Filesystem (pb:) installed, fsid: 4
Loading "flash://lost+found/00004"...############

Error loading "flash:/lost+found/00004"
Interrupt within 5 seconds to abort boot process.
Boot process failed...
```

Jednalo se tedy o neúspěšný pokus zavést IOS ze souboru `00004`, umístěného v paměti `flash`, v adresáři `lost+found`. To je sice neobvyklé jméno souboru i adresáře, ale pokud by soubor `00004` skutečně obsahoval IOS, byl by zaveden. V daném případě však zřejmě soubor `00004` operační systém neobsahuje. Silnou indicii poskytuje adresář `lost+found`; takový adresář vytváří operační systém Unix v situaci, kdy odhalí nekonzistentnost systému souborů (například v průběhu rutinních kontrol při startu). Do něj pak umisťuje nalezená torza souborů, kterým přiděluje jména právě ve formě čísel.
Je tedy třeba zjistit skutečný obsah paměti flash, resp. zda se v ní nalézá soubor obsahující IOS.

```
switch:
switch: flash_init
  Initializing Flash...
  ...The flash is already initialized.
switch: load_helper

switch: dir flash:
Directory of flash:/

 2  drwx          64  Jan 01 1970 00:00:10 +00:00  lost+found
 3  -rwx     3132032  Jan 01 1970 01:38:13 +00:00  c2950-i6q4l2-
mz.121-22.EA12.bin
 5  -rwx        1735  Mar 01 1993 00:10:43 +00:00  config.text
 6  -rwx           5  Mar 01 1993 00:10:43 +00:00  private-
config.text
```

V paměti flash se tedy mimo adresáře `lost+found` a konfiguračního souboru `config.text` nalézá soubor `c2950-i6q4l2-mz.121-22.EA12.bin`, pravděpodobně obsahující IOS, o čemž svědčí i jeho velikost 3 132 032 B. Nejjednodušším pokračováním je pokusit se manuálně zavést z tohoto souboru IOS.

```
switch: boot flash:/c2950-i6q4l2-mz.121-22.EA12.bin
```

Experiment potvrdil hypotézu, v paměti flash se opravdu nalézal funkční IOS (výpis je zkrácen).

```
C2950 Boot Loader (C2950-HBOOT-M) Version 12.1(11r)EA1, RELEASE
SOFTWARE (fc1)
Compiled Mon 22-Jul-02 17:18 by antonino
WS-C2950T-24 starting...
...
Loading

"flash:/c2950-i6q4l2-mz.121-
22.EA12.bin"...########################################
################################################################################
################################################################################
##########################

File "flash:/c2950-i6q4l2-mz.121-22.EA12.bin" uncompressed and
installed, entry point: 0x80010000
...
S3> enable
S3#
```

Tím se situace podstatně zjednodušila, neboť lze plně využít příkazů IOSu (resp.jeho shellu) k manipulaci se systémem souborů.

```
S3#dir flash:lost+found
Directory of flash:/lost+found/
 4  -rw-      366080  Jan 01 1970 00:00:10 +00:00  00004
7741440 bytes total (4239360 bytes free)
```

Je tudíž velmi pravděpodobné, že adresář `lost+found` stejně jako soubor `00004` vytvořil IOS, když při svém startu nalezl v paměti flash nekonzistentní záznamy v tabulkách popisujících alokaci dat na tomto médiu. Do tohoto souboru umístil datové bloky, které byly označeny jako obsazené, avšak současně nenáležely žádnému existujícímu souboru. Zmíněná situace mohla nastat při při nedokončené modifikaci systémů souborů (například při vypnutí přepínače v okamžiku ukládání konfiguračního souboru, ať již manuálního nebo protokolem vtp).
Protože zavaděč IOSu při hledání souboru s IOSem postupuje abecedně podle jmen souborů, narazil jako na první na soubor `0004` a pokusil se z něj zavést IOS, což se nepovedlo. Jako nejjednodušší řešení se v dané situaci jeví soubor `00004` i adresář `lost+found` smazat.

```
S3#delete flash:lost+found/00004
Delete filename [lost+found/00004]?
Delete flash:lost+found/00004? [confirm]
S3#

S3#rmdir flash:lost+found
Remove directory filename [lost+found]?
Delete flash:lost+found? [confirm]
Removed dir flash:lost+found
```

Následný hladký restart operačního systému IOS ukázal, že problém byl vyřešen.

```
S3#reload
Proceed with reload? [confirm]

00:03:18: %SYS-5-RELOAD: Reload requested

C2950 Boot Loader (C2950-HBOOT-M) Version 12.1(11r)EA1, RELEASE
SOFTWARE (fc1)
Compiled Mon 22-Jul-02 17:18 by antonino
WS-C2950T-24 starting...
...
...done initializing flash.
Boot Sector Filesystem (bs:) installed, fsid: 3
Parameter Block Filesystem (pb:) installed, fsid: 4
Loading
"flash:/c2950-i6q4l2-mz.121-
22.EA12.bin"...########################################
################################################################################
################################################################################
##########################

File "flash:/c2950-i6q4l2-mz.121-22.EA12.bin" uncompressed and
installed, entry point: 0x80010000
S3>
```

6. Protokol IPv6 v prostředí systémů Cisco

Problematika protokolu IPv6 je velmi rozsáhlá (viz [Satrapa]). V tomto textu proto nebudou rozebírány detaily, ale ukázány některé typické situace.

Přesto je vhodné připomenout některá základní fakta:

- protokol IPv6 leží na 3. vrstvě referenčního modelu síťové architektury dle ISO/OSI. S ním sousedící vrstvy, tj. druhá a čtvrtá, zůstávají nezměněny. Protokol IPv6 tedy pracuje např. nad protokolem Ethernet a přenáší údaje protokolů TCP či UDP,
- zásadní odlišnost mezi protokolem IPv4 a IPv6 spočívá v délce adresy, která u IPv6 činí 128 bitů. Ostatní rozdíly jsou z hlediska funkce málo významné,
- každé rozhraní bude mít více IPv6, viz dále,
- protokol IPv6 nyní používá dva druhy adres: globální a linkové lokální. Globální adresa je přidělena centrálně a je unikátní. Linková lokální (Link Local) má vždy místní význam, přiděluje si ji každý IPv6 uzel sám a umožňuje snadnou výměnu některých informací (například dohodu mezi směrovači.)

Protokol IPv6 není kompatibilní s protokolem IPv4, proto vyvstává problém přechodu od IPv4 k IPv6. Tento přechod bude, jak se ukazuje, trvat dlouhá léta a v zásadě může mít tyto podoby:

- opuštění IPv4 a přechod výhradně na IPv6 – lze realizovat jen v uzavřených, spíše menších a přísně centrálně spravovaných sítích,
- využití dvojího zásobníku – každý systém by současně používal jak IPv4, tak IPv6, což by se týkalo jak koncových počítačů, tak i směrovačů. Fakticky by se jednalo o jistou formu návratu k multiprotokolovým řešením (viz koexistence protokolů IP a IPX),
- tunelování – vytvoření IPv6 ostrovů (či jednotlivých počítačů), propojených prostřednictvím IPv4 sítí.

Ukázka základní konfigurace směrovače 1841, osazeného navíc čtyřportovým modulem, realizujícím přepínač. (Jednotlivá rozhraní Fa0/1/0 – Fa0/1/3 tohoto modulu lze konfigurovat jen na 2. vrstvě, nemohou tedy mít individuální IP adresy. Přístup k tomuto rozhraní na 3. vrstvě je možný prostřednictvím vnitřního rozhraní Vlan 1)

Manuální konfigurace rozhraní

```
Router> enable
Router# conf term
Router(config)# ipv6 unicast-routing
Router(config)# ipv6 cef
Router(config)# interface Vlan 1
Router(config-if)# ipv6 address 2001:0DB8:0:3::1/64
Router(config-if)# ipv6 nd prefix 2001:0DB8:0:3::/64
Router(config-if)# other-config-flag
Router(config-if)# ipv6 enable
Router(config-if)# no shutdown
Router(config-if)# exit
```

Konfigurace směrovacího protokolu RIPng

```
Router(config)# ipv6 router rip test
Router(config-rtr)# exit

Router(config)# interface FastEthernet 0/0
Router(config-if)# ipv6 address 2001:0DB8:0:2::1/64
Router(config-if)# ipv6 nd prefix 2001:0DB8:0:2::/64
Router(config-if)# ipv6 nd other-config-flag
Router(config-if)# ipv6 enable
Router(config-if)# ipv6 rip test enable
Router(config-if)# exit
Router(config)# interface FastEthernet 0/1
Router(config-if)# ipv6 address 2001:0DB8:0:3::1/64
Router(config-if)# ipv6 nd prefix 2001:0DB8:0:3::/64
Router(config-if)# ipv6 nd other-config-flag
Router(config-if)# ipv6 enable
Router(config-if)# ipv6 rip test enable
Router(config-if)# end
```

7. AAA – Authentication, Authorization, Accounting (autentizace, autorizace a účtování)

Autentizace, autorizace a účtování jsou formy zabezpečení nezbytné při řízení vzdáleného přístupu k místnímu směrovači či zdrojovým datům. AAA je navrženo tak, aby administrátor mohl tyto služby nastavit buď globálně nebo pomocí konfigurace konkrétní linky či rozhraní.

Aktivace AAA automaticky deaktivuje jiné formy řízení přístupu. To znamená, že nelze současně použít jiné příkazy, například `login local` a `login`. Tyto starší typy řízení byly zaměřeny především na ověření uživatele, avšak AAA umožňuje řídit každý přístup uživatele ke zdrojovým datům a dále poskytuje vytváření účtů mimo tento směrovač. Pomocí AAA lze selektivně využít zabezpečovacích služeb a informací, které náleží danému směrovači. Cisco podporuje služby RADIUS, TACACS+ i Kerberos. Mimo služeb týkajících se zabezpečení sítě AAA umožňuje uchovávat ověřovací rozhodnutí o lokální uživatelské databázi směrovače, heslech linek a přístupových heslech. Při použití AAA spolu se zabezpečovacím serverem lze z jednoho místa řídit přístup ke směrovačům a ostatním síťovým zařízením.

Pro praktickou realizaci AAA je třeba mít k dispozici příslušný server, dále v textu se předpokládá RADIUS a to ve dvou verzích. Pro platformu Linux je to FreeRadius, pro Windows pak WinRadius.

Autentizace:

Jedná se o mechanismus, který slouží k identifikaci uživatelů předtím, než je jim povolen přístup k síťovým složkám a službám. AAA provádí identifikaci pomocí dialogů „přihlášení/heslo“, mechanismy „žádost/odezva“ a podporovaných rozlišovacích technologií. Mají-li být použity protokoly zabezpečovacích serverů nebo záložní metody ověřování, je nutno zároveň použít AAA, i když pro samotné nastavení ověřování není potřeba. Vhodné metody pro AAA ověřování jsou RADIUS, TACACS+, Kerberos, místní uživatelská databáze, hesla linek a přístupová hesla.

Autorizace:

Autorizace slouží k řízení přístupu k systémovým zdrojům. Vymezuje dovolené operace uživatelem poté, co byl prověřen směrovačem. Může jít o jednorázovou autorizaci, autorizaci každé služby a autorizaci každého uživatele. Autorizaci lze nastavit výhradně za použití AAA. Mezi vhodné metody patří RADIUS, TACACS+.

Účtování:

Účtováním se v daném kontextu myslí zejména činnost spojená s 3. vrstvou, tj. zabývá se protokolem IP. Shromažďuje se počet bytů a paketů zpracovaných síťovým zařízením a to na základě zdrojové a cílové IP adresy. Je měřen pouze tranzitní provoz a to při výstupu ze zařízení. Provoz generovaný zařízením nebo v něm ukončený součástí statistiky není. Jsou shromažďovány detaily ohledně dílčích IP adres, takže se lze zabývat jednotlivými uživateli (např. je zpoplatnit, hledat provozní excesy apod.). Pro potřeby operátorů lze v každém okamžiku sejmout snímek provozu. K tomuto účelu jsou udržovány dvě účetní databáze:

- aktivní databáze,
- databáze kontrolních bodů.

V prvním případě se předpokládá použití řádkového příkazu nebo SNMP žádosti pro získání informace o okamžitém stavu zařízení (může jich být i větší počet), ve druhém se tyto charakteristiky periodicky a synchronně snímají a „zmrazují“.

Protokol RADIUS

RADIUS server představuje klíčovou komponentu celého autentizačního systému, která uchovává seznam oprávněných uživatelů a jejich hesel. Na tento server pak přichází autentizační žádosti od jednotlivých klientů (suplikantů), na které server odpovídá pozitivně či negativně.

Existuje řada programového vybavení realizujícího RADIUS server jak pro různá operační prostředí, tak i s různými podmínkami používání – od freeware až po plně komerční produkty. Příklady použití jsou uvedeny v dalším textu, nejprve bez použití protokolu SSH, pak s ním.

FreeRadius

Instalace:

FreeRadius představuje freeware, které lze stáhnout z jeho domovských stránek www.freeradius.org ve formě zdrojových textů nebo již vytvořený balíček (např. rpm):

příkaz: $ *rpm -i název.rpm*

Při instalaci ze zdrojových textů je třeba postupovat standardně, postup je popsán v dokumentech přímo v aplikaci FreeRadius:

příkaz: $ ./configure
$ make
$ make install

FreeRadius se spustí příkazem **radiusd** + další volby (lze je vypsat příkazem radiusd –h, popis v manuálových stránkách), pro otestování lze použít příkaz **radtest** (popis v manuálových stránkách).

Konfigurace:

radiusd.conf:

Základní konfigurační soubor je ***radiusd.conf***. Zde je určeno umístění konfiguračních, souborů, logovacího souboru, dále je zde nastaveno, co se má logovat, bezpečnost aplikace (počet možných připojení, atd).

log_file = ${logdir}/radius.log – toto je defaultní logovací soubor a jeho umístění
log_auth = yes
log_auth_badpass = yes
log_auth_goodpass = yes

clients.conf:

Další konfigurační soubor je ***clients.conf***, kde je nutné přidat klienta (tímto klientem bude jedno z ethernetových rozhraní směrovače, na kterém je připojeno PC s Radius serverem). Dále je uvedeno několik příkladů i defaultních nastavení pro localhost tak, aby bylo možné používat testovací nástroj – radtest.

vlastní nastavení: *client 192.168.2.1 {*

secret = radius
shortname = ethernet
nastype = cisco
}

- *192.168.2.1* je IP adresa rozhraní na směrovači
- *secret = radius* je heslo pro cisco a radius server (ověřují si ho navzájem, proto musí být nastaveno jak na směrovači, tak na radius serveru stejně)
- *shortname = ethernet* je zkratka pro klienta

- *nastype = cisco* Tento příkaz určuje, jaký typ komunikace se použije. Tato komunikace je předdefinována, podle hodnoty tohoto příkazu (např. cisco, computone, livingstone, atd.)

users:

Poslední soubor, který je nutné upravit, je ***users***. Zde jsou definováni uživatelé. Je zde možno nastavit řadu atributů – konkrétní položky a jejich hodnoty je třeba konsultovat s dokumentací. Je zde opět několik příkladů (a popis), takže jde přibližně určit, co jaké nastavení znamená.

metlosh Auth-Type := Local, User-Password == "metlosh"

Auth-Type := Local určuje typ autentizace, v tomto případě je lokální (toto PC) Při nastavení „*System*“ používá pro autentizaci soubor passwd z lokálního PC (tj. ne hodnoty ze souboru *users*!)

User-Password == "metlosh" nastavuje heslo uživatele

Existuje mnoho dalších důležitých nastavení, jako: Fall-Through (určuje, jestli se po porovnání záznamu tohoto uživatele bude pokračovat dále v prohledávání souboru users), Reply-Message (po úspěšné autentizaci vypíše zadanou zprávu), Framed-Protocol, Framed-IP-Address, Framed-IP-Netmask, Framed-Routing, Framed-Filter-Id, atd.

Po tomto nastavení lze vyzkoušet funkčnost pomocí příkazu **radtest** již na nového uživatele.

WinRadius

Tento program lze stáhnout např. od výrobce, http://www.brothersoft.com. WinRadius je komerční program, pro účely ověřování jej lze provozovat 5 hodin zdarma.

Samotné nastavení tohoto serveru může být pro teoreticky méně vyzbrojené administrátory jednodušší nežli u FreeRadius pro Linux. FreeRadius má větší možnosti nastavení a správy.

Instalace:

Je velice jednoduchá, staženou aplikaci stačí rozbalit.

Konfigurace:

1. Spustit instalační program **WinRadius.exe**
2. Kliknout na tlačítko"Configure ODBC automatically" v "Settings/Database...".
3. Restartovat WinRadius. Nyní by mělo být vše nastaveno.

4. Přidat uživatele do WinRadius kliknutím na tlačítko "+". (Nutné jsou první dvě položky – User name a Password, další položky jako Group, Cash prepaid, Address není nutno vyplňovat).

Pak lze funkčnost serveru otestovat a to spuštěním **RadiusTest.exe** a sledováním výpisů v okně serveru.

Cisco směrovač

Je nutné mít IOS s podporou SSH, což je u nových směrovačů pravidlem. Problémy by však mohly být u menších směrovačů, zejména pak u starších, a u přepínačů.

Konfigurace:
Dále se nastaví rozhraní (není nutno popisovat) a parametry k aaa a radius serveru:

```
1. Router(config)# aaa new-model
2. Router(config)# radius-server host 192.168.2.2 auth-port 1812 acct-
   port 1813
3. Router(config)# radius-server retransmit 1
4. Router(config)# radius-server key radius
5. Router(config)# aaa authentication login default group radius
6. Router(config)# aaa authorization exec default group radius
7. Router# debug aaa
8. Router# debug radius
9. Router# debug packet
```

1) Tento příkaz aktivuje novou sadu nastavení (pro AAA) která se používá v IOS od verze 12.0.
2) Definice IP adresy Radius serveru a portů, na kterých bude naslouchat.
3) Nepovinné, počet pokusů o autorizaci.
4) Nastavení klíče (hesla).
10. 5,6) Nastavení skupin pro autentizaci a autorizaci.
11. 7,8,9) Ladění – výpis probíhajících činností. Podle rozdílných IOS se tyto příkazy liší či chybí, např. v některých není debug radius, ale `debug authorization`, `debug authentication`.

Jestliže se po úspěšné autorizaci (připojení se k Cisco směrovači) má měnit konfigurace- tj. přejít do privilegovaného módu, je nutné nastavit heslo:

```
Router(config)# enable password pristup
```

Výpis úplné konfigurace směrovače

```
      Current configuration : 726 bytes
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Router
!
aaa new-model
aaa authentication login default group radius
aaa authorization exec default group radius
```

```
!
ip subnet-zero
!
!
!
!
call rsvp-sync
interface Ethernet0/0
 ip address 192.168.2.1 255.255.255.0
 half-duplex
!
interface Serial0/0
 no ip address
 shutdown
!
interface Ethernet0/1
 ip address 192.168.1.1 255.255.255.0
 half-duplex
!
interface Serial0/1
 no ip address
 shutdown
!
ip classless
ip http server
!
!
radius-server host 192.168.2.2 auth-port 1812 acct-port 1813
radius-server key radius
!
dial-peer cor custom
!
!
!
!
!
line con 0
line aux 0
line vty 0 4
end
```

Pro WinRadius je konfigurace kromě hesla (příkaz *`radius-server key`)* stejná. Po tomto nastavení by se měla podařit autorizace (např. vzdálený přístup pomocí telnetu) Zadáno musí být IP rozhraní, kterým jsme připojeni k směrovači. Pak už budeme tázáni na username a password.

Výpis činnosti směrovače:

Výpis z činnosti směrovače po nastavení `debug aaa, radius, packet` (úspěšný pokus):

```
AAA: parse name=tty66 idb type=-1 tty=-1
00:09:40: AAA: name=tty66 flags=0x11 type=5 shelf=0 slot=0 adapter=0 port=66 channel=0
00:09:40: AAA/MEMORY: create_user (0x826B9C9C) user='NULL' ruser='NULL' ds0=0 po
          rt='tty66' rem_addr='192.168.1.2' authen_type=ASCII service=LOGIN priv=1 initial_task_id='0'
00:09:40: AAA/AUTHEN/START (3251495553): port='tty66' list='' action=LOGIN service=LOGIN
00:09:40: AAA/AUTHEN/START (3251495553): using "default" list
00:09:40: AAA/AUTHEN/START (3251495553): Method=radius (radius)
00:09:40: AAA/AUTHEN (3251495553): status = GETUSER
```

```
00:09:42: AAA/AUTHEN/CONT (3251495553): continue_login (user='(undef)')
00:09:42: AAA/AUTHEN (3251495553): status = GETUSER
00:09:42: AAA/AUTHEN (3251495553): Method=radius (radius)
00:09:42: AAA/AUTHEN (3251495553): status = GETPASS
00:09:44: AAA/AUTHEN/CONT (3251495553): continue_login (user='metlosh')
00:09:44: AAA/AUTHEN (3251495553): status = GETPASS
00:09:44: AAA/AUTHEN (3251495553): Method=radius (radius)
00:09:44: RADIUS: ustruct sharecount=1
00:09:44: Radius: radius_port_info() success=1 radius_nas_port=1
00:09:44: RADIUS: Initial Transmit tty66 id 0 192.168.2.2:1812, Access-Request,len 78
00:09:44:        Attribute 4 6 C0A80201
00:09:44:        Attribute 5 6 00000042
00:09:44:        Attribute 61 6 00000005
00:09:44:        Attribute 1 9 6D65746C
00:09:44:        Attribute 31 13 3139322E
00:09:44:        Attribute 2 18 06C45434
00:09:44: RADIUS: Received from id 0 192.168.2.2:1812, Access-Accept, len 26
00:09:44:        Attribute 6 6 00000006
00:09:44: RADIUS: saved authorization data for user 826B9C9C at 8268EEF8
00:09:44: AAA/AUTHEN (3251495553): status = PASS
00:09:44: tty66 AAA/AUTHOR/EXEC (3584428966): Port='tty66' list='' service=EXEC
00:09:44: AAA/AUTHOR/EXEC: tty66 (3584428966) user='metlosh'
00:09:44: tty66 AAA/AUTHOR/EXEC (3584428966): send AV service=shell
00:09:44: tty66 AAA/AUTHOR/EXEC (3584428966): send AV cmd*
00:09:44: tty66 AAA/AUTHOR/EXEC (3584428966): found list "default"
00:09:44: tty66 AAA/AUTHOR/EXEC (3584428966): Method=radius (radius)
00:09:44: AAA/AUTHOR (3584428966): Post authorization status = PASS_ADD
00:09:44: AAA/AUTHOR/EXEC: Processing AV service=shell
00:09:44: AAA/AUTHOR/EXEC: Processing AV cmd*
00:09:44: AAA/AUTHOR/EXEC: Processing AV priv-lvl=15
00:09:44: AAA/AUTHOR/EXEC: Authorization successful
```

Z tohoto výpisu lze poznat, jak se do proměnných GETUSER a GETPASS ukládají zadané hodnoty. Dále že přišla na Radius server žádost o přijetí požadavku. Radius tento požadavek přijal a bylo to z IP adresy 192.168.2.2:1812. Poslední řádek vypisuje, že autorizace byla úspěšná.

Výpis hlášení FreeRadius serveru (úspěšný pokus):

```
rad_recv: Access-Request packet from host 192.168.2.1:1645, id=0, length=78
        NAS-IP-Address = 192.168.2.1
        NAS-Port = 66
        NAS-Port-Type = Virtual
        User-Name = "metlosh"
        Calling-Station-Id = "192.168.1.2"
        User-Password = "metlosh"
modcall: entering group authorize for request 2
  modcall[authorize]: module "preprocess" returns ok for request 2
  modcall[authorize]: module "chap" returns noop for request 2
  modcall[authorize]: module "attr_filter" returns noop for request 2
rlm_eap: EAP-Message not found
  modcall[authorize]: module "eap" returns noop for request 2
    rlm_realm: No '/' in User-Name = "metlosh", looking up realm NULL
    rlm_realm: No such realm "NULL"
  modcall[authorize]: module "realmslash" returns noop for request 2
    rlm_realm: No '@' in User-Name = "metlosh", looking up realm NULL
    rlm_realm: No such realm "NULL"
  modcall[authorize]: module "suffix" returns noop for request 2
    users: Matched DEFAULT at 152
```

```
    users: Matched metlosh at 220
  modcall[authorize]: module "files" returns ok for request 2
  modcall[authorize]: module "mschap" returns noop for request 2
modcall: group authorize returns ok for request 2
  rad_check_password:  Found Auth-Type Local
auth: type Local
auth: user supplied User-Password matches local User-Password
Login OK: [metlosh/metlosh] (from client ethernet port 66 cli 192.168.1.2)
Sending Access-Accept of id 0 to 192.168.2.1:1645
        Service-Type = Administrative-User
Finished request 2
Going to the next request
--- Walking the entire request list ---
Waking up in 6 seconds...
--- Walking the entire request list ---
Cleaning up request 2 ID 0 with timestamp 40bf358c
Nothing to do.  Sleeping until we see a request.
```

Na Radius server přišel paket s požadavkem (Access-Request), dále je vypsáno od koho, z jakého IP (klienta-NAS), IP odkud se user snaží připojit, jeho jméno, heslo. Dále se zjišťuje typ autorizace (tj. kde bude hledat uživatele: jestli v *users* nebo v systému – *passwd*), zkontroluje uživatele a jeho heslo. Server pošle paket s informací o úspěšné autorizaci (Access-Accept) a čeká na další žádosti.

Výpis hlášení WinRadius serveru (úspěšný pokus):

```
32 2004y6m3d 17h31m34s Message Type=Access_Request
33 2004y6m3d 17h31m34s ID=5, Length=78
34 2004y6m3d 17h31m34s NAS IP address=3232235777
35 2004y6m3d 17h31m34s NAS port=66
36 2004y6m3d 17h31m34s NAS port type=5
37 2004y6m3d 17h31m34s User name=metlosh
38 2004y6m3d 17h31m34s Calling number=192.168.1.2
39 2004y6m3d 17h31m34s Password ciphed text=§=)&+ÂţpÐ8X¦á
40 2004y6m3d 17h31m34s Password deciphed text=metlosh
41 2004y6m3d 17h31m34s User (metlosh) authenticate OK.
```

Výpis je dosti podobný výpisu FreeRadius serveru, je z něj vcelku jasné, co na serveru probíhá. Zajímavý je údaj s zašifrovaným heslem.

Výpis z debug hlášení směrovače (po nastavení debug aaa, radius, packet, neúspěšný pokus – při nastavení : Auth-Type := System na FreeRadius serveru):

```
00:16:50: RADIUS: Initial Transmit tty67 id 1 192.168.2.2:1812, Access-Request,len 78
00:16:50:        Attribute 4 6 C0A80201
00:16:50:        Attribute 5 6 00000043
00:16:50:        Attribute 61 6 00000005
00:16:50:        Attribute 1 9 6D65746C
00:16:50:        Attribute 31 13 3139322E
00:16:50:        Attribute 2 18 D6F903E5
00:16:52: RADIUS: Received from id 1 192.168.2.2:1812, Access-Reject, len 20
00:16:52: RADIUS: saved authorization data for user 826B9978 at 0
00:16:52: AAA/AUTHEN (3356768723): status = FAIL
00:16:52: AAA/AUTHEN/ABORT: (3356768723) because Unknown.
00:16:52: AAA/MEMORY: free_user_quiet (0x826B9978) user='metlosh' ruser='NULL' p
          ort='tty67' rem_addr='192.168.1.2' authen_type=1 service=1 priv=1
00:16:52: AAA: parse name=tty67 idb type=-1 tty=-1
```

```
00:16:52: AAA: name=tty67 flags=0x11 type=5 shelf=0 slot=0 adapter=0 port=67 channel=0
00:16:52: AAA/MEMORY: create_user (0x826B9978) user='NULL' ruser='NULL' ds0=0 port='tty67'
          rem_addr='192.168.1.2' authen_type=ASCII service=LOGIN priv=1 initial_task_id='0'
```

Začátek se nijak neliší až po přijetí paketu s požadavkem. Z dalšího výpisu je vidět, že přišel paket se zamítnutou žádostí o přístup (authorizaci) z neznámého důvodu.

Výpis hlášení FreeRadius serveru (neúspěšný pokus, Auth-Type := System):

```
  rad_check_password:  Found Auth-Type System
auth: type "System"
modcall: entering group authenticate for request 0
  modcall[authenticate]: module "unix" returns notfound for request 0
modcall: group authenticate returns notfound for request 0
auth: Failed to validate the user.
Login incorrect: [metlosh/metlosh] (from client ethernet port 67 cli 192.168.1.2)
Delaying request 0 for 1 seconds
Finished request 0
Going to the next request
--- Walking the entire request list ---
Waking up in 1 seconds...
--- Walking the entire request list ---
Waking up in 1 seconds...
--- Walking the entire request list ---
Sending Access-Reject of id 1 to 192.168.2.1:1645
Waking up in 4 seconds...
--- Walking the entire request list ---
Cleaning up request 0 ID 1 with timestamp 40bf3737
Nothing to do.  Sleeping until we see a request.
```

Výpis je rozdílný až od místa zjišťování Auth-Type. Autorizace se nepodařila, protože uživatel nebyl nalezen. Server proto pošle paket se zamítnutím žádosti a čeká na další požadavky.

Výpis hlášení WinRadius serveru (neúspěšný pokus, zadáno špatné heslo):

```
2 2004y6m3d 17h33m59s Message Type=Access_Request
3 2004y6m3d 17h33m59s ID=6, Length=78
4 2004y6m3d 17h33m59s NAS IP address=3232235777
5 2004y6m3d 17h33m59s NAS port=66
6 2004y6m3d 17h33m59s NAS port type=5
7 2004y6m3d 17h33m59s User name=metlosh
8 2004y6m3d 17h33m59s Calling number=192.168.1.2
9 2004y6m3d 17h33m59s Password ciphed text=-ł¦ÓbŐ-¦ÖAÜÓ¦hĐÔ
10 2004y6m3d 17h33m59s Password deciphed text=kokot
11 2004y6m3d 17h33m59s Reason: Wrong password or secret
12 2004y6m3d 17h33m59s User (metlosh) authenticate failed.
```

V tomto případě je výpis směrovače stejný jako v předešlém případě (při nastavení: Auth-Type := System na FreeRadius serveru), tj. přišel paket se zamítnutou žádostí o přístup (autorizaci) z neznámého důvodu.

Varianta komunikace s RADIUS serverem za použití SSH

8. Popis SSH

SSH (Secure Shell) je protokol sloužící – podobně jako Telnet – k přístupu ke vzdálenému počítači, důležitý rozdíl však spočívá v tom, že veškerá komunikace probíhající pomocí protokolu SSH je šifrovaná, a tedy podstatně bezpečnější než prostřednictvím protokolu Telnet. Bývá často udáváno, že protokol SSH je "bezpečnější náhradou" protokolů jako telnet, ftp, rsh, rcp, rlogin a dalších.

FreeRadius

Nastavení zůstává stejné, jen v souboru **users** přibude jeden řádek:

Service-Type = Shell-User

Tento atribut určute typ služby, o kterou uživatel žádá, či typ služby, která by měla být poskytnuta. Bez tohoto nastavení se objeví chybové hlášení: ***no appropriate authorization type for user.*** Přesný popis lze najít v RFC 2058 (http://www.faqs.org/rfcs/rfc2058.html).

Cisco směrovač

Nastavení je opět stejné, až na několik příkazů souvisejících s SSH:

```
Router(config)# hostname hostname
Router(config)# ip domain-name domainname
Router(config)# crypto key generate rsa
Router(config)# ip ssh {[timeout seconds]} | [authentication-
retries integer]}
Router# debug ip ssh
Router# debug ip ssh client
```

hostname *hostname*	nastavení jména směrovače
ip domain-name *domainname*	nastavení jména domény
crypto key generate rsa	aktivace SSH serveru pro lokální a vzdálenou autentizaci na směrovači
ip ssh	nastavení proměných SSH – timeout (1–120 sekund), authentication-retries – počet pokusů o autentizaci (1–5)
Router# ***debug ip ssh***	nastavení pro výpis probýhajících činností (týkajících se SSH)
Router# ***debug ip ssh client***	

Pro kontrolu, jestli je SSH server funkční slouží příkaz:

Router# ***show ip ssh***

Výpis konfigurace směrovače

```
Current configuration : 846 bytes
!
version 12.2
service timestamps debug uptime
```

```
service timestamps log uptime
no service password-encryption
!
hostname pokus
!
aaa new-model
aaa authentication login default group radius
aaa authorization exec default group radius
enable password cisco
!
ip subnet-zero
!
!
ip domain-name SSH
!
ip ssh time-out 5
ip ssh authentication-retries 2
!
call rsvp-sync
!
!
!
!
!
!
!
!
interface Ethernet0/0
 ip address 192.168.2.1 255.255.255.0
 half-duplex
!
interface Serial0/0
 no ip address
 shutdown
!
interface Ethernet0/1
 ip address 192.168.1.1 255.255.255.0
 half-duplex
!
interface Serial0/1
 no ip address
 shutdown
!
no ip classless
ip http server
!
!
radius-server host 192.168.2.2 auth-port 1812 acct-port 1813
radius-server retransmit 1
radius-server key radius
!
```

```
dial-peer cor custom
!
!
!
!
!
line con 0
line aux 0
line vty 0 4
!
end
```

Debug

Výpis z debug hlášení směrovače (po nastavení debug aaa, radius, packet, ip ssh, ip ssh client úspěšný pokus):

SSH0: starting SSH control process
03:03:10: SSH0: sent protocol version id SSH-1.5-Cisco-1.25
03:03:10: SSH0: protocol version id is – SSH-1.5-PuTTY-Release-0.54
03:03:10: SSH0: SSH_SMSG_PUBLIC_KEY msg
03:03:11: SSH0: SSH_CMSG_SESSION_KEY msg – length 112, type 0x03
03:03:11: SSH: RSA decrypt started
03:03:12: SSH: RSA decrypt finished
03:03:12: SSH: RSA decrypt started
03:03:12: SSH: RSA decrypt finished
03:03:12: SSH0: sending encryption confirmation
03:03:12: SSH0: keys exchanged and encryption on
03:03:15: SSH0: SSH_CMSG_USER message received
03:03:15: SSH0: authentication request for userid metlosh
03:03:15: AAA: parse name=tty66 idb type=-1 tty=-1
03:03:15: AAA: name=tty66 flags=0x11 type=5 shelf=0 slot=0 adapter=0 port=66 channel=0
03:03:15: AAA/MEMORY: create_user (0x823D7098) user='NULL' ruser='NULL' ds0=0 port='tty66'
rem_addr='192.168.1.2' authen_type=ASCII service=LOGIN priv=15 initial_task_id='0'
03:03:15: AAA/AUTHEN/START (1169735414): port='tty66' list='' action=LOGIN service=LOGIN
03:03:15: AAA/AUTHEN/START (1169735414): using "default" list
03:03:15: AAA/AUTHEN/START (1169735414): Method=radius (radius)
03:03:15: AAA/AUTHEN (1169735414): status = GETPASS
03:03:15: SSH0: SSH_SMSG_FAILURE message sent
03:03:17: SSH0: SSH_CMSG_AUTH_PASSWORD message received
03:03:17: AAA/AUTHEN/CONT (1169735414): continue_login (user='metlosh')
03:03:17: AAA/AUTHEN (1169735414): status = GETPASS
03:03:17: AAA/AUTHEN (1169735414): Method=radius (radius)RADIUS: ustruct sharecount=1
03:03:17: Radius: radius_port_info() success=1 radius_nas_port=1
03:03:17: RADIUS: Initial Transmit tty66 id 15 192.168.2.2:1812, Access-Request, len 78
03:03:17: Attribute 4 6 C0A80101
03:03:17: Attribute 5 6 00000042
03:03:17: Attribute 61 6 00000005
03:03:17: Attribute 1 9 6D65746C
03:03:17: Attribute 31 13 3139322E
03:03:17: Attribute 2 18 D39259E2
03:03:17: RADIUS: Received from id 15 192.168.2.2:1812, Access-Accept, len 26
03:03:17: Attribute 6 6 00000006
03:03:17: RADIUS: saved authorization data for user 823D7098 at 82691738
03:03:17: AAA/AUTHEN (1169735414): status = PASS
03:03:17: SSH0: authentication successful for metlosh
03:03:17: SSH0: requesting TTY

03:03:17: SSH0: setting TTY – requested: length 24, width 80; set: length 24, width 80
03:03:17: SSH0: SSH_CMSG_EXEC_SHELL message received
03:03:17: tty66 AAA/AUTHOR/EXEC (480959312): Port='tty66' list='' service=EXEC
03:03:17: AAA/AUTHOR/EXEC: tty66 (480959312) user='metlosh'
03:03:17: tty66 AAA/AUTHOR/EXEC (480959312): send AV service=shell
03:03:17: tty66 AAA/AUTHOR/EXEC (480959312): send AV cmd*
03:03:17: tty66 AAA/AUTHOR/EXEC (480959312): found list "default"
03:03:17: tty66 AAA/AUTHOR/EXEC (480959312): Method=radius (radius)
03:03:17: AAA/AUTHOR (480959312): Post authorization status = PASS_ADD
03:03:17: AAA/AUTHOR/EXEC: Processing AV service=shell
03:03:17: AAA/AUTHOR/EXEC: Processing AV cmd*
03:03:17: AAA/AUTHOR/EXEC: Processing AV priv-lvl=15
03:03:17: AAA/AUTHOR/EXEC: Authorization successful

Z výpisu je zřejmé, že se nejdříve posílají údaje o verzích protokolu. Dále proběhne výměna klíčů, dekódování, pošle se zpráva o zašifrování – **keys exchanged and encryption on.** Zbylá část probíhá v podstatě stejně jako v předchozím případě (bez SSH).

Výpis hlášení směrovače (po nastavení debug aaa, radius, packet, ip ssh, ip ssh client neúspěšný pokus – špatné heslo):

03:11:11: RADIUS: Received from id 17 192.168.2.2:1812, Access-Reject, len 20
03:11:11: RADIUS: saved authorization data for user 823D7098 at 0
03:11:11: AAA/AUTHEN (2561472781): status = FAIL
03:11:11: SSH0: password authentication failed for metlosh
03:11:11: AAA/AUTHEN/START (4126144607): port='tty66' list='' action=LOGIN service=LOGIN
03:11:11: AAA/AUTHEN/START (4126144607): using "default" list
03:11:11: AAA/AUTHEN/START (4126144607): Method=radius (radius)
03:11:11: AAA/AUTHEN (4126144607): status = GETPASS
03:11:11: SSH0: SSH_SMSG_FAILURE message sent
03:11:16: SSH0: authentication failed for metlosh (code=18)
03:11:16: AAA/MEMORY: free_user (0x823D7098) user='metlosh' ruser='NULL' port='tty66'
rem_addr='192.168.1.2' authen_type=ASCII service=LOGIN priv=15
03:11:16: SSH0: Session disconnected – error 0x12

Opět zde není celý vypis, ale až od části, kde se texty liší. Rozdíl od normální autentizace a autentizace s SSH je v tom, že s SSH je přesně vypsáno, proč se autentizace nezdařila. V předešlých případech byl důvod neznámý, nyní je řečeno: **password authentication failed for metlosh** tj. nesouhlasí heslo, proto se autentizace nepodařila. Spojení (sezení) bylo přerušeno.

9. Problematika IOSu firewallů, IDS/IPS a traffic shapingu (QoS)

Základem bezpečnosti je filtrační pravidlo zvané Access Control List, ACL. V současnosti existují tyto typy ACL:

- základní – vyhodnocuje se jen IP adresa odesilatele,
- rozšířené – vyhodnocuje se protokol, obě adresy a oba porty, některé bity z IP záhlaví, DSCP, priorita apod.,
- dynamické (Lock and Key) – vše blokováno, uživatel se musí telnetem přihlásit na směrovač a při úspěchu se aktivuje příslušné ACL, čímž se odblokuje daný provoz,
- pojmenované – zlepšený typ standardního nebo rozšířeného ACL, lze snadno editovat,
- reflexívní – podle odchozího provozu automaticky nastavuje pravidlo umožňující příslušný příchozí provoz; ten je jinak kompletně blokován,
- časově omezené – od-do apod.,
- Context-Based Access Control – fakticky stavový firewall, obdobně jako dynamický ACL sleduje odchozí aplikační vrstvu (formální správnost, ne obsah) a povoluje příchozí; detekuje některé známé útoky (SYN flood, podezřelá sekvenční čísla mimo aktuální okno, dokáže rušit „half-open“ spojení),
- autentizační proxy – uživatel musí být autentizován pomocí TACACS+ nebo RADIUS, pak se mu povolí komunikace,
- Turbo ACL – překompilované ACL, jen u nejvýkonnějších směrovačů, kde ACL mohou mít značný rozsah,
- distribuované časově omezené – užívané jen u výkonných směrovačů osazených linkovými kartami pro vysoké rychlosti (OC) a při použití VPN. Umožňují zpracování přímo kartou a ne procesorem,
- přijímací – – užívané jen u výkonných směrovačů osazených linkovými kartami pro vysoké rychlosti, umožňují chránit gigabitový směrovací procesor před nežádoucí zátěží,
- pro ochranu infrastruktury – minimalizují rizika a účinek přímého útoku proti zařízením infrastruktury; do těchto povolují pouze explicitně uvedený (oprávněný) provoz, rovněž povolují veškerý další tranzitní provoz,
- tranzitní – používají se ke zvýšení zabezpečení sítě, explicitně povolují pouze požadovaný provoz do sítě nebo sítí.

ACL však mají univerzálnější použití jako rozhodovací prvek všude tam, kde je třeba porovnat zapracovávaná data s daným vzorem a při shodě pak provést akci. Touto akcí může být kromě blokování či povolení provozu například překlad adresy (NAT), umožnění zpětného přístupu, aktivace pravidla atd.

Funkce Cisco IOS firewallu (dle Petra Grygárka)
Cisco IOS Firewall poskytuje integrované funkce firewallu a tím zvyšuje flexibilitu a bezpečnost Cisco routeru. Stručný přehled jeho nejdůležitějších vlastností:

- Kontextově závislé řízení přístupu CBAC – interním uživatelům nabízí bezpečné řízení přístupu podle jednotlivých aplikací, podléhá mu veškerý provoz sítě.
- Detekce průniku – Okamžité monitorování, zadržení a reakce na zneužití sítě, je postavena na množině signatur reprezentující nejběžnější typy útoků.

- Detekce a prevence průniku odepřením služeb – brání a ochraňuje prostředky směrovače a strojů v sítí proti běžným útokům; kontroluje hlavičky paketů a podezřelé pakety zahazuje.
- Blokování java appletů – brání síť proti zlomyslným java appletům.
- Okamžitá varovná hlášení – v reálném čase zaznamenává varování o útocích.
- Záznam auditu – podrobné sledování provozu; pro podrobné sestavy zaznamenává časové razítko, zdrojový hostitelský systém, cílový hostitelský systém, porty, dobu trvání a celkový počet přenesených bajtů.
- Záznam událostí – Pomocí záznamu událostí může síťový administrátor sledovat v reálném čase potenciální prolomení bezpečnosti a jiné nestandardní aktivity.
- Autentizace partnerských směrovačů – zajišťuje příjem paketů jen od důvěryhodných zdrojů.

CBAC

Podstatou je firewall, tvořený dvěma (interními) komponentami, jedna z nich plní funkci systému detekce průniku, druhá realizuje kontextově závislé řízení přístupu (Context-Based Access Control – CBAC). CBAC udržuje stavové tabulky pro všechny odchozí spojení procházející směrovačem (tj. je třeba definovat porty a směry) přičemž prověřuje TCP a UDP komunikaci, z ní vybírá data aplikační vrstvy a pak podle nich příslušným způsobem vyplňuje tabulky. Provoz přicházející na dané rozhraní pak porovnává s údaji v této tabulce a zjišťuje, zda se jedná o korektní spojení, tj. odpovědi na podněty pocházející z vnitřní sítě, a pak rozhoduje o povolení či zákazu.

Specifická podpora aplikací a CBAC

CBAC umožňuje směrovači rozpoznat a identifikovat datové toky, specifické pro typické aplikace, kterými jsou:

- CUSeeMe Protocol
- ftp
- h323
- http
- rcmd
- realaudio
- rpc
- smyl
- sqlnet
- streamworks
- tcp
- FTP
- Udp
- Vdolive

CBAC umožňuje identifikovat poškozené pakety nebo podezřelé aplikační datové toky a tyto zakázat (event. povolit). CBAC také dovoluje stahovat Java kódy z důvěryhodných severů a blokovat z nedůvěryhodných.

CBAC a Denial of Service (DoS)

Ochrana proti útokům typu Denial-of-Service (DoS) aktivuje protiopatření ke zmírnění hrozeb a samozřejmě zaznamenává údaje o časech, příslušné výstrahy. Lze například čelit útokům typu TCP SYN Flood, fragmentace apod. a to např. ovlivněním parametrů pro navazování TCP spojení. K tomuto účelu využívá CBAC časové limity (timeouty) a prahové hodnoty, které jsou konfigurovatelné, dále pak určit, jak dlouho mají být uchovávány stavové informace informace o každém spojení dané relace a kdy je zahodit. Ohledně UDP a ICMP je třeba uvážit, že tyto protokoly jsou bezstavové, tudíž k rozhodnutí o tom, zda „spojení“, které jich používá, již není aktivní, vyžaduje časovač; po jeho vypršení budou příchozí pakety, které by měly k takovému spojení náležet, zahozeny. Velmi užitečný příkaz k identifikaci DoS útoků je "`ip inspect audit-trail`", který zaznamenává všechny potenciální DoS spojení včetně zdrojové a cílové IP adresy a TCP nebo UDP portů.

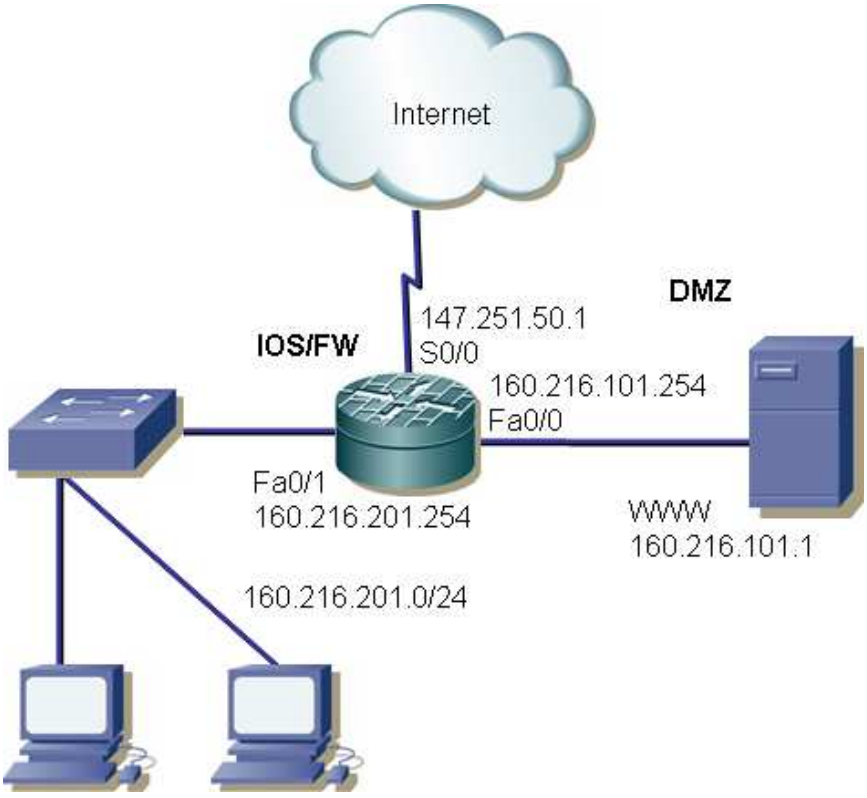
Konfigurace CBAC

Konfigurace CBAC na směrovačích Cisco se provádí v pěti krocích. Jsou to:

1. Volba rozhraní, na kterém bude kontrola aplikována. Toto rozhraní může být interní nebo externí (tj. připojené do vnitřní či vnější sítě). CBAC se zabývá pouze směrem prvního paketu, který iniciuje spojení.
2. Konfigurace ACL ve správném směru a na zvoleném rozhraní tak, aby byl povolen provoz a následná kontrola CBAC.
3. Konfigurace globálních časových limitů a prahové hodnoty pro navázaná spojení či relace.
4. Definice inspekčních pravidel určujících, přesně které protokoly budou zkontrolovány CBAC.
5. Aplikace inspekčního pravidla na rozhraní ve správném směru.

Příklad – varianta konfigurace CBAC

Hraniční směrovač se třemi rozhraními je připojen do Internetu prostřednictvím rozhraní Serial 0/0 s IP adresou 147.251.50.1, prostřednictvím rozhraní FastEthernet0/1 s IP adresou 160.216.201.254 je připojen do vnitřní sítě 160.216.201.0/24. Na rozhraní FastEthernet 0/0 je realizována demilitarizovaná zóna, kde sídlí webový server 160.216.101.1. Požaduje se umožnit přístup z Internetu na tento webový server. Interní adresy jsou z konspiračních důvodů ukryty prostřednictvím statického překladu adres a překladu adres portů (NAT, PAT) na rozhraní Serial 0/0 a to pro veškerou odchozí komunikaci z vnitřní sítě do Internetu. Funkčnost lze ověřit příkazem `show ip inspect sessions`.


Internet
DMZ
147.251.50.1
S0/0
IOS/FW
160.216.101.254
Fa0/0
Fa0/1
160.216.201.254
WWW
160.216.101.1
160.216.201.0/24

```
! Odchozi CBAC inspekcni pravidla
 ip inspect name CBAC-IN-OUT tcp
 ip inspect name CBAC-IN-OUT ftp
 ip inspect name CBAC-IN-OUT h323
 ip inspect name CBAC-IN-OUT rcmd
 ip inspect name CBAC-IN-OUT http
 ip inspect name CBAC-IN-OUT netshow
 ip inspect name CBAC-IN-OUT realaudio
 ip inspect name CBAC-IN-OUT rtsp
 ip inspect name CBAC-IN-OUT sqlnet
 ip inspect name CBAC-IN-OUT streamworks
 ip inspect name CBAC-IN-OUT tftp
 ip inspect name CBAC-IN-OUT udp
 ip inspect name CBAC-IN-OUT vdolive

! Prichozi CBAC inspekcni pravidla pro prichozi http provoz
 ip inspect name CBAC-OUT-IN http

! Rozhrani do DMZ
 interface FastEthernet0/0
 ip address 160.216.101.254 255.255.255.0
 ip nat inside
 full-duplex
 no cdp enable
 !
! Rozhrani do vnitrni LAN
 interface FastEthernet0/1
 ip address 160.216.201.254 255.255.255.0
 ip nat inside
 full-duplex
 no cdp enable
!

! Vnejsi rozhrani do Internetu
! Poznamka – prichozi ACL a CBAC pravidla jsou aplikovana jak
! pro prichozi, tak i odchozi kontrolu
 interface Serial0/0
 description CONNECTED TO INTERNET
 bandwidth 1024
 ip address 147.251.50.1 255.255.255.252
 ip access-group FIREWALL in
 ip nat outside
 ip inspect CBAC-OUT-IN in
 ip inspect CBAC-IN-OUT out

ip nat inside source list 122 interface Serial0/0 overload
 ip nat inside source static tcp 160.216.101.1 80 147.251.50.1 80
extendable no-alias
 ip classless
 ip route 0.0.0.0 0.0.0.0 147.251.50.2

! Toto ACL bude pouzito odchozim CBAC pravidlem pro otevreni docasne
! diry pro zpětný provoz
 ip access-list extended FIREWALL
 permit icmp any any echo-reply
 permit tcp any host 147.251.50.1 eq 80
 deny   ip any any log

access-list 122 permit ip 160.216.201.0 0.0.0.255 any
```

Definice politiky aplikačního firewallu

Jedná se o zajímavou možnost, která usnadňuje tvorbu politiky pro danou aplikaci. K tomuto účelu slouží příkaz `appfw`. Jeho použití je velmi rozsáhlé. Dále jsou uvedeny dva ilustrační příklady, první se zabývá protokolem HTTP, druhý instantním messagingem.

```
! Definice aplikační politiky firewally pro HTTP
appfw policy-name mypolicy
 application http
  audit trail on
  strict-http action allow alarm
  content-length maximum 1 action allow alarm
  content-type-verification match-req-rsp action allow alarm
  max-header-length request 1 response 1 action allow alarm
  max-uri-length 1 action allow alarm
  port-misuse default action allow alarm
  request-method rfc default action allow alarm
  request-method extension default action allow alarm
  transfer-encoding type default action allow alarm
!
!
! Aplikace politiky na inspekční pravidlo.
ip inspect name firewall appfw mypolicy
ip inspect name firewall http
!
!
! Aplikace inspekčního pravidla na veškerý HTTP provoz vstupující
! pres rozhrani FastEthernet0/0
interface FastEthernet0/0
 ip inspect firewall in
!
!
```

Následující příklad ukazuje aplikační politiku "`my-im-policy`", která povolí textový chat pro uživatele instantního messengeru Yahoo! a blokuje provoz dalších instantních messengerů.

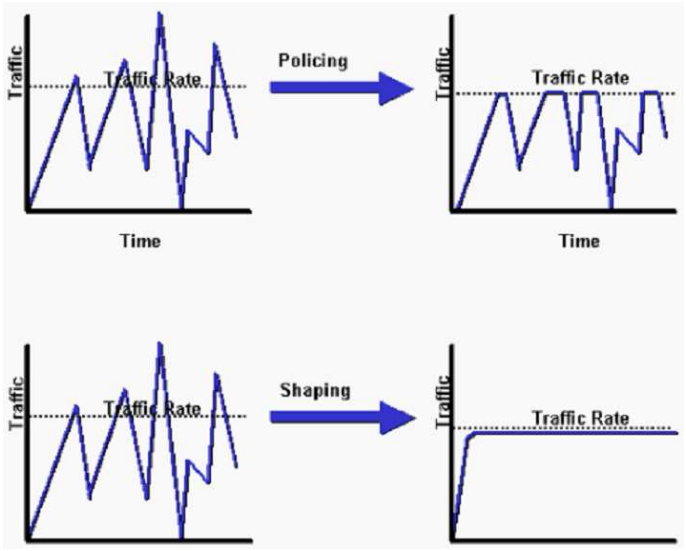
```
! Definice aplikacni politiky firewallu pro instantni messaging
! Yahoo povolit, ostatní zakazat
appfw policy-name my-im-policy
 application http
  port-misuse im reset
!
 application im yahoo
  server permit name scs.msg.yahoo.com
  server permit name scsa.msg.yahoo.com
  server permit name scsb.msg.yahoo.com
  server permit name scsc.msg.yahoo.com
  service text-chat action allow
  service default action reset
!
 application im aol
  server deny name login.cat.aol.com
!
 application im msn
  server deny name messenger.hotmail.com
!
ip inspect name test appfw my-im-policy

interface FastEthernet0/0
 description Inside interface
 ip inspect test in
```

Regulace přenosové rychlosti jako prvek aktivní ochrany
Cisco používá dvě základní metody pro omezení přenosové rychlosti

- policing (data přesahující sjednanou rychlost jsou okamžitě zahazována – i při chvilkové zátěži),
  - Committed Access Rate (CAR),
  - Class-Based Policing,
- shaping (data přesahující sjednanou rychlost jsou ukládána do bufferů a odbavena později, lze-li).
  - Class-based Shapping a Distributed Traffic Shaping
  - Generic Traffic Shaping
  - Frame Relay Traffic Shaping

Cisco doporučuje používat Class-Based Policing nebo Class-based Shapping/Distributed Traffic Shaping.



Porovnání policingu a shapingu

Příklad konfigurace policingu
Dohodnutá rychlost činí 8000 b/s, burst (shluk – tj. velikost bufferu) 1000 B

```
Router(config)# policy-map police-setting
Router(config-pmap)# class access-match
Router(config-pmap-c)# police 8000 1000 conform-action transmit exceed-
action drop
```

Příklad konfigurace shapingu

```
policy-map parent
    class class-default
     shape average 3300000 103000 0
      service-policy child
```

Argument `3300000` udává smluvenou rychlost, argument `103000` pak velikost burstu (shluku – tj. velikost bufferu).

10. Kvalita služby u zařízení Cisco

10.1 Pojem QoS

Poprve byla zkratka QoS (Quality of service) použita v RFC 1946 z května 1996 s názvem „Native ATM Support for ST2+" [RFC1946]. Když nahlédneme do jednotlivých norem, najdeme celou škálu výkladů tohoto pojmu, např. ITU-T Recommendation E.800 [ITU-TE.800] QoS definuje jako „stupeň splnění požadavků pomocí sady vnitřních charakteristik". Řada dalších definic již požadavky třídí anebo navazuje na poskytování služeb v rámci SLA – Service Level Agreement); kde SLA je chápáno jako sada parametrů a jejich hodnot, které definují síťové služby.

Tato učební pomůcka se týká zařízení Cisco a u této společnosti je poskytování služeb zarámováno speciální architekturou zvanou Cisco Next-Generation Network Framework (NGN) založené je na třech konvergencích:

- Konvergence aplikací – data, hlas a video pro uživatele s PC, notebookem, PDA či tabletem;
- Konvergence služeb – poskytovaných přes kabeláž, WLAN, celulární síť;
- Konvergence sítí – eliminující specifické sítě, např. ATM. Frame Relay, SAN, PSTN, optické, mobilní, vysokorychlostní, širokopásmové atd.

Sítě NGN musí splňovat specifické požadavky, a to dostupnost služeb i v chybové prostředí (zajištěno redundancí síťových prvků), předvídatelnost výkonnosti sítě a QoS. Stabilita sítí NGN vyžaduje zajištění bezpečnosti jednotlivých prvků sítě, ochranu jádra sítě (jejích IP adres, ochrana před jejím zahlcením), ochranu vrstvy řízení sítě a ochranu síťových služeb.

Spokojenost uživatele s poskytovanými službami nezávisí pouze na síťových parametrech, nýbrž i na širší škále parametrů (kompresní schémata atd.) typických pro hlas a video, které spadají do šířeji chápaných služeb označovaných někdy jako QoE (Quality of Experience). Pokud měříme QoS, zjišťujeme zpoždění paketů, jeho rozptyl (jitter) a ztráty paketů, zatímco při měření QoE zjišťujeme i další parametry: echo, šum, hlasitost a MOS (Mean Opinion Scores). S používáním termínu QoE se setkáme u Psytechnics, což je společnost zaměřená na diagnostiku kvality hlasu a videa, u Centillium Communications, společnossti dodávající produkty pro tzv. "last mile" typu FTTP (Fiber-To-The-Premises) a VoIP. Setkáme se sle i s dalšími šířejí pojatými termíny – Multilayered QoS (Extreme Networks), „reliable QoS" (Cisco) atd. Neboli i u Cisca je nyní pojem QoS chápán v širším slova smyslu.

Efekt použití mechanismu QoS je dobře viditelný např. u bezdrátové sítě připojené přes Access Point ke kabelové síti Ethernet. QoS pro bezdrátové sítě LAN se zaměřuje na stanovení priority typu downstream, tj. toku z Access Pointu směrem ke koncovým stanicím.

Je hezké, že koncepce QoS řeší problémy smíšeného provozu, kdy má hlas přednost před videem a to před datovými přenosy. Ale co se stane, pokud síť zaplavuje více toků videa, jak je mezi sebou rozlišovat a upřednostňovat? Tuto situaci použití mechanismu QoS neřeší.

Na závěr těchto úvah dobrá zpráva [Ethernet2010] z hlediska orientace se na produkty společnosti Cisco. Jeho přepínače podle výsledků provedených testů ve srovnání s obdobnými produkty společností Dell, Blade Network Technologies, HP a Nortel zajišťují výrazně vyšší kvalitu QoS – při stejně vysoké zátěži, kdy výpadky u konkurence se pohybovaly v rozmezí 25–90 % provozu, přepínače Cisco ztrácely jediný paket. Tento výsledek je dán použitím architektury ASIC (Application-specific Integrated Circuit) v přepínačích Cisco.

10.2 Parametry QoS a typy služeb

QoS je dána čtyřmi parametry: zpoždění, jeho rozptyl (jitter), ztráty paketů a odezva (echo).

a) Zpoždění

Zpoždění rozeznáváme

- procesní – od příchodu paketu na vstupní rozhraní po jeho výdej cestou výstupního rozhraní;
- frontové – při čekání na zpracování;
- serializační – přenosu po médiu;
- propagační – doba vlastního přenosu paketu.

Zpoždění závisí na vzdálenosti, kodeku (čas pro digitalizaci a kompresi hlasového vzorku), serializačním zpoždění (čas pro vyslání paketu na linku), ukládání paketů do vyrovnávací paměti a řazení do front. Z toho všeho lze mechanismy QoS ovlivnit pouze ukládání paketů do vyrovnávací paměti a řazení do front. Tento parametr je typu „end-to-end“. Hodnoty serializačního zpoždění pro rámce Frame relay ukazuje tabulka 10.1.

Tabulka 10.1: Serializační zpoždění pro rámce Frame relay různé velikosti.
Serializační zpoždění = velikost rámce (bit)/šířka pásma linky (b/s)

	1 byte	64 byte	128 byte	256 byte	512 byte	1024 byte	1500 byte
56 kb/s	143 µs	9 ms	18 ms	36 ms	72 ms	144 ms	214 ms
64 kb/s	125µ	8 ms	16 ms	32 ms	64 ms	126 ms	187 ms
128 kb/s	62,5 µs	4 ms	8 ms	16 ms	32 ms	64 ms	93 ms
256 kb/s	31 µs	2 ms	4 ms	8 ms	16 ms	32 ms	46 ms
512 kb/s	15,5 µs	1 ms	2 ms	4 ms	8 ms	16 ms	32 ms
768 kb/s	10 µs	640 µs	1,28 ms	2,56 ms	5,12 ms	10,24 ms	15 ms
1536 kb/s	5 µs	320 µs	640 µs	1,28 ms	2.56 ms	5,12 ms	7,5 ms

Je si třeba uvědomit, že ne vždy lze velikost serializačního zpoždění ovlivnit. Například pro satelitní spoje je charakteristické sterilizační zpoždění 250 až 900 ms a moc s tím nenaděláme. Vzdálenost mezi hladinou moře a geostatickou družicí je na úrovni rovníku 35 786 km. Něco málo je možné např. a cenu prudce rostoucích nákladů získat zvýšením rychlosti vysílaných radiových vln, ale to je asi tak vše.

b) Rozptyl zpoždění

Jitter závisí na době kódování či dekódování v rámci konkrétního kodeku a na rozptylu dob čekání ve frontách. Typickou situací je řada proudů paketů z jednoho zdroje. Tento parametr je třeba zjišťovat pro každou linku zvlášť.

c) Ztráty paketů v případě zahlcení

Pakety se při zahlcení se běžně vyhazují náhodným způsobem. Moderní algoritmy upřednostňují vyhazování v závislosti na délce fronty. V každém případě ztráty paketů způsobují trhanou řeč či obraz. QoS je třeba nastavovat pro každou linku zvlášť.

d) Odezva (echo)

Odezvu způsobují nehomogenity prostředí, jímž signál prochází. Jsou dva typy odezvy: elektrická odezva (konvertory analogového na digitální signál, koncovky, přechody dvou na čtyři páry) a akustická odezva (např. špatná headset). Jisté míry odezvy se v rámci zachování kvality volání nevyhneme, problémem se odezva stává, pokud zpoždění přesáhne 20 ms. K potlačení odezvy slouží příslušné zařízení (echo canceller), skládající se z adaptivního filtru, nelineárního procesoru a detektoru tónů.

Rozeznávají se tři typy služeb:

a) Bez záruky (Best effort)

Zajišťuje základní konektivitu a nic negarantuje; tak funguje klasický Internet.

b) Diferencované služby

Provoz je roztříděn do tříd podle požadavků na jednotlivé služby s cílem škálovat tok paketů. Třídy jsou rozlišovány a pakety jednotlivých tříd jsou obsluhovány specifickým způsobem. V tomto případě nejsme schopni rozlišovat jednotlivé toky paketů, řídíme agregované toky dat. Aplikace nejsou nijak modifikované a nepoužívají žádnou signalizaci mezi uzly. Platí, že není garantována žádná kvalita služby, jde jen o jistý mechanismus, jak zvýšit kvalitu poskytovaných služeb, tzv. soft QoS.

c) Integrované (garantované) služby

Služba si předem vyžádá rezervaci toků tak, aby síť vyhověla specifickým požadavkům toku, např. aby zajišťovala dostupnou šířku pásma. V tomto případě jsme schopni rozlišovat jednotlivé toky paketů. Vytváří jisté garance a označuje se za hard QoS. Tato garance je ale nepřímá – není k dispozici potřebné pásmo, ale vyrovnávací paměti v době zjišťování předpokládané pro zajištění tohoto pásma, což jsou rovnou dvě „snad“.

O službě typu „best efford“ stačí jen uvést, že je výhodná tehdy, když je trvale k dispozici dostatek pásma a tudíž není třeba zavádět nějaké zvláštní mechanismy. Pokud tomu tak není, nejsou různé typy služeb od sebe rozlišovány. Dále se tímto typem služby nebudeme zabývat.

Otázkou je, jaké pásmo vlastně potřebujeme pro jednotlivé služby:

Telefonní hovor: Kodeky Cisco DSP jsou schopny ztrátu jednoho paketu kompenzovat predikčním algoritmem. Ztráta dvou paketů již ale způsobí znatelnou pauzu v řeči. Za meze pro přijatelné parametry jsou považovány: zpoždění 150 ms, rozptyl zpoždění 30 ms a ztráta 1 % paketů. V závislosti na kodeku je zapotřebí pásmo 17–106 kb/s, pokud zahrneme režii 2. vrstvy, pak je vhodné počítat až se 150 kb/s.

Videokonference: ((infra) rámce obsahují plné vzorky I, tzn., že je třeba očekávat cca 1/3 rámců o délce 1024–1518 a více než polovinu rámců nad 512 bytů. Vzorky P (predictive) B (Bi-dir) vedou k rámcům o velikosti 128 až 256 bytů a zaberou cca 1/5 kapacity kanálu. Neboli klíčovým problémem je vypořádat se s rámci přenášejícími vzorky I a neřešit průměrné hodnoty. Pokud by nebyly v pořádku přeneseny vzorky I, z čeho by se počítaly vzorky P a B?

Interaktivní video: vyžaduje jako hlas zpoždění 150 ms, rozptyl zpoždění 30 ms a ztráta 1 % paketů, což dosáhne při 384 kb/s a pokud zahrneme záhlaví 2. vrstvy, musíme počítat se 460 kb/s.

Transakční systémy: liší se podle verze o řádové hodnoty, neboli je třeba vážit konkrétní verzi.

10.3 Model diferencovaných služeb

U tohoto modelu dále zkráceně označovaného jako DiffServ používáme tyto základní pojmy:

- BA – Behaviour Aggregate – množina paketů se stejnou DSCP hodnotou, tyto pakety jsou pak po lince posílány stejným směrem;
- PHB – Per-hop Behavior – což označuje individuální chování uzlu vůči jednotlivým agregovaným tokům (BA) paketů;
- Mechanismus PHB – specifický algoritmus nebo operace, která je implementovaná v uzlu.

Mechanismus PHB je realizován ve třech krocích:

Krok 1. Rozlišení provozu a stanovení požadavků na výkonnost sítě pro různé typy provozu
Příklad: zpoždění do 150 ms, jitter do 30 ms, ztrátovost do 1% (hodnoty charakteristické pro hlas)

Krok 2: Roztřídění do tříd provozu
Příklady: Nízké zpoždění, nízká priorita.

Krok 3: Zdokumentování politiky QoS.

Klíčové údaje pro QoS signalizaci IP sítí, obsahují dvě doporučení ITU, a to:

- Y.1541: kvantifikovat třídy QoS potřeb uživatelských aplikací v ve výkonnostních terminologii IP sítí – viz tabulka 1.2 převzatá z [ITU-TY.1541];
- Y.1221: „kontrakt o provozu“ doplňuje QoS třídy o popis vlastnosti toku a jeho limity. Kontrakt o provozu by měl podle této normy obsahovat kapacitní údaje (např. dedikované pásmo, statisticky používané pásmo) a popisné údaje (např. max. velikost paketu).

Význam zkratek v tabulce 10.2 je následující:

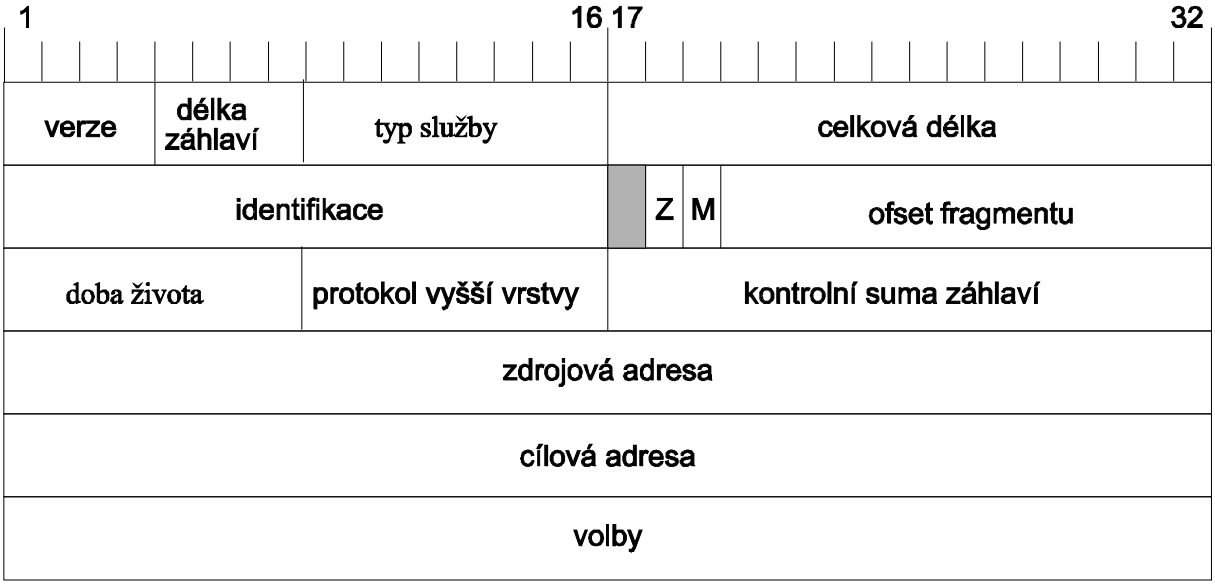
- IPTD – IP Packet Transfer Delay
- IPDV – IP Packet Delay Variation
- IPLR – IP Packet Loss Ratio
- IPER – IP Packet Error Ratio

Tabulka 10.2: Klasifikaci QoS dle doporučení ITU Y.1541.

Třída QoS	Charakteristika	IPTD	IPDV	IPLR	IPER
0	Přenos v reálném čase, citlivé na rozptyl zpoždění, vysoká interaktivita	100 ms	50 ms	$1x10^{-3}$	$1x10^{-4}$
1	Přenos v reálném čase, citlivé na rozptyl zpoždění, interaktivita	400 ms	50 ms	$1x10^{-3}$	$1x10^{-4}$
2	Transakční data, vysoká interaktivita	100 ms	bez limitu	$1x10^{-3}$	$1x10^{-4}$
3	Transakční data, interaktivita	400 ms	bez limitu	$1x10^{-3}$	$1x10^{-4}$
4	Citlivé na ztrátu paketů (krátké transakce, videořetězce, důležitá data)	1 s	bez limitu	$1x10^{-3}$	$1x10^{-4}$

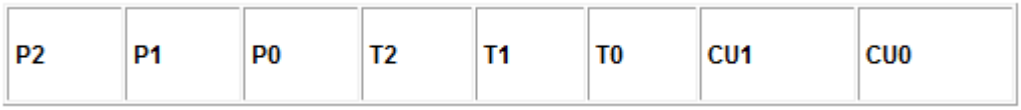
5	Ostatní aplikace v IP sítích se základním nastavením	bez limitu	bez limitu	bez limitu	bez limitu

DiffServ [RFC 2475] nezajišťuje přímou rezervaci pásma, ale zajišťuje dynamické rozlišení úrovně služeb požadované datovým tokem na základě informace v záhlaví paketu. Je proto podstatně vhodnější pro implementaci, neboť staticky nezabírá pásmo po dobu, kdy jej proces nevyužívá. K rozlišení úrovně služeb se používá pole ToS (Type of Service – typ služby) v záhlaví IP paketu (viz obr. 10.1).



Obr. 10.1: Záhlaví paketu IP.

Pole ToS bylo původně používáno v souladu s obr. 10.2, kde bity P2 až P0 označují IP precedenci, bity T2 až T0 cílovou funkci přenosu (zpoždění, průchodnost, spolehlivost) a dva bity CU (Currently Unused) označovaly rezervaci pro budoucí použití.



Obr. 10.2: Použití jednotlivých bitů pole ToS.

Hodnoty IP precedencí byly stanoveny takto (uvedeno raději z [RFC791] bez překladu):

111 – Network Control
110 – Internetwork Control
101 – critical
100 – Flash Override
011 – Flash
010 – Immediate
001 – Priority
000 – Routine

Praktické použití preference může být 5 (nejvyšší) pro hlas, 4 pro video a 3 pro signalizaci. Typ precedence lze stanovit v rámci konfigurace např. následujícím způsobem:

```
dial-peer voice 10 VoIP
IP precedence 5
```

Protože využití pole ToS neposkytovalo potřebné množství variant pro QoS bity T2 až T0 žádný výrobce nepoužíval, byl daný prostor v rámci modelu DiffServ přerozdělen (viz obr. 10.3):

- DSCP (Differentiated Services Code Point), kde bity 0 až 5 definují PHB index;
- ECN (Explicit Congestion Notification), což jsou bity 6 a 7 původně rezervované pro budoucí použití).

Obr. 10.3: Použití jednotlivých bitů pole ToS pro DSCP.

Na základě hodnoty indexu PHB rozhoduje směrovač v příslušné doméně, jak bude nakládáno s konkrétním paketem. Stejná hodnota indexu RHB může mít odlišný význam pro různé domény.

IETF definuje tři typy mechanismů PHB:

- EF (Expedited Forwarding), což je nejvyšší typ služby, který zajistí služby virtuální pronajaté linky;
- AF (Assured Forwarding), který zajišťuje rozlišení úrovně služeb pro různé uživatele a procesy;
- DF (Default), což je standardní služba typu „Best Effort“ nezajišťující žádnou úroveň kvality služeb ani garanci výkonu.

DSCP je tvořeno šesti bity umožňujících vytvořit 64 kombinací, viz [RFC2474]:

- 1–32 jsou určeny pro standardní akce (tzv. Pool 1);
- 33–48 jsou určeny pro experimentální a lokální užití (tzv. Pool 2);
- 49–64 jsou určeny pro standardní aplikace (tzv. Pool 3), používaný pokud nevystačuje Pool 1.

Způsob implementace mechanismu PHB ukazuje tabulka 10.3.

Tabulka 10.3: Způsob implementace mechanismu PHB

	vstupní IP plánování	**výstupní IP plánování**	**ochrana před zahlcením**
EF	hlídání průměrné hodnoty, pakety mimo kontrakt jsou vyhazovány	nejvyšší priorita (odstraňování provozních špiček)	žádné prioritní pakety nelze vyhodit
AF	hlídání průměrné hodnoty dávky, pakety dávky mimo kontrakt se značkují	vyšší priorita	co je v souladu s kontraktem, nemůže být vyhozeno, je-li třeba, jsou vyhazovány pakety mimo kontrakt
BE	nic	nejnižší priorita	pakety se prioritně vyhazují

Hodnota DSCP se dále dělí na dvě tříbitové hodnoty, kde první tři bity určují třídu CS (Class Selector), další trojice bitů pak označují prioritu P (Precedence). V RFC4594 [RFC4594] byla uvedena doporučení pro značení DSCP hodnot pro různé druhy datových přenosů (jedno

z nich je uvedeno v tabulce 10.4, v tabulce 10.5 jsou uvedeny příslušné parametry přenosu), toto RFC pak bylo aktualizováno v rámci RFC 5865 [RFC5865].

Tabulka 10.4: Značení DSCP hodnot dle RFC 4594

Služba	Typ třídy	Hodnota DSCP	Hodnoty CS–P–DSCP	Použité značení pro PHB	Příklady aplikací
Administrativa	CS7	111000	7–0–56	RFC 2474	Informace pro směrování a kontrolu
Řízení sítě	CS6	110000	6–0–48	RFC 2474	Informace pro směrování a kontrolu
Telefonie	EF	101110	5–6–46	RFC 3246	IP Telefonie – přenos
Signalizace	CS5	101000	5–0–40	RCF 2474	IP Telefonie – signály
Multimediální konference	AF41 AF42 AF43	100010 100100 100110	2–4–34 4–4–36 4–6–38	RFC 2597	H.323/V2 video konference
Interaktivní komunikace v reálném čase	CS4	100000	4–0–32	RFC 2474	Video konference a interaktivní hry
Multimediální streaming	AF31 AF32 AF33	011010 011100 011110	3–2–26 3–4–28 3–6–30	RFC 2597	Přenos video a audio signálu
Broadcast video	CS3	011000	3–0–24	RFC 2474	TV a živé přenosy
Data s malým zpožděním	AF21 AF22 AF23	010010 010100 010110	2–2–18 2–4–20 2–6–22	RFC 2597	Webové klient/servet transakce
Operace a Management	CS2	010000	2–0–16	RFC 2474	Dohled nad sítí
Data s vysokou průchodností	AF11 AF12 AF13	001010 001100 001110	1–2–10 1–4–12 1–6–14	RFC 2597	Ukládání a odesílání dat pro aplikace
Standard	DF (CS0)	000000	0–0–0	RFC 2474	Pro nespecifikované aplikace
Data nízké priority	CS1	001000	1–0–8	RFC 3662	Ostatní operace

Tabulka 10.5: Hodnoty metrik pro jednotlivé třídy.

Služba	Typ třídy	Hodnota DSCP	IPTD	IPDV	IPLR
Administrativa	CS7	111000	0,05–1s	0 s	0 – 10^{-3}
Řízení sítě	CS6	110000	1–10 s	0 s	10^{-2} – 10^{-3}
Telefonie	EF	101110	100–400 ms	30–50 ms	10^{-2} – 10^{-3}
Signalizace	CS5	101000	100–400 ms	30–50 ms	10^{-2} – 10^{-3}

Multimediální konference	AF41 AF42 AF43	100010 100100 100110	100–400 ms	30–50 ms	10^{-2} – 10^{-3}
Interaktivní komunikace v reálném čase	CS4	100000	100–400 ms	30–50 ms	10^{-2} – 10^{-3}
Multimediální streaming	AF31 AF32 AF33	011010 011100 011110	5–10 ms	0 s	10^{-2} – 10^{-3}
Broadcast video	CS3	011000	nespecifikováno	nespecifikováno	nespecifikováno
Data s malým zpožděním	AF21 AF22 AF23	010010 010100 010110	20–100 ms	1–50 ms	0
Operace a Management (OAM)	CS2	010000	nespecifikováno	nespecifikováno	nespecifikováno
Data s vysokou průchodností	AF11 AF12 AF13	001010 001100 001110	1–50 ms	0 ms	0 – 10^{-3}
Standard	DF (CS0)	000000	nespecifikováno	nespecifikováno	nespecifikováno
Data nízké priority	CS1	001000	nespecifikováno	nespecifikováno	nespecifikováno

Urychlené předávání (Expedited Forwarding) má hodnotu DSCP 46 = 101110, tu obvykle dostane přenos hlasových paketů, zatímco provoz pro inicializaci telefonního hovoru bývá použita hodnota CS3. Interaktivní video dostává hodnotu AF41.

Tabulka 10.6 ukazuje způsoby konverze hodnot DSCP na IP precedenci a tabulka 10.7 značkování DSCP v závislosti na typu služby podle dokumentace Cisco.

Tabulka 10.6 Konverze hodnot DSCP na IP precedenci

PHB	Dekadická hodnota	Binární hodnota	Hodnota IP precedence
default	0	000 000	0
CS1	8	001 000	1
CS2	16	010 000	2
CS3	24	011 000	3
CS4	32	100 000	4
CS5	40	101 000	5
CS6	48	110 000	6
CS7	56	111 000	7

Tabulka 10.7: Značkování DSCP v závislosti na typu služby podle dokumentace Cisco

Doporučené značkování DSCP	
Služba	DSCP
VoIP	EF
Multicast video	AF41

Unicast video 1 (50 %)	AF42
Unicast video 2 (50 %)	AF43
Signalizace	CS3
Datový přenos	Defaultní

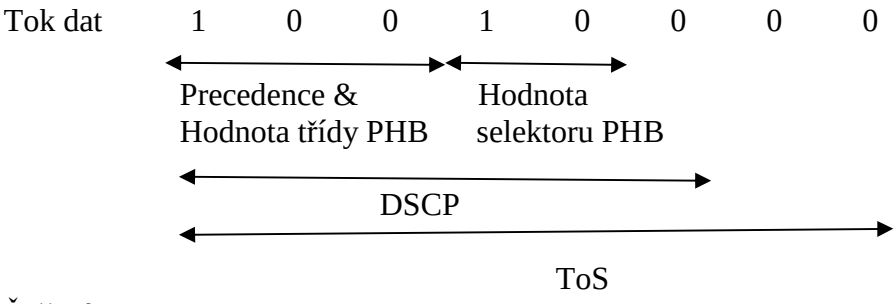
Zajištěné předávání (Assured Forwarding) je nejširší kategorie, viz tabulka 10.8. Zařízení, které podporuje IP precedenci, prověřuje jen tři bit nalevo. Každá třída obsahuje tři priority vyřazení paketu. Např. paket AF13 bude pravděpodobněji zahozen než paket AF11.
AF41 je zde nejlepší číslo a AF13 nejhorší.

Tabulka 10.8: Třídy mechanismu Assured Forwarding

Intenzita vyhazování	Třída 1	Třída 2	Třída 3	Třída 4
malá intenzita	001010 AF11 DSCP 10	010010 AF21 DSCP 18	011010 AF31 DSCP 26	100010 AF41 DSCP 34
střední intenzita	001100 AF12 DSCP 12	010100 AF22 DSCP 20	011100 AF32 DSCP 28	100100 AF 42 DSCP 36
velká intenzita	001110 AF13 DSCP 14	010110 AF23 DSCP 22	011110 AF33 33 DSCP 30	100110 AF43 DSCP 38

Příklad 10.1

Zadání



Řešení
Hodnota precedence je 4, typ třídy je AF42, DSCP je 36, ToS je 144. AF42 je nejvyšší AF podtřída s tím, že jsou třídy s vyšší i nižší prioritou při vyhazování paketů při přeplnění vyrovnávacích pamětí.

Příklad 10.2

Zadání
Tok dat 0 1 0 0 1 0 0 0

Řešení

Hodnota precedence je 2, typ třídy je AF21, DSCP je 18, ToS je 72. AF21 je druhá nejnižší podtřída, ale při vyhazování paketů při přeplnění vyrovnávacích pamětí je poslední na řadě.

Příklad 10.3
Použití značkování DSCP pro videokonferenční služby (viz RFC 4594) podle protokolu H.323:

AF41 = zvuk videokonference RTP/UDP.
AF41 = řízení obrazu videokonference RTCP/TCP.
AF41 = tok obrazu videokonference do rychlosti A.
AF42 = tok obrazu videokonference v rozmezí rychlostí rychlosti od A po B (A < B).
AF43 = tok obrazu videokonference rychlostí vyšší než B.

Kde vlastně DSCP nastavujeme?

```
Pepa(config)# class-map match-all VOIP
 1751-uut1(config-cmap)# match ip dscp ?
   <0-63>   Differentiated services codepoint value
   af11     Match packets with AF11 dscp (001010)
   af12     Match packets with AF12 dscp (001100)
   af13     Match packets with AF13 dscp (001110)
   af21     Match packets with AF21 dscp (010010)
   af22     Match packets with AF22 dscp (010100)
   af23     Match packets with AF23 dscp (010110)
   af31     Match packets with AF31 dscp (011010)
   af32     Match packets with AF32 dscp (011100)
   af33     Match packets with AF33 dscp (011110)
   af41     Match packets with AF41 dscp (100010)
   af42     Match packets with AF42 dscp (100100)
   af43     Match packets with AF43 dscp (100110)
   cs1      Match packets with CS1(precedence 1) dscp (001000)
   cs2      Match packets with CS2(precedence 2) dscp (010000)
   cs3      Match packets with CS3(precedence 3) dscp (011000)
   cs4      Match packets with CS4(precedence 4) dscp (100000)
   cs5      Match packets with CS5(precedence 5) dscp (101000)
   cs6      Match packets with CS6(precedence 6) dscp (110000)
   cs7      Match packets with CS7(precedence 7) dscp (111000)
   default  Match packets with default dscp (000000)
   ef       Match packets with EF dscp (101110)
 Pepa1(config-cmap)# match ip dscp af31
```

Podle preference lze i vybírat pakety:

```
Pepa1(config)# access-list 101 permit ip any any ?
dscp       Match packets with given dscp value
fragments  Check non-initial fragments
log        Log matches against this entry
log-input  Log matches against this entry, including input
           interface
precedence Match packets with given precedence value
time-range Specify a time-range
tos        Match packets with given TOS value
```

Pro video byly v průběhu vývoje vytvořeny „Best practices“, které zachycuje RFC 4594 [RFC4594]:

- CS4 pro interaktivní videoprovoz v reáném čase a interaktivní hry;
- CS3 pro televisní signál v rámci videa na vyžádání;
- AF4 pro video či multimediální konferenci.

Otázkou je, kolik tříd provozu zvolit. V praxi se postupuje tak, že se zvolí model o méně vrstvách a ty jsou v případě potřeby zjemňované, viz tabulka 10.9. Výchozí model člení QoS do 4 tříd, a pak je postupně zjemněn na 8 a 12 tříd.

Problém vznikne, když je třeba po některé toky přenášet po samostatné cestě (napomíná to použití bitů T2 až T0 u IP precedence). Tuto situaci nelze řešit pouze využitím IP preference nebo pomocí hodnot DSCP, řeší ho ale plně sítě MPLS (MultiProtocol Label Switching). Pokud je datový tok přenášen po stejné trase, je tato v terminologii MPLS nazývána jako Label Switched Patch (LPS). Pakety, které požadují stejné Diffserv chování, se nazývají Behaviour Aggregate (BA). Toto řešení umožňuje síťovému MPLS administrátorovi pružně volit mapování BA do LPS. Jednotlivé sady BA lze mapovat na stejnou nebo odlišné LPS.

Ve vstupním bodě do Diffserv domény jsou pakety klasifikovány a je jim přidělena hodnota DSCP odpovídající jejich BA. V každém tranzitním uzlu je pak jejich DSCP použito k výběru hodnoty PHB určující způsob plánování a (v některých případech) i pravděpodobnost vyřazení paketu.

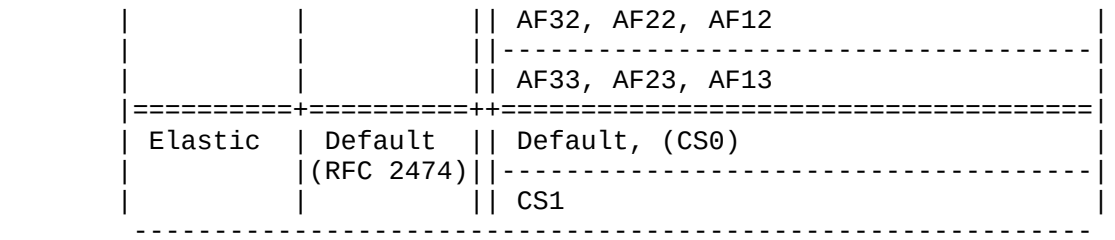
Tabulka 10.9: Příklad postupného zjemňování škály úrovní služeb.

Model o 4 službách	**Model o 8 službách**	**Model o 12 službách**
Aplikace reálného času	Hlas	Hlas
	Interaktivní video	Interaktivní komunikace v reálném čase
		Multimediální konference
	Streamované video	Širokopásmové video
		Multimediální streaming
Signalizace a řízení sítě	Signalizace	Signalizace
Kritická data	Řízení sítě	Řízení sítě
	Kritická data	Síťový management
		Transakční přenosy
		Dávkové přenosy
Bez priority (Best efford)	Bez priority (Best efford)	Bez priority (Best efford)
	Scavenger (smetí)	Scavenger (smetí)

Naopak při agregaci toků je třeba zvolit opačná postup, návrh je obsahem RFC 5127 [RFC5127] z roku 2008 – viz tabulka 10.10.

Tabulka 10.10: Agregace provozu podle RFC 5127

Agregovaný provoz	Agregovaná úroveň	DSCP
Network Control	CS (RFC 2474)	CS6
Real-Time	EF (RFC 3246)	EF, CS5, AF41, AF42, AF43, CS4, CS3
Assured Elastic	AF (RFC 2597)	CS2, AF31, AF21, AF11



Vstupující paket je v rámci Diffserv provozu v okrajovém směrovači sítě označen prioritní informací, podle které ho pak další směrovače směrují. Ať je ale hodnota DSCP jakákoliv, nemá vliv na výběr cesty. Paketu vstupujícímu do sítě MPLS je v rámci Diffserv provozu v okrajovém směrovači sítě označkovány v poli DSCP a podle této hodnoty jsou pakety přednostně obsluhovány případně vyhazovány v jednotlivých směrovačích na trase.

Značkovat lze nejen pomocí polí ToS a DSCP, ale i pomocí speciálních polí v sítích VLAN, Frame Relay, ATM a MPLS. Postupně si je zde popíšeme.

U sítí VLAN značkujeme buď podle mezinárodního standardu IEEE 802.1Q anebo v rámci konkrétního firemního standardu. Cisco má svůj starší firemní způsob zapouzdření VLAN zvaný ISL (Inter-Switch Link) a založený na externím přístupu k rámci. Zapouzdření zde tvoří 26bitové záhlaví obsahující 15bitový VLAN identifikátor a zápatí tvořené 4bytovou kontrolní sumou. Zde QoS vyjadřuje hodnota CoS (Class of Service).

Cisco přepínače rovněž respektují standard 802.1Q. Zapouzdření rámce do VLAN podle normy 802.1Q na CoS (Class of Service) je řešeno tak, že pole tag zde má tři části: prioritu (3 bity), identifikátor formátu (1 bit) a identifikátor VLAN sítě (12 bitů). Pole priority není definováno pomocí VLAN standardu, nýbrž pomocí vlastního standardu IEEE 802.1p, který je doplňkem protokolu 802.1D. Bylo stanoveno osm prioritních úrovní CoS:

0 … „best effort“ (defaultní hodnota);
1 … pozadí;
2 … standard;
3 ... pro kritické obchodní aplikace;
4 …pro multimédia;
5 …video se zpožděním < 100 ms;
6 …hlas se zpožděním < 10 ms;
7 …řízení sítě.

V případě směrovačů musíme řešit problém převodu DSCP příchozích paketů na CoS, s kterými pracují směrovače. Lze realizovat převod podle tabulky 10.11 i jakýkoliv jiný.

Tabulka 10.11: Převod DSCP na CoS.

DSCP	0-7	8-15	16-23	24-31	32-39	40-47	48-55	56-63
CoS	0	1	2	3	4	5	6	7

Příklad 10.4

Zadání

Rozhodněte o CoS IP precedenci a DSCP pro Cisco Call Manager

Řešení

Je možný postup v souladu s doporučením obsa6en7mv [Garcia]

a) Řízení hlasuVoIP

H.323 = TCP 1720, 11xxx (RAS = TCP 1719)

Skinny = TCP 2000–2002

ICCP = TCP 8001-8002

MGCP = UDP 2427, TCP 2428

CoS = 3, IP Prec = 3, DSCP = AF31 (26)

b) Vlastní přenos hlasu VoIP

UDP 16384–32767

CoS = 5, IP Prec = 5, DSCP = EF (46)

c) Řízení videa

Video Control Channels

H.323 = TCP 1720, 11xxx (RAS = TCP 1719)

CoS = 3, IP Prec = 3, DSCP = AF31 (26)

d) Přenos videa

UDP 16384-32767

CoS = 4, IP Prec = 4, DSCP = AF41 (42)

e) Přenos dat

Zde je to dost individuální

CoS = 0-2, IP Prec = 0–2, DSCP = 0–23

V záhlaví sítí Frame Relay máme bit DE (Discard Eligibility), který slouží ke značkování zahoditelných rámců při přetížení, u buněk ATM máme obdobný bit CLP (Cell Loss Priority). Před průchodem směrovačem musí být proto značka CoS (Class of Service) přeznačena na DSCP nebo ToS, neboli značka 2. vrstvy na značku 3. vrstvy, jinak by provoz ze směrovače odcházel s hodnotou CoS = 0.

I když se doporučuje značkovat provoz co neblíže u zdroje, nechceme, aby si priority nastavovali koncoví uživatelé. Proto se na přepínačích vytváří tzv. trust boundary (hranice důvěry), kdy nedůvěřujeme příchozím značkám od koncových uživatelů. Výjimkou jsou IP telefony, které značkují pakety a lze hranici důvěry rozšířit až k nim.

A nakonec – jako je tomu u IPv6? V záhlaví paketu protokolu IPv6 jsou vyčleněny dvě pole (viz obr. 10.4) sloužící k úpravě kvality služby:

- pole pro identifikaci datového toku, tzv. značka toku (Flow Lable) o velikosti 20 bitů, což je oproti IPv4 novinka; hodnota v tomto poli
  - identifikuje pakety určitého datového toku;
  - označení provedené zdrojem dat se během přenosu nemění
  - fragmentace ani šifrování nepředstavují problém jako je tomu u protokolu IPv4

Toto pole dává možnosti (zatím nevyužité) řízení toku srovnatelné s mechanismem sítě ATM virtuální cesta – virtuální kanál či BA MPLS. Do RFC 3697 se nevědělo, jak s tímto polem zacházet.

- pole tzv. třída provozu (Traffic Class) o velikosti 8 bitů;
  - funkčně ekvivalentní poli ToS v IPv4.

Obr. 10.4: Povinná část záhlaví protokolu IPv6.

Toky jsou rozlišovány podle trojice údajů: IP adresa odesílatele, IP adresa příjemce a značka toku. Shodují-li se všechny tři hodnoty, patří pakety ke stejnému toku.

Pole Flow Label pomáhá identifikovat pakety téhož datového toku, s nimiž mohou směrovače na cestě zacházet jednotným způsobem. Konkrétní mechanismy jsou však dosud pouze ve stádiu návrhů. Díky údaji v poli Flow Label směrovač provádí vyhledání ve směrovací tabulce pro daný datový tok pouze jedenkrát a výsledek vloží pro další použití do cache paměti. Není tudíž třeba pro každý paket opakovaně prohledávat ARP tabulku.

Zopakujme si v rámci tabulky 10.12, jaké máme možnosti pro značkování.

Tabulka 10.12: Přehled možností pro značkování

Značka	Oblast působení	Rozsah hodnot
IP precedence	celá síť	8 (2 rezervované)
DSCP	celá síť	64 (32 nepoužívaných)
Experimentální bit MPLS	MPLS síť (volitelně celá)	8
DE bit Frame Relay	Frame Relay síť	2 (0 a 1)
ATM CLP bit	ATM síť	2 (0 a 1)
IEE 82.1p (resp. CoS)	LAN	8 (0 až 7)
QoS Group	lokální	100 (0 až 99)

V další části kapitoly budou probírány funkční oblasti QoS:

- Klasifikace a značkování
- Omezení datových toků podle definovaných profilů
- Prioritizace odesílání paketů na médium
- Řízení zahlcení

- Vyjednávání end-to-end politik
- Speciální mechanismy pro efektivnější využití pomalých linek

10.4 Klasifikace pomocí nástroje NBAR

NBAR (Network Based Application Recognition) řeší problém s klasifikací webových aplikací anebo aplikací klient – server a poskytuje statistiky provozu. Vyhledání jednotlivých aplikací (vrstva 4 až 7) je prováděno příkazem show o syntaxi:

```
Pepa# show ip nbar [filter | pdlm | port-map | protocol-discovery |
resources | trace | unclassified-port-stats | vision]
```

například

```
Pepa(config)#interface fastethernet 0/0
Pepa(config-if)#ip nbar protocol-discovery
Pepa#show ip nbar protocol-discovery
FastEthernet0/0
                          Input                     Output
                          -----                     ------
   Protocol               Packet Count              Packet Count
                          Byte Count                Byte Count
                          5min Bit Rate (bps)       5min Bit Rate (bps)
                          5min Max Bit Rate (bps)   5min Max Bit Rate
(bps)
   ----------------------- ------------------------ ---------------------
--
   ftp                    4317                      10757
                          279012                    14127498
                          0                         62000
                          15000                     363000
   dhcp                   134                       0
                          82812                     0
                          1000                      0
                          1000                      0
   pop3                   70                        59
                          4356                      7487
                          0                         0
                          0                         1000
   smtp                   65                        67
                          6298                      5142
                          0                         0
                          0                         0
   http                   3                         2
```

Co je namapováno na konkrétním rozhraní, si lze zkontrolovat pomocí příkazu show:

```
Pepa# show ip nbar port-map

port-map bgp      udp 179
port-map bgp      tcp 179
port-map cuseeme  udp 7648 7649
port-map cuseeme  tcp 7648 7649
port-map dhcp     udp 67 68
port-map dhcp     tcp 67 68
```

Seznam podporovaných protokolů (Release 12.4T) je uveden v tabulce 10.13. Statické protokoly jsou rozpoznány na základě dobře známých čísel portů, dynamické (Citrix, FTP,

http atd.) pomocí inspekce relace. Podporovány jsou i některé protokoly, které nejsou založeny na protokolech TCP ani UDP: EGP, EIGRP, GRE, ICMP a IPSec.

Tabulka 10.13: Seznam podporovaných protokolů nástrojem NBAR.

Název protokolu	Popis protokolu
arp	IP Address Resolution Protocol (ARP)
bgp	Border Gateway Protocol
bridge	bridging
cdp	Cisco Discovery Protocol
citrix	Citrix Systems Metaframe
clns	ISO Connectionless Network Service
clns_es	ISO CLNS End System
clns_is	ISO CLNS Intermediate System
cmns*	ISO Connection-Mode Network Service
compressedtcp	komprimované TCP
cuseeme	CU-SeeMe desktop video conference
dhcp	Dynamic Host Configuration
directconnect	Direct Connect
dns	Domain Name Server lookup
edonkey	eDonkey
egp	Exterior Gateway Protocol
eigrp	Enhanced Interior Gateway Routing Protocol
exchange	Microsoft RPC for Exchange
fasttrack	FastTrack Traffic (KaZaA, Morpheus, Grokster, atd.)
finger	Finger
ftp	File Transfer Protocol
gnutella	Gnutella Version 2 Traffic (BearShare, Shareeza, Morpheus, and so on)
gopher	Gopher
gre	Generic Routing Encapsultation
h323	H323 Protocol
http	World Wide Web traffic
cmp	Internet Control Message
imap	Internet Message Access Protocol
ip	IPv4
ipinip	IP v IP
ipsec	IP Security Protocol (ESP/AH)
ipv6	IPv6

irc	Internet Relay Chat
kazaa2	Kazaa Version 2
kerberos	Kerberos
l2tp	Layer 2 Tunnel Protocol
ldap	Lightweight Directory Access Protocol
llc2	llc2
mgcp	Media Gateway Control Protocol
napster	Napster traffic
netbios	NetBIOS
netshow	Microsoft Netshow
nfs	Network File System
nntp	Network News Transfer Protocol
novadigm	Novadigm Enterprise Desktop Manager (EDM)
ntp	Network Time Protocol
ospf	Open Shortest Path First
pad*	packet assembler/disassembler (PAD) links
pcanywhere	Symantec pcANYWHERE
pop3	Post Office Protocol
printer	print spooler/ldp
rcmd	Berkeley Software Distribution (BSD) r-příkazy (rsh, rlogin, rexec)
rip	Routing Information Protocol
rsrb	Remote Source-Route Bridging
rsvp	Resource Reservation Protocol
rtp	Real-Time Protocol
rtsp	Real-Time Streaming Protocol
secure-ftp	FTP over Transport Layer Security/Secure Sockets Layer (TLS/SSL)
secure-http	Secured HTTP
secure-imap	Internet Message Access Protocol over TLS/SSL
secure-irc	Internet Relay Chat over TLS/SSL
secure-ldap	Lightweight Directory Access Protocol over TLS/SSL
secure-nntp	Network News Transfer Protocol over TLS/SSL
secure-pop3	Post Office Protocol over TLS/SSL
secure-telnet	Telnet over TLS/SSL
sip	Session Initiation Protocol
skinny	Skinny Protocol
smtp	Simple Mail Transfer Protocol
snapshot	Snapshot routing support
snmp	Simple Network Protocol
socks	SOCKS

sqlnet	Structured Query Language (SQL)*NET for Oracle
sqlserver	Microsoft SQL Server
ssh	Secured shell
streamwork	Xing Technology StreamWorks player
sunrpc	Sun remote-procedure call (RPC)
syslog	System Logging Utility
telnet	Telnet
tftp	Trivial File Transfer Protocol
vdolive	VDOLive streaming video
vofr	Voice over Frame Relay packets
xwindows	X-Windows remote access

Pokud daná aplikace není podporována, lze ji doplnit nahráním příslušného modulu PDLM (Packet Description Language Module). Není nutný update ani reboot, je však třeba nastavit CEF (Cisco Express Forwarding), což je speciální Cisco technologie hardwarového přepínání na 3. vrstvě síťové architektury.

Příklad 10.5

Zadání

Zjištění podporovaných aplikací a jejich rozšíření o PDLM modul Bit Torrentu.

Řešení

Nejprve zjistíme, zda je daná aplikace podporovaná, příkazem

```
pepa(config-cmap)#match protocol ?
```

anebo jinak

```
pepa#sh ip nbar port-map
```

Když zjistíme, že není, zavedeme příslušný PDLM modul

1) Kopírujeme PDLM modul do flash paměti směrovače:

```
pepa#copy tftp flash
Address or name of remote host []? 192.168.1.254
Source filename []? bittorrent.pdlm
Destination filename [bittorrent.pdlm]?
Accessing tftp://192.168.1.254/bittorrent.pdlm...
Erase flash: before copying? [confirm]n
Loading bittorrent.pdlm from 192.168.1.254 (via FastEthernet0.1): !
[OK – 4125 bytes]

Verifying checksum... OK (0xA1BF)
4125 bytes copied in 0.192 secs (21484 bytes/sec)
pepa#sh flash:

System flash directory:
File Length Name/status
1 9773168 c1700-k9o3sy7-mz.123-10.bin
2 4125 bittorrent.pdlm
[9777424 bytes used, 6737644 available, 16515068 total]
```

```
16384K bytes of processor board System flash (Read/Write)
```

2.) Nastavíme CEF

```
pepa#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
pepa(config)#ip cef
```

3.) Zahrneme nový modul do konfigurace

```
 pepa#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
pepa (config)#ip nbar pdlm bittorrent.pdlm
pepa (config)#
```

Výsledkem je

?

```
ip cef
ip nbar pdlm bittorrent.pdlm
!
4.) Create a class-map and policy map and apply it to the interface
concerned:
class-map match-all bittorrent
  match protocol bittorrent
!
policy-map bittorrent-policy
  class bittorrent
   drop
!
interface FastEthernet0/0
 description smer k LAN
 ip address 192.168.1.1 255.255.255.0
 ip nat inside
 service-policy input bittorrent-policy
 speed 100
 full-duplex
!
```

Další možností jak přidat další protokoly nebo aplikace je využití volby „custom“ [custom2006] a v jejím rámci vyspecifikovat příslušné porty, případně texty v zátěži, viz pár příkladů:

```
Pepa(config)# ip nbar custom aplikace1 4 ascii PEPA source tcp 4567
! 4 ASCII znaky PEPA zátěže
Pepa(config)# ip nbar custom virus 7 hex 0x56 destination udp 3000
! obsah 0x56 v sedmi bytech zátěže
Pepa(config)# ip nbar custom medium_nove 6 decimal 90 tcp 4500
```

```
!90 v 6. byte zátěže
Pepa(config)# ip nbar custom msn tcp 6700
Pepa(config)# ip nbar custom mail_x destination udp 8202
Pepa(config)# ip nbar custom mail_y destination udp range 3000 4000
```

Vraťme se k možnosti zjišťovat údaje o dané aplikaci. Opět je třeba upozornit, že musí být aktivován CEF. Většinou zjišťujeme parametry nejaktivnějších aplikací, v následujícím případě pěti aplikací:

```
Pepa# show ip nbar protocol-discovery top-n 5
 Ethernet2/0
                            Input                     Output
                            -----                     ------
  Protocol                  Packet Count              Packet Count
                            Byte Count                Byte Count
                            30sec Bit Rate (bps)      30sec Bit Rate (bps)
                            30sec Max Bit Rate (bps) 30sec Max Bit Rate
(bps)
--------------------------- ------------------------ ----------------------
--
   rtp                      3272685                   3272685
                                                 242050604
242050604
                            768000                    768000
                            2002000                   2002000
   gnutella                 513574                    513574
                            118779716                 118779716
                            383000                    383000
                            987000                    987000
   ftp                      482183                    482183
                            37606237                  37606237
                            121000                    121000
                            312000                    312000
   http                     144709                    144709
                            32351383                  32351383
                            105000                    105000
                            269000                    269000
   netbios                  96606                     96606
                            10627650                  10627650
                            36000                     36000
                            88000                     88000
   unknown                  1724428                   1724428
                            534038683                 534038683
                            2754000                   2754000
                            4405000                   4405000
   Total                    6298724                   6298724
                            989303872                 989303872
                            4213000                   4213000
                            8177000                   8177000
```

Klasifikace NBAR se často používá při výběru QoS pro konkrétní protokol:

```
Pepa> enable
Pepa# configure terminal
Pepa(config)# class-map citrix
Pepa(config-cmap)# match protocol citrix
Pepa(config-cmap)# end
```

Při výběru http komunikace zde máme celou škálu možností:

```
match protocol http
```

anebo

```
match protocol fasttrack file-transfer "*"
match protocol gnutella file-transfer "*"
match protocol kazaa2 file-transfer "*"
match protocol napster
match protocol edonkey
match protocol bittorrent
match protocol fasttrack
match protocol directconnect
match protocol winmx
```

Klasifikaci NBAR je často používána pro identifikaci různých virů. Např. na webovém serveru, který naslouchá na portu 80, jsou logovány následující záznamy, které identifikují vir Nimda:

```
GET /scripts/root.exe?/c+dir
GET /MSADC/root.exe?/c+dir
GET /c/winnt/system32/cmd.exe?/c+dir
GET /d/winnt/system32/cmd.exe?/c+dir
GET /scripts/..%5c../winnt/system32/cmd.exe?/c+dir
GET /_vti_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe?/c+dir
GET /_mem_bin/..%5c../..%5c../..%5c../winnt/system32/cmd.exe?/c+dir
GET
/msadc/..%5c../..%5c../..%5c/..\xc1\x1c../..\xc1\x1c../..\xc1\x1c../winnt/s
ystem32/cmd.exe?/c+dir
GET /scripts/..\xc1\x1c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..\xc0/../winnt/system32/cmd.exe?/c+dir
GET /scripts/..\xc0\xaf../winnt/system32/cmd.exe?/c+dir
GET /scripts/..\xc1\x9c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%35c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%35c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%5c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%2f../winnt/system32/cmd.exe?/c+dir
```

Takovéto http zprávy lze identifikovat např. tímto způsobem:

```
Pepa(config)#class-map match-any http-hacks
Pepa(config-cmap)#match protocol http url "*.ida*"
Pepa(config-cmap)#match protocol http url "*cmd.exe*"
Pepa(config-cmap)#match protocol http url "*root.exe*"
Pepa (config-cmap)#match protocol http url "*readme.eml*"
```

Další protkol, který bývá značkován, je RTP. Přistoupit k tomu jde různě, buď značkovat všechny audio RTP přenosy

```
match protocol rtp audio
```

anebo

```
match protocol rtp video
```

anebo

```
match protocol rtp payload type
```

Typy zátěže (PT – payload type) byly původně definovány v RFC 1890, ale od RFC 3551 z roku 2003 má tuto záležitost v péči IANA [RTP2011], viz tabulka 10.14.
Příklad použití (x označuje hexadecimální číslo, b binárné)

```
match protocol rtp payload-type "0, 1, 4 – 0x10, 10001b – 10010b, 64"
```

Pomocí NBAR lze také zajisti vyhazování obrázků a při http komunikaci mezi klientem a serverem. Jsou možná dvě řešení tohoto problému, buď:

```
class-map match-any OBRAZKY
match protocol http mime image/jpeg
```

Tabulka 10.14: Jednotlivé typy zátěže protokolu RTP.

Typ zátěže	Kódování	audio/video A/V	Taktovací kmitočet (Hz)	Počet kanálů	RFC
0	PCMU	A	8000	1	[RFC3551]
1	rezervováno	A	8000	1	[RFC3551]
2	rezervováno	A	8000	1	[RFC3551]
3	GSM	A	8000	1	[RFC3551]
4	G723	A	8000	1	[RFC3551] [Kumar]
5	DVI4	A	8000	1	[RFC3551]
6	DVI4	A	16000	1	[RFC3551]
7	LPC	A	8000	1	[RFC3551]
8	PCMA	A	8000	1	[RFC3551]
9	G722	A	8000	2	[RFC3551]
10	L16	A	44100	1	[RFC3551]
11	L16	A	44100	1	[RFC3551]
12	QCELP	A	8000	1	[RFC3551]
13	CN	A	8000	1	[RFC3389]
14	MPA	A	90000	1	[RFC3551] [RFC2250]
15	G728	A	8000	1	[RFC3551]
16	DVI4	A	11025	1	[DiPol]
17	DVI4	A	22050	1	[DiPol]
18	G729	A	8000	1	[RFC3551]
19	rezervováno	A			
20	nepřiděleno	A			
21	nepřiděleno	A			
22	nepřiděleno	A			
23	nepřiděleno	A			
24	nepřiděleno	V			
25	CelB	V	90000		[RFC2029]
26	JPEG	V	90000		[RFC2435]
27	nepřiděleno	V			
28	nv	V	90000		[RFC3551]
29	nepřiděleno	V			
30	nepřiděleno	V			
31	H261	V	90000		[RFC4587]
32	MPV	V	90000		[RFC2250]
33	MP2T	AV	90000		[RFC2250]
34	H263	V	90000		[Zhu]
35-71	nepřiděleno	?			
72-76	Rezervováno pro RTCP	konflikt			[RFC3551]
77-95	nepřiděleno	?			
96-127	dynamické	?			[RFC3551]

Toto řešení ale blokuje všechny obrázky. Pokud je třeba tento požadavek zablokovat v HTTP REQUESTS, ale ne v HTTP RESPONSES, je třeba volit jiný způsob:

```
class-map match-any OBRAZKY
 match protocol http url "*.gif"
 match protocol http url "*.jpeg|*.jpg"
```

a pak

```
policy-map VYHOD_OBRAZKY
 class IMAGES
   drop

interface Serial0/1
 service-policy input VYHOD_OBRAZKY
```

…

NBAR má i svá omezení:

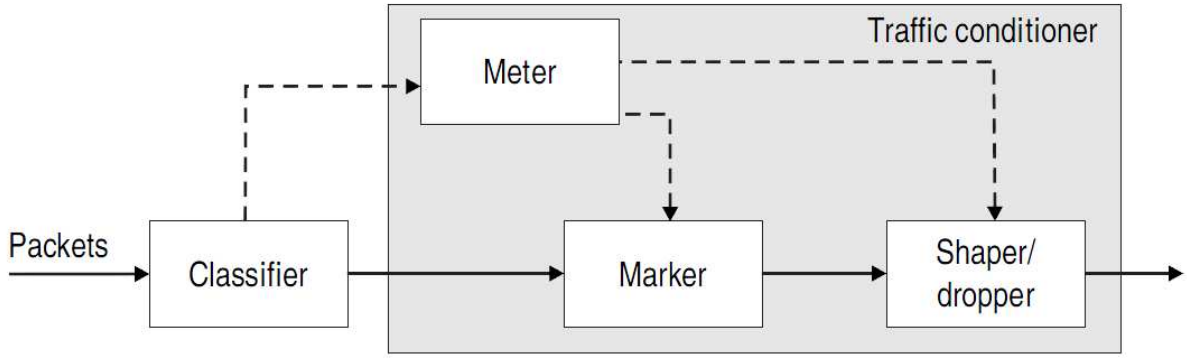
- nelze ho použít pro jiný než IP provoz
- nelze použít více než 24 souběžných URL, hostů nebo konstrukcí typu match
- match s více než 400 byte v URL
- multicast a způsoby přepínání odlišné od CEF
- fragmentované pakety
- klasifikace SHTTP
- asymetrické toky s protokoly typu stateful
- pakety tekoucí od nebo ke směrovači, na kterém běží NBAR

Také nelze NBAR konfigurovat na rozhraní:

- Fast EtherChannel
- rozhraní, které používá tunelling nebo šifrování
- VLANy
- rozhraní typu
- multilink PPP

10.5 CAR (Commited Access Rate) a PBR (Policy Based Routing)

Značkování nastavením IP preference či DSCP může být prováděno aplikací nebo uzlem sítě (viz obr. 10.5). Cisco podporuje tři varianty nastavení této funkce, a to CAR (Commited Access Rate), PBR (Policy Based Routing) a QPPB (QoS Policy Propagation using BGP (Border Gateway Control)). Poslední varianta nebude v tomto materiálu představena.



Obr. 10.5: Implementace politik na směrovačích Cisco.

CAR je starší způsob uplatnění politiky. Může být použit na příchozí i odchozí provoz. Může být použit na příchozí i odchozí provoz, nezpožďuje pakety a může být implementován pro různé způsoby značkování.

CAR poskytuje dvě funkce v jednom příkazu:

- omezení rychlosti;
- nastaveni IP precedence.

U MQC naproto CAR nastavení IP precedence či DSCP nevyžaduje současné nastavení rychlosti.

Možné akce jsou:

- transmit
- drop
- continue (běž na následující pravidlo na seznamu)
- set IP Precedence bity a transmit
- set IP Precedence bity a continue
- set skupinu QoS a transmit
- set skupinu QoS a continue

Konfigurační parametry zahrnují

- domluvenou rychlost – committed rate [b/s] / v přírustcích 8 kb/s;
- kolik bytů může být přeneseno v jedné dávce nad domluvenou rychlost, aniž by byla situace řešena – normal burst size [byte];

- kolik bytů může být přeneseno v jedné dávce nad domluvenou rychlost, aniž by byly vyhazovány – extended burst size [byte].

Cisco na základě testování TCP doporučuje následující hodnoty pro parametry rozšířené dávky:

Normální dávka = konfigurovaná rychlost * (1 byte)/(8 bitů) * 1,5 sekund

Rozšířená dávka = 2 * normální dávka

Tzn., že pokud je průměrná přenosová rychlost 10 Mb/s, pak by normální velikost dávky měla být podle [policing] 10–20 M/p a velikost rozšířené dávky 20 to 40 Mb/s. Hodnoty, které my použijeme v příkladech byly stanoveny bez nějakého kontextu

Formální zápis je

[no] rate-limit {input|output}
[access-group [rate-limit] <*acl-index*> | qos-group <*qos-group*>]
<*bps*> <*normal-burst*> <*extended-burst*>
conform-action { drop|
transmit|
continue|
set-prec-transmit <*precedence*> |
set-prec-continue <*precedence*>
set-qos-group-transmit <*qos-group*>
set-qos-group-continue <*qos-group*>}
exceed-action { drop|
transmit|
continue|
set-prec-transmit <*precedence*> |
set-prec-continue <*precedence*>|
set-qos-group-transmit <*qos-group*>|
set-qos-group-continue <*qos-group*>}

Značkování provozu může být prováděno pomocí ACL:

[no] access-list rate-limit *acl-index* {*precedence* | *mac-address* | mask *prec-mask*}

kde acl-index je číslo seznamu, při číslech 1 až 99 se pakety klasifikují podle precedence nebo precedenční masky, při číslech 100 až 199 se pakety klasifikují podle MAC adresy a 200 až 299 podle experimentálního bitu v MPLS záhlaví paketů sítí.

mask *prec-mask* je IP precedenční maska; a dvoučíslicové číslo. Používá se, pokud je třeba přiřadit vice preference stejnému rychlostnímu limitu (precedence jsou mapována do bitů: precedence 0 je 1 bit, precedence 1 je 2 bit atd.).

Relevantní kontrolní příklady jsou:

show access-lists rate-limit [*acl-index*]

show interface [*interface*] rate-limit

Příklad 10.6

Zadání

Nastavte CAR na bázi IP precedence. Pro provoz přicházející ze sítě 160.216.1.0 nastavte precedenci 5, pro provoz přicházející z jiných sítí nastavte preferenci 4.

Řešení – preference pomocí CAR

```
Pepa(config)#access-list 1 permit 160.216.1.0 0.0.0.255
Pepa(config)#interface serial 0/0
Pepa(config-if)#rate-limit access-group 1 input 2000000 2000 3000 conform-
action set prec-transmit 5 exceed-action set-prec-transmit 5
Pepa(config-if)#rate-limit input 2000000 2000 3000 conform-action set prec-
transmit 4 exceed-action set-prec-transmit 4
```

Řešení – preference pomocí PBR

```
Pepa(config)#access-list 1 permit 160.216.1.0 0.0.0.255
Pepa(config)#interface serial 0/0
Pepa(config-if)#route-map přiděleni permit 10
Pepa(config-if-route-map)#route-match ip address 1
Pepa(config-if-route-map)#set ip precedence 5
Pepa(config-if)#route-map přiděleni permit 20
Pepa(config-if-route-map)#set ip precedence 4
```

Řešení – skupiny pomocí CAR

```
Pepa(config)#access-list 1 permit 160.216.1.0 0.0.0.255
Pepa(config)#interface serial 0/0
Pepa(config-if)#rate-limit access-group 1 input 2000000 2000 3000 conform-
action set-qos-transmit 3 exceed-action set-qos-transmit 3
Pepa(config-if)#rate-limit input 2000000 2000 3000 conform-action set-qos-
transmit 0 exceed-action set-qos-transmit 0
```

Příklad 10.7

Zadání

Na pakety přicházející z rozhraní s0/0 nastavte preferenci 0, zatímco na pakety, přicházející z rozhraní s0/1 nastavte preferenci 5.

Řešení

```
Pepa(config)#interface serial 0/0
Pepa(config-if)#rate-limit input 2000000 2000 3000 conform-action set prec-
transmit 0 exceed-action set-prec-transmit 0
Pepa(config)#interface serial 0/1
Pepa(config-if)#rate-limit input 2000000 2000 3000 conform-action set prec-
transmit 5 exceed-action set-prec-transmit 5
```

Pakety přichází od zákazníka s různou preferencí a je úkolem poskytovatele pakety označené preferencí 5 upřednostňovat pře pakety s preferencí 0.

Příklad 10.8

Zadání
Omezte rychlost pro webové služby na 1 Mb/s, při nedodržení nad 2 kB snižte preference na 0, pro ftp přenosy rovněž na 1 Mb/s, při překročení o 3 kB pakety nad limit vyhazujte, u ostatních vyhazujte při překročení o 2 kB.

Řešení

```
Pepa(config)#interface serial 0/0
Pepa(config-if)#description 2Mb/s ke smerovaci R2
Pepa(config-if)#rate-limit input access-group 101 1000000 2000 3000
conform-action set-prec-transmit 5 exceed-action set-prec-transmit 0
Pepa(config-if)#rate-limit input access-group 102 1000000 2000 3000
conform-action set-prec-transmit 5 exceed-action drop
Pepa(config-if)#rate-limit input 500000 1000 2000 conform-action set-prec-
transmit 5 exceed-action drop
Pepa(config-if)#exit
Pepa(config)#access-list 101 permit tcp any any eq www
Pepa(config)#access-list 102 permit tcp any any eq ftp
```

Pomocí ACL si je možné rozdělit pásmo, viz např. následující příklad:

Příklad 10.9

Zadání

Rozdělte vhodným způsobem pásmo 2 Mb/s mezi webové, ftp a ostatní aplikace

Řešení

```
Pepa#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Pepa(config)#access-list 101 permit tcp any eq www any
Pepa(config)#access-list 101 permit tcp any any eq www
Pepa(config)#access-list 102 permit tcp any eq ftp any
Pepa(config)#access-list 102 permit tcp any any eq ftp
Pepa(config)#access-list 102 permit tcp any eq ftp-data any
Pepa(config)#access-list 102 permit tcp any any eq ftp-data
Pepa(config)#access-list 103 permit ip any any
Pepa(config)#interface serial 0/0
Pepa(config-if)#rate-limit output access-group 101 1200000 2000 3000
conform-action transmit 5 exceed-action drop
Pepa(config-if)#rate-limit output access-group 102 300000 2000 3000
conform-action transmit 5 exceed-action drop
Pepa(config-if)#rate-limit output access-group 102 500000 2000 3000
conform-action transmit 5 exceed-action drop
Pepa(config-if)#exit
Pepa(config)#end
Pepa#
```

Seznamy ACL ovšem není vždy nutné použít, viz následující příklad:

```
Pepa#configure terminal
```

```
Enter configuration commands, one per line.  End with CNTL/Z.
Pepa(config)#interface serial 0/0
Pepa(config-if)#rate-limit output dscp 14 40000 2000 3000 conform-action
transmit exceed-action drop
Pepa(config-if)#rate-limit output dscp 22 40000 2000 3000 conform-action
transmit exceed-action drop
Pepa(config-if)#rate-limit output dscp 30 40000 2000 3000 conform-action
transmit exceed-action drop
Pepa(config-if)#exit
Pepa(config)#end
Pepa#
```

U CAR jsou velmi užitečné příkazy „continue“, které umožňují vytvářet poměrně složité konstrukce:

Zadání

V případě, že provoz na výstupu rozhraní překročí 50 000 b/s, snižte precedenci webového provozu na 3 a u všeho ostatního provozu ji snižte na 0. A pokud dojde k překročení hodnoty 100 000 b/s, začněte redukovat ten ostatní provoz.

Řešení

```
Pepa#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Pepa(config)#access-list 101 permit tcp any eq www any
Pepa(config)#access-list 101 permit tcp any any eq www
Pepa(config)#access-list 102 permit ip any any
Pepa(config)#interface serial 0/0
Pepa(config-if)#rate-limit output 50000 2000 3000 conform-action transmit
exceed-action continue
Pepa(config-if)#rate-limit output access-group 101 100000 2000 3000
conform-action set-prec-transmit 3 exceed-action continue
Pepa(config-if)#rate-limit output access-group 102 100000 2000 3000
conform-action set-prec-transmit 0 exceed-action drop
Pepa(config-if)#exit
Pepa(config)#end
Pepa#
```

Příklad 10.10

Zadání

Omezte webový provoz na 50 000 b/s a celkový provoz na 100 000 b/s.

Řešení

```
Pepa(config)#access-list 101 permit tcp any eq www any
Pepa(config)#access-list 101 permit tcp any any eq www
Pepa(config)#access-list 102 permit ip any any
Pepa(config)#interface serial 0/0
Pepa(config-if)#rate-limit input 50000 2000 3000 conform-action transmit
exceed-action continue
Pepa(config-if)#rate-limit input access-group 101 100000 2000 3000 conform-
action drop exceed-action continue
Pepa(config-if)#rate-limit input access-group 102 100000 2000 3000 conform-
action transmit exceed-action drop
Pepa(config-if)#exit
Pepa(config)#end
Pepa#
```

A ještě si ukažme variantu ACL specifickou pro CAR (80 je číslo ACL a 5 hodnota precedence):

```
Pepa#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Pepa(config)#access-list rate-limit 80 5
Pepa(config)#interface serial 0/0
Pepa(config-if)#rate-limit output access-group rate-limit 80 50000 2000 3000
conform-action transmit exceed-action drop
Pepa(config-if)#exit
Pepa(config)#end
Pepa#
```

Pokud by byla použita precedenční bitová maska, je si třeba uvědomit, že má 8 bitů a precedence 1 má hodnotu masky 00000001, precedence 2 má hodnotu masky 00000011 atd a např. maska 00000111 označuje precedence 1, 2, a 3. Neboli příkaz

```
Pepa(config)#access-list rate-limit 56 mask 07

se týká preference 1, 2 a 3 (binárně 111 neboli 7).

Obdobné postupy platí pro MPLS:

Pepa(config)#access-list rate-limit 256 mask 42
```

i Mac adresy:

```
Pepa(config)#access-list rate-limit 155 0000.0c07.ac01
```

Na závěr sady příkladů si ukážeme, jak lze CAR použít pro ochranu záplavou paketů ICMP limitováním tohoto toku na max. 256 kb/s:

```
Pepa(config)#access-list 100 permit icmp any any
Pepa(config)#interface serial 0/0
Pepa(config-if)#rate-limit iutput access-group 100 256000 2000 3000
conform-action transmit exceed-action drop
```

Na závěr podkapitoly si uvedeme základní informace o modelu tzv. kbelíku (token bucket).
Daný model se opírá o představu kbelíku, který je zaplňován tokeny, což je hodnota přidělovaná jednomu paketu. Token bucket má tři komponenty: velikost dávky (burst size) někdy označovaná jako velikost smluvené dávky Bc (Committed Burst), průměrná rychlost (mean rate), někdy se používá termín CIR (Committed Information Rate), and a časový interval (Tc). Platí, že

Průměrná rychlost = velikost dávky / time interval

Používá se metafora, že příchozím paketům jsou přidělovány tokeny a ty se při předávání na rozhraní ukládají do kbelíku a z něj vysíláním zase odebírají rychlostí přenosu – s každým paketem tolik tokenů, jakou má hodnotu. Je-li kbelík tokeny zaplněn, další tokeny jsou zahazovány. Pokud není ve kbelíku dostatek tokemů, paket je označen nebo zrušen (v případě CAR) anebo čeká, až bude ve kbelíku dostatek tokenů (případ GTS). Mechanismus CAR používá jediný token bucket.

Tento příklad ukazuje, že na kumulativní (složený) dluh se v případě vyřazení paketu ruší, ale aktuální dluh zůstává.
Pro tento příklad předpokládejme tyto parametry:

- rychlost přidělování tokenu je 1 datová jednotka za časovou jednotku
- normální velikost dávky je 2 datové jednotky
- rozšířená velikost dávky je 4 datové jednotky

Po dvou časových jednotkách musí být použita normální dávka a jsme v dluhu počínajíc doby 3:

čas	příchody datových jednotek	Aktuální dluh	Složený dluh
1	2	0	0
2	2	0	0
3	2	1	1
4	2	2	3
5	2	3 (dočasně)	6 (dočasně)
5	2	2	0
6	2	3	3
7	2	4 (dočasně)	7 (dočasně)
7	2	3	0

V krocích 5 a 7 dochází k vyhození paketu a tím pádem k vynulování kumulativního dluhu.

10.6 MQC (Modular QoS Configuration)

Modulární CLI konfiguraci QoS používá Cisco od IOS 12.0(5)T, a to pro různé QoS modely ať již protokolu IP, MPLS, Frame Relay či ATM.

Modulární přístup je realizován v těchto etapách:

1. Vytvoření modulů definující třídy provozu – modulů class-map;
2. Vytvoření modulů definujících QoS politiky pro přiřazené třídy – modulů policy-map;
3. Přiřazení modulů politik na jednotlivé rozhraní (service-policy).

Prvním krokem v MQC je diferenciace (kvalifikace). K defininici třídy provozu používá MQC příkaz

`class-map`

Pakety patřící do dané třídy jsou definovány přípazem

`match`

K definici třídy provozu se moduly class-map mohou opírat o využití již probraných služeb NBAR (match protokol):

```
Pepa(config)# class-map noip
Pepa(config-cmap)# match not protocol ip
Pepa(config-cmap)# exit
```

mohou být použity i další varianty klauzule match – match access group s odkazem na ACL (Accsess Control List), např.:

```
Pepa(config)# class-map class3
Pepa(config-cmap)# match access-group 101
Pepa(config-cmap)# exit
```

dále match input-interface

```
Pepa(config) # class-map ethernet1
Pepa(config-cmap)# match match input-interface ethernet1
```

a match mpls experimental:

```
Pepa(config-cmap)# match mpls experimental topmost 3
```

nakonec ani match v modulu class být použita ani nemusí:

```
Pepa(config-pmap)# class trida1
Pepa(config-pmap-c)# bandwidth 3000
Pepa(config-pmap-c)# queue-limit 30
Pepa(config-pmap)# exit
```

Lze se odvolávat na délku paketu

```
Pepa(config-cmap)# match packet length min 100 max 300
```

na typ rozhraní

```
Pepa(config-cmap)# match port-type routed
Pepa(config-cmap)# match port-type switched

Zde značkuji všechny pakety vyjma IP:

Pepa(config)# class-map noip
Pepa(config-cmap)# match not protocol ip
Pepa(config-cmap)# exit
```

Přehled možných příkazů match, které lze použít v MQC poskytuje tabulka 10.15 převzato z [Cisco1].

Tabulka 10.15: Přehled příkazů match, které mohou být použity v MQC.

Příkaz	Kritérium
match access-group	ACL (Access Control List)
match any	Všechny pakety jsou úspěšné.
match class-map	Třída (class), třídy lze vnořovat do sebe.
match cos	Bity CoS (Class of Service) 2. vrstvy síťové architektury.
match destination address mac	Cílová MAC adresa.
match discard-class	Třída discard.
match [**ip**] **dscp**	Hodnota DSCP (Differentiated Service Code Point), v jednom příkazu match může být použito až 8 hodnot DSCP.
match field	Pole definovaná v souborech PHDF (Protocol Header Description Files.
match fr-dlci	Hodnota DLCI (Data-link Connection Identifier) sítě Frame Relay.
match input-interface	Vstupní rozhraní.
match ip rtp	Protokol RTP (Real-Time Transport Protocol) nad IP.
match mpls experimental	Experimentální bit (EXP) protokolu MPLS (Multiprotocol Label Switching (MPLS).
match mpls experimental topmost	Experimentální bit (EXP) protokolu MPLS (Multiprotocol Label Switching (MPLS) v nejvyšším záhlaví (zde se předpokládá vnoření záhlaví do sebe).
match not	Vyřazovací kritérium.
match packet length	Délka paketu uvedená v záhlaví IP paketu.
match port-type	Typ portu.
match [**ip**] **precedence**	IP precedence uvedená v záhlaví IP paketu.
match protocol	Specifický protokol. *Poznámka:* Příkaz **match protocol** (NBAR) je použit při konfiguraci NBAR (Network-Based Application Recognition) pro označení typů protokolu známých pro NBAR.
match protocol citrix	Provoz protokolu Citrix
match protocol fasttrack	Konfiguruje NBAR to k označení provozu FastTrack z třídy P2P (peer-to-peer).
match protocol	Konfiguruje NBAR to k označení provozu

gnutella	Gnutella z třídy P2P (peer-to-peer).
match protocol http	Konfiguruje NBAR to k označení provozu HTTP (Hypertext Transfer Protocol) pomocí URL, hosta, typu MIME (Multipurpose Internet Mail Extension) type nebo polí v záhlavích HTTP paketů.
match protocol rtp	Konfiguruje NBAR to k označení provozu RTP (Real-Time Transport).
match qos-group	Specifické hodnoty skupiny QoS.
match source-address mac	Zdrojová MAC adresa.
match start	Konfiguruje kritérium match na základě záhlaví datagramu (2. vrstva) nebo síťového záhlaví (3. vrstva).
match tag	Typ tag v Ethernetovém záhlaví.

Pro operace s třídou provozu se používají dvě klíčová slova, a to

- match-any, vyjadřující logickou operaci OR;
- match-all, vyjadřující logickou operaci AND.

Vezněme následující dva příklady:

```
Pepa(config)# class-map match-all zpusob1
Pepa(config-cmap)# match protocol ip
Pepa(config-cmap)# match qos-group 4
Pepa(config-cmap)# match access-group 101
```

```
a
Pepa(config)# class-map match-any zpusob2
Pepa(config-cmap)# match protocol ip
Pepa(config-cmap)# match qos-group 4
Pepa(config-cmap)# match access-group 101
```

V prvním případě musí být splněny všechny tři podmínky, aby byl paket označkován jako třídy zpusob1. Ve druhém jsou sekvenčně testovány počínajíc první uvedené a v případě splnění je paket značkován jako třídy zpusob2. Pokud není ani jediná podmínka splněna, paket ke označen jako třídy defaut-class.

Uveďme si příklady na použití klíčových slov:

```
Pepa(config)# class-map match-all podminky1
Pepa(config-cmap)# match protocol ip
Pepa(config-cmap)# match qos-group 4
Pepa(config-cmap)# match access-group 101
```

```
Pepa(config)# class-map match-any podminky2
Pepa(config-cmap)# match protocol ip
Pepa(config-cmap)# match qos-group 4
Pepa(config-cmap)# match access-group 101
```

Třída **`podminky1`** vyžaduje splnění všech podmínek, zatímco třída **`podminky2`** vyžaduje splnění jen jediné podmínky, a to kterékoliv.

Následující příklad ukazuje, způsob vytvoření klauzule class-map, v jejímž rámci probíhá výběr VLAN s ID 1000:

```
Pepa# configure terminal
Pepa(config)# class-map match-any vlan1000
Pepa(config-cmap)# match input vlan 1000
```

```
Pepa(config-cmap)# exit
```

Další příklad ukazuje, způsob vytvoření klauzule class-map, v jejímž rámci probíhá výběr VLAN s ID 100, 200, a 300:

```
Pepa# configure terminal
Pepa(config)# class-map match-any seznamID
Pepa(config-cmap)# match input vlan 100 200 300
Pepa(config-cmap)#
```

A další popisuje výběr VLAN s čísly od 2000 po 2999:

```
Pepa# configure terminal
Pepa(config)# class-map match-any tisíc_vlan
Pepa(config-cmap)# match input vlan 2000-2999
Pepa(config-cmap)#
```

A poslední příklad na téma VLAN obsahuje celý výčet jejich ID:

```
Pepa# configure terminal
Pepa(config)# class-map match-any seznamID_slozitejsi
Pepa(config-cmap)# match input vlan 1 5 10-99 2000-2499
Pepa(config-cmap)#
```

Pokud nechceme vycházet z použití modulu class-map, můžeme vyjít z její defaultní hodnoty – žádné QoS a zpracování FIFO (first-in, first-out):

```
Pepa(config)# policy-map policy1
Pepa(config-pmap)# class class-default
Pepa(config-pmap-c)# fair-queue 10
Pepa(config-pmap-c)# queue-limit 20
```

Podmínky značkování mohou být uspořádány hierarchicky, pak hovoříme o hnízděném značkování. Zde mohou nastat dvě různé situace:

a) obě třídy mají stejné charakteristiky (match-any)

```
Pepa(config)# class-map match-any trida2
Pepa(config-cmap)# match protocol ip
Pepa(config-cmap)# match qos-group 3
Pepa(config-cmap)# match access-group 2
Pepa(config-cmap)# exit

Pepa(config)# class-map match-all trida1
Pepa(config-cmap)# match class-map trida2
Pepa(config-cmap)# match destination-address mac 00.00.00.00.00.00
Pepa(config-cmap)# exit
```

Jak je vidět, toto řešení jen zkracuje u třídy trida2 zápis.

b) Obě třídy mají různé charakteristiky (match-all a match-any)

```
Pepa(config)# class-map match-all trida3
Pepa(config-cmap)# match protocol ip
Pepa(config-cmap)# match qos-group 4
Pepa(config-cmap)# exit

Pepa(config)# class-map match-any trida4
Pepa(config-cmap)# match class-map trida3
Pepa(config-cmap)# match destination-address mac 00.00.00.00.00.00
Pepa(config-cmap)# match access-group 2
Pepa(config-cmap)# exit

Pepa(config)# policy-map politika1
Pepa(config-pmap)# class trida4
```

```
Pepa(config-pmap-c)# police 8100 1500 2504 conform-action transmit exceed-
action
set-qos-transmit 4
Pepa(config-pmap-c)# end
```

Zde je třída trida3 jednou z možností třídy trida4. V tomto případě je pro politiku použita jen třída trida4. Ale to už jsme u politiky, tak se do ní pusťme.

Druhým krokem, který následuje po konfiguraci tříd, je definice QoS politik pro předtím definované třídy provozu. Tyto politiky jsou definovány příkazem:

policy-map

Příkazy, které lze použít během druhého kroku, ukazuje tabulka 10.16 převzato z [Cisco1]. Nejde o kompletní přehled, protože se použité příkazy liší v závislosti na verzi IOS.

Tabulka 10.16: Příkazy používané v rámci definice MQC politiky.

Příkaz	Účel
bandwidth	Nastavuje mechanismus CBWFQ (Class-Based Weighted Fair Queuing).
fair-queue	Specifikuje počet front, které budou rezervovány pro provoz dané třídy.
drop	Ruší pakety specifikované třídy provozu.
identity policy	Vytváří identitu politiky.
police	Konfiguruje politiku provozu.
police (control-plane)	Konfiguruje politiku provozu pro řídicí vrstvu (control plane).
police (EtherSwitch)	Definuje policer pro klasifikovaný provoz.
police (percent)	Konfiguruje politiku provozu na základě procent pásma dostupného na rozhraní.
police (two rates)	Konfiguruje politiku provozu s použitím dvou základních rychlostí sítí Frame relay – CIR (Committed Information Rate) a PIR (Peak Information Rate).
police rate pdp	Konfiguruje politiku PDP (Packet Data Protocol) založenou na rychlosti. *Poznámka:* Tento příkaz je vyhrazen pro podpůrný mód GGSN (Gateway GPRS (General Packet Radio Service) Support Node).
priority	Určuje prioritu pro třídu provozu příslušející k dané politice.
queue-limit	Specifikuje či modifikuje maximální počet paketů ve frontě pro třídu konfigurovanou v dané politice.
random-detect	Nastavuje WRED (Weighted Random Early Detection) či DWRED (Distributed WRED).
random-detect	Konfiguruje parametry WRED na hodnotu

discard-class	discard-class pro třídu v dané politice.
random-detect discard-class-based	Konfiguruje WRED na základě hodnoty paketu discard class.
random-detect ecn	Nastavuje hodnotu ECN (Explicit Congestion Notification).
random-detect exponential-weighting-constant	Konfiguruje exponenciální váhový faktor, pro průměrnou velikost fronty kalkulovanou pro frontu rezervovanou pro třídu.
random-detect precedence	Konfiguruje parametry algoritmu WRED pro hodnotu IP Precedence pro třídu politiky.
service-policy	Specifikuje název politiky použité při značkování (pro hnízděné (hierarchicky uspořádané) politiky.
set atm-clp	Pokud je konfigurována policy map, nastavuje bit CLP (Cell Loss Priority). Týká se sítí ATM.
set cos	Nastavuje na odchozí pakety bity CoS (Class of Service).
set discard-class	Značkuje pakety značkou „discard-class“.
set [**ip**] **dscp**	Značkuje pakety nastavením bitů DSCP (Differentiated Services Code Point) v poli ToS (Type of Service).
set fr-de	Mění u všech rámců opouštějících rozhraní bit DE (Discard Eligible) v adresním poli rámce Frame Relay na 1.
set mpls experimental	Určuje hodnotu, na kterou mohou být EXP bity nastaveny v případě, že pakety vyhoví podmínkám dané politiky.
set precedence	Nastavuje hodnotu precedence v záhlaví paketu.
set qos-group	Nastavuje identifikátor (ID) skupiny QoS. Ten může být později použit.
shape	Seřízne provoz na bitovou rychlost v souladu se specifikovaným algoritmem.
shape adaptive	Konfiguruje rozhraní nebo point-to-point subrozhraní Frame Relay na odhad dostupného pásma s využitím bitu BECN (Backward Explicit Congestion Notification) v případě, že je nastaven sharping.
shape fecn-adapt	Konfiguruje rozhraní Frame Relay tak, aby ve zprávách testovací odpovědi dle Q.922 vracelo obdržené bity FECN (Forward Explicit Congestion Notification) jako bity BECN (Backward Explicit Congestion Notification) a tím informovalo o zahlcení druhou stranu sítě.

Defautní třída má vždy název class-default a tyto charakteristiky:

- 10 front pro provoz, který nesplní značkovací kritéria definovaná v politice;
- maximálně 20 paketů na frontu:

```
Pepa(config)# policy-map policy1
Pepa(config-pmap)# class class-default
Pepa(config-pmap-c)# fair-queue 10
Pepa(config-pmap-c)# queue-limit 20
```

Tak jako značkování, i politiky mohou být hierarchicky uspořádané (hnízděné): Uveďme si příklad, kdy u Frame relay child policy zodpovídá za prioritní přenos a parent policy seřezává provoz kanály PVC v souladu s CIR. Neboli nejdříve je provoz seříznut na 10 Mb/s a pak dostane prioritu hlas:

```
Pepa(config)# policy-map child
Pepa(config-pmap)# class voice
Pepa(config-pmap-c)# priority 50

Pepa(config)# policy-map parent
Pepa(config-pmap)# class class-default
Pepa(config-pmap-c)# shape average 10000000
Pepa(config-pmap-c)# service-policy child
```

Pro jisté aplikace je hierarchie politik něčím přirozeným: jedna pro celé rozhraní, jiné pro každé subrozhraní. Hierarchicky jsou uspořádány i protokoly (lze např. rozčlenit aplikace sady TCP/IP na jednotlivé aplikační protokoly).

Při provádění politik je pořadí jejich vykonávání dáno interním nebo externím (intra) uspořádáním daným sekvenčními čísly.

Provádění politiky podle interních pravidel proběhne v těchto krocích:

1. Vstupní ACL
2. Návěští či precedence zdroje
3. Návěští či precedence cíle
4. Vstupní limit rychlosti
5. Směrování na základě politiky
6. Výstupní ACL
7. Výstupní limit rychlosti
8. Výstupní politika vyřazování (např. WRED)
9. Výstupní plánování (např. WFQ) a sharping

Provádění politik řízené sekvenčními čísly má výhodu, že můžeme přepsáním jednoho čísla změnit původní logiku, např. změnou hodnoty 10 na 5 otočíme pořadí provádění příkazů.

```
policy-map politika1
   class-map trida1
      rate-limit 10
   class-map trida2
      rate-limit 20

na

policy-map politika1
```

```
   class-map trida1
      rate-limit 10
   class-map trida2
      rate-limit  5
```

Vhodnou volbou politiky se můžeme např. bránit proti počítačovým útokům. Např. útok záplavou příkazů ARP (WiFi sítě atd.).

Nastavením limitu se lze bránit proti útoku typu arp scan:

```
class-map match-any arp
   match protocol arp
!
policy-map omezeni_arpu
   class arp
      police 8000 1500 1500 conform-action transmit exceed-action drop
violate-action drop
```

Následující komplexnější příklad ukazuje náročnější způsob opbrany proti jednomu z nejznámějších internetových útoků.

Příklad 10.11

Zadání

Řešte obranu proti infekci typu Code Red

Analýza

Počáteční infekce se pokouší poslat dlouhý požadavek HTTP GET na IIS server:

```
2001-08-04 16:32:23 10.101.17.216 – 10.1.1.75 80 GET /default.ida
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNN%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u
7801%u9090%u9090%u8190%u00c3%u0003%u8b00%u531b%u53ff%u0078%u0000%u00=a 403
```

Code Red II místo N používá X, ale zase je hledán soubor default.ida:

```
2001-08-04 15:57:35 10.7.35.92 – 10.1.1.75 80 GET /default.ida XXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX%u9090
%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090%
u9090%u8190%u00c3%u0003%u8b00%u531b%u53ff%u0078%u0000%u00=a 403 -
```

Další varianta Code Red vypadá takto:

2001-08-06 22:24:02 10.30.203.202 – 10.1.1.9 80 GET /x.ida AAAAAAAAAAAAAA

```
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA=X 403 HTTP/1.1
-
```

Blokovat tuto infekci lze třemi způsoby.

Řešení

Metoda A: Použití ACL na výchozí rozhraní

Pro likvidaci útoku lze použít hodnotu DSCP rovnou 1 a předpokládat, že žádný jiný provoz nebude mít tak nízkou hodnotu.

```
Pepa(config)#class-map match-any http-hacks
Pepa(config-cmap)#match protocol http url "*default.ida*"
Pepa(config-cmap)#match protocol http url "*cmd.exe*"
Pepa(config-cmap)#match protocol http url "*root.exe*"
!
Pepa(config)#policy-map mark-inbound-http-hacks
Pepa(config-pmap)#class http-hacks
Pepa(config-pmap-c)#set ip dscp 1
!
Pepa(config)#interface serial 0/0
Pepa(config-if)#service-policy input mark-inbound-http-hacks
!
Pepa(config)#access-list 105 deny ip any any dscp 1
Pepa(config)#access-list 105 permit ip any any
Pepa(config)#interface ethernet 0/1
Pepa(config-if)#ip access-group 105 out
```

Verifikace

```
Pepa#show access-list 105
Extended IP access list 105
 deny ip any any dscp 1 log (2406 matches)
 permit ip any any (731764 matches)
.
```

Metoda B: Použití směrování na bázi politiky, tzv. Policy-Based Routing (PBR)

```
Pepa(config)#access-list 106 permit ip any any dscp 1
Use the route-map command to build a routing policy.
Pepa(config)#route-map null_policy_route 10
Pepa(config-route-map)#match ip address 106
Pepa(config-route-map)#set interface Null0
Apply the route-map to the input interface.
Pepa(config)#interface serial 0/0
Pepa(config-if)#ip policy route-map null_policy_route
```

Verifikace

```
Pepa#show access-list 106
Extended IP access list 106
 permit ip any any dscp 1 (1506 matches)
!
Pepa#show log
Aug 4 13:25:20: %SEC-6-IPACCESSLOGP:
 list 105 denied tcp A.B.C.D.(0) -> 10.1.1.75(0), 6 packets
Aug 4 13:26:32: %SEC-6-IPACCESSLOGP:
list 105 denied tcp A.B.C.D.(0) -> 10.1.1.75(0), 6 packets
```

Metoda C: Použití politik na bázi tříd (Class-Based Policing)

Jde o nejvíce škálovatelnou metodu

```
Pepa(config)#policy-map drop-inbound-http-hacks
Pepa(config-pmap)#class http-hacks
Pepa(config-pmap-c)#police 1000000 31250 31250
 conform-action drop exceed-action drop violate-action drop
```

```
Pepa(config)#interface serial 0/0
Pepa(config-if)#service-policy input drop-inbound-http-hacks
```

Verifikace

```
Pepa#show policy-map interface serial 0/0
    Serial0/0
 Service-policy input: drop-inbound-http-hacks
   Class-map: http-hacks (match-any)
      5 packets, 300 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: protocol http url "*default.ida*"
        5 packets, 300 bytes
        5 minute rate 0 bps
      Match: protocol http url "*cmd.exe*"
        0 packets, 0 bytes
        5 minute rate 0 bps
      Match: protocol http url "*root.exe*"
        0 packets, 0 bytes
        5 minute rate 0 bps
      police:
        1000000 bps, 31250 limit, 31250 extended limit
        conformed 5 packets, 300 bytes; action: drop
        exceeded 0 packets, 0 bytes; action: drop
        violated 0 packets, 0 bytes; action: drop
        conformed 0 bps, exceed 0 bps, violate 0 bps
    Class-map: class-default (match-any)
      5 packets, 300 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
Match: any
```

Řešení byla převzata z [CodeRed06]

V jedné politice lze použít až 256 tříd. Každá může být v rámci politiky obsluhována vlastním mechanismem. Podporovány jsou:

- CBWFQ
- CBLLQ
- Class-Based Policing
- Class-Based Sharping
- Class-Based Marking

Třetím (finálním) krokem konfigurace metodou MQC je aplikování politiky na některé z rozhraní pomocí příkazu

service-policy

Pomocí klíčových slov input či output lze pak určit, zda se bude politika vztahovat na tok příchozích či odchozích paketů. Příkaz service-policy lze rovněž vztáhnout na virtuální kanály sítí Frame relay a logická rozhraní typu Tunnel nebo FastEtherChannel.
Jedna politika může být přiřazena na více rozhraní, ale každé rozhraní může mít jen jednu politiku ve směru do zařízení a jednu ve směru od zařízení:

```
Pepa(config)# interface ethernet1/1
Pepa(config-if)# service-policy output politika1
Pepa(config-if)# exit

Pepa(config)# interface fastethernet1/0/0
```

```
Pepa(config-if)# service-policy output politika1
Pepa(config-if)# exit
```

Uveďme si pár příkladů na aplikaci všech tří kroků MQC: postup od definice class-map přes policy-map k service policy.

Příklad 10.12
Zadání
Potřebujeme v rámci policy map ořezat provoz z VLAN na sériovém portu na nasmlouvanou hodnotu CIR (Committed Information Rate) 1 Mb/s (1000000 b/s).

Řešení
```
Pepa(config)# policy-map polika1000
Pepa(config-pmap)# class vlan1000
Pepa(config-pmap-c)# exit
Pepa(config-pmap)# shape average 10000000
Pepa(config-pmap)# interface s0/0
Pepa(config-if)# service-policy output politika1000
```

Příklad 10.13
Zadání
Analyzujte následující příkazy: Všimněte si, že pomocí ACL můžeme redukovat toky jak během prvního kroku (class-map), tak během druhého kroku (policy map). Dále si všimněte, že stejnou politiku mohu aplikovat na libovolný počet rozhraní daného zařízení.

```
Pepa(config)# class-map class1
Pepa(config-cmap)# match access-group 101
Pepa(config-cmap)# exit

Pepa(config)# class-map class2
Pepa(config-cmap)# match access-group 102
Pepa(config-cmap)# exit

Pepa(config)# policy-map politika1
Pepa(config-pmap)# class class1
Pepa(config-pmap-c)# bandwidth 3000
Pepa(config-pmap-c)# queue-limit 30
Pepa(config-pmap-c)# exit
Pepa(config-pmap)# class class2
Pepa(config-pmap-c)# bandwidth 2000
Pepa(config-pmap-c)# exit


Pepa(config)# interface ethernet1/1
Pepa(config-if)# service-policy output politika1
Pepa(config-if)# exit
Pepa(config)# interface fastethernet1/0/0
Pepa(config-if)# service-policy output politika1
Pepa(config-if)# exit
```

Významným nástrojem MQC je značkování (marking). Značkovat lze příchozí i odchozí pakety. Na vstupu může být kombinovaný s jakoukoliv QoS funkcí, na výstpu s policingrem.

Značkovat lze

- IP precedenci (0 až 7), např. `set ip precedence` *`priority`*
- IP DSCP (0 až 63 resp. jménem) např. `set ip dscp` *`af11`*
- skupinu QoS (0 až 99), např. `set qos-group 1`

- experimentální bit MPLS (0 až 7, jen na vstupu), např. `set mpls experimental 1`
- CoS bity (ISL bity Cisco) u LAN (0 až 7) např. `set cos 1`
- bit DE (Discard Eligible) u sítí Frame relay (1), např. `set fr-de`
- bit CLP u sítí Frame Relay (1, pouze na výstupu), např. `set atm-clp`

Pokud chceme zjistit, kolik paketů bylo označkování, použijeme příkaz

show policy-map interface *interface*

a daný údaj si přečteme pod položkou match.

Uveďme si komplexnější příklad na řešení s využitím značkování:

Příklad 10.14

Zadání

Uvažujme scénár, dle kterého vzdálená pobočka vzájemně komunikuje s vedením firmy a po WAN lince jsou přenášena data v rámci transakční komunikace pomocí protokolu Citrix a probíhá rovněž komunikace pomocí hlasových a video služeb. Videoprovoz zahrnuje videokonferenci v reálném čase a VOD (video-on Demand) pro e-learningový portál. E-learningová aplikace server používá kodek BMPEG (Bundled MPEG), který je konfigurován pro dynamicky sesouhlasené relace s RTP zátěží typu 97 pro relaci. Pokud by nebylo použito QoS BMPEG, VOD by spotřeboval veškeré pásmo, které by potřebovala interaktivní komunikace.

Řešení

NBAR identifikuje jednotlivé protokoly aplikační vrstvy, my si je s jeho pomocí na vstupu značkujeme a na základě těchto značek vždy na výstupu zacházíme s příslušnými toky příslušných čtyř aplikací patřičným způsobem – v tomto příkladě přidělujeme jednotlivým aplikacím procento z celkového pásma. Dané řešení může být stejné u vedení firmy i pobočky.

```
Class-map voice
  Match rtp protocol audio
Class-map video-konference
  Match rtp protocol video
Class-map vod
 Match rtp payload-type "97"
Class-map transakcni_aplikace
  Match protocol notes
Class-map interaktivni-komunikace
  Match protocol citrix
Policy-map vstupni_znackovani
  Class voice
    Set ip dscp ef
  Class video-konference
    Set ip dscp af41
  Class vod
    Set ip dscp af11
  Class transakcni_aplikace
    Set ip dscp af21
  Class interaktivni-komunikace
    Set ip dscp af31
```

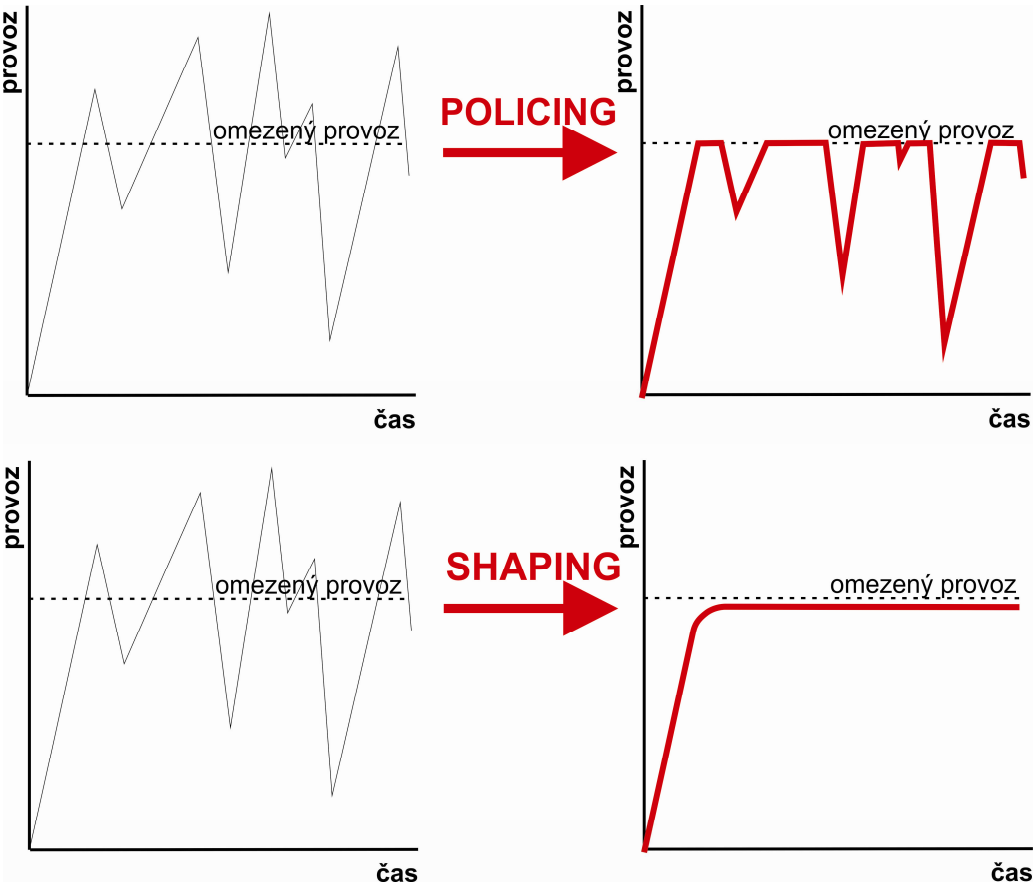
```
Interface Input
 Service-policy input vstupni_znackovani

Class-map voice
 Match ip dscp ef
Class-map video-konference
  Match ip dscp af41
Class-map vod
 Match ip dscp af11
Class-map transakcni_aplikace
  Match ip dscp af21
Class-map interaktivni-komunikace
  Match ip dscp af31
Policy-map QoS_Politika
  Class voice
    Priority percent 10
  Class video-konference
    Bandwidth remaining percent 20
  Class vod
    Bandwidth remaining percent 35
  Class transakcni_aplikace
    Bandwidth remaining percent 15
  Class interaktivni-komunikace
    Bandwidth remaining percent 30
  Class class-default
    Fair-queue
Interface Output
 Service-policy output QoS_Politika
```

10.7 Management rychlosti provozu – policing a sharping

Existují dva základní přístupy (viz obr. 10.6) k managementu rychlosti provozu na rozhraní v případě, že je provoz větší než stanovená mez, a to:

- traffic policing (uplatnění politky na provoz) – nadbytečný tok, který překračuje daný limit, je zahozen;
- traffic sharping – nadbytečný tok, který překračuje daný limit, je zařazen do speciální fronty ve vyrovnávací paměti a obsloužen, až provoz klesne pod stanovený limit.



Obr. 10.6: Porovnání efektu policingu a sharpingu: při policingu je provoz, který přesahuje limit, vyřazen bez náhrady, zatímco při sharpingu je uložen do vyrovnávací paměti a odeslán, až to omezení dovolí.

Výhodou policingu je jednoduchost (nepřidává se další fronta vyžadující paměť). Výhodou sharpingu je, že se nic nezahazuje a speciálně u Frame relay je situace řešena jeho mechanismy.

Class-based policing je vylepšenou verzí mechanismu CAR. Omezuje provoz tříd na nadstavenou bitovou rychlost. Pro měření průchozí rychlosti paketů používá model token bucket s jedním nebo se dvěmi kyblíky pro určení, zda je paket v povoleném rozmezí rychlostí (average bit ratr), zda je překračuje, ale ještě je to v povolených mezích (excess burst), či už překračuje i tyto meze (violates).

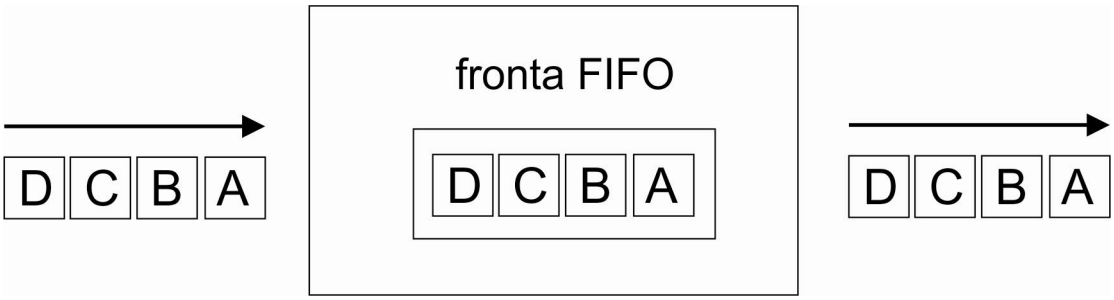
10.8 Management ochrany před zahlcením front

Cisco směrovače používají celou škálu mechanismů řazení paketů do front. Jsou zde dvě skupiny mechanismů implementovaných jako softwarové fronty:

a) Mechanismy řazení do front nevyžadující MQC (Modular QoS CLI):
   - FIFO (First in First Out)
   - PQ (Priority Queuing)
   - CQ (Custom Queuing)
   - WFQ (WFQ)
   - MDRR (Modifie Deficit Round Robin)
   - IP RTP prioritizace

b) Mechanismy řazení do front vyžadující MQC (Modular QoS CLI):
   - CBWFQ (Class Based Weighted Fair Queuing)
   - LLQ (Low Latency Queuing)

Každé rozhraní má svůj softwarový a hardwarový systém řazení do front. Hardwarový systém je vždy typu FIFO (First In – First Out) – viz obr. 10.7. Dlouhé hardwarvé fronty snižují výkon softwarových front, krátké hardwarové fronty zase vedou k volání velkého množství přerušení (a tím přetěžování procesoru).



Obr. 10.7: Struktura řazení do front typu FIFO.

Každý softwarový mechanismus řazení do front obsahuje tři hlavní komponenty:

- Klasifikace paketů,
- Určení způsobu zařazení paketu do fronty,
- Plánování vkládání paketů do hardwarové fronty.

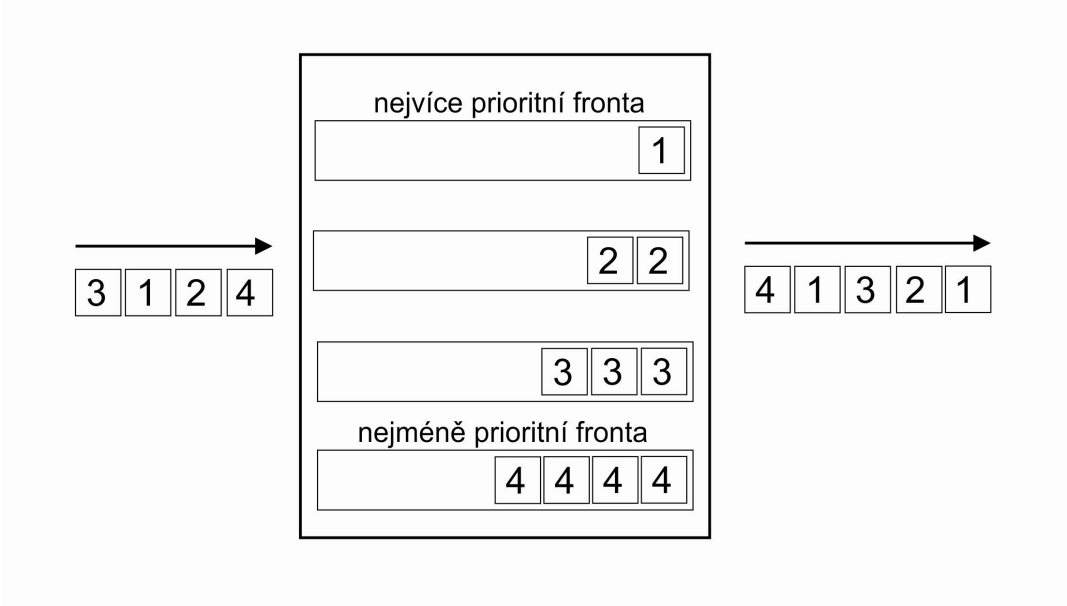
10.8.1 FIFO

Jde o jednoduchý a rychlý mechanismus podporovaný na všech platformách a verzích IOSu. Nevýhodou zde je, že nerozlišuje mezi významností toků, agresivní toky si pak mohou linky monopolizovat (starvation), vede k velkému rozkmitu (což je velmi nevýhodné zvláště pro hlas).

Způsob řazení do front FIFO je nastaven defaultně na všech rozhraních, na nichž je defautní šířka pásma rovna či větší než 2 Mb/s. Je-li méně, je nastaven způsob WFQ, ale jeho vypnutím systém přejde na FIFO. U Cisco lze do výstupní FIFO fronty defaultně vložit až 40 paketů, příkazem `hold-queue <buffers> out` ale lze tuto hodnotu změnit.

10.8.2 PQ (Priority Queuing)

U tohoto mechanismu (viz obr. 10.8) se hledí jen na priority. Režim dokáže vyhladovět provoz s nízkou prioritou.



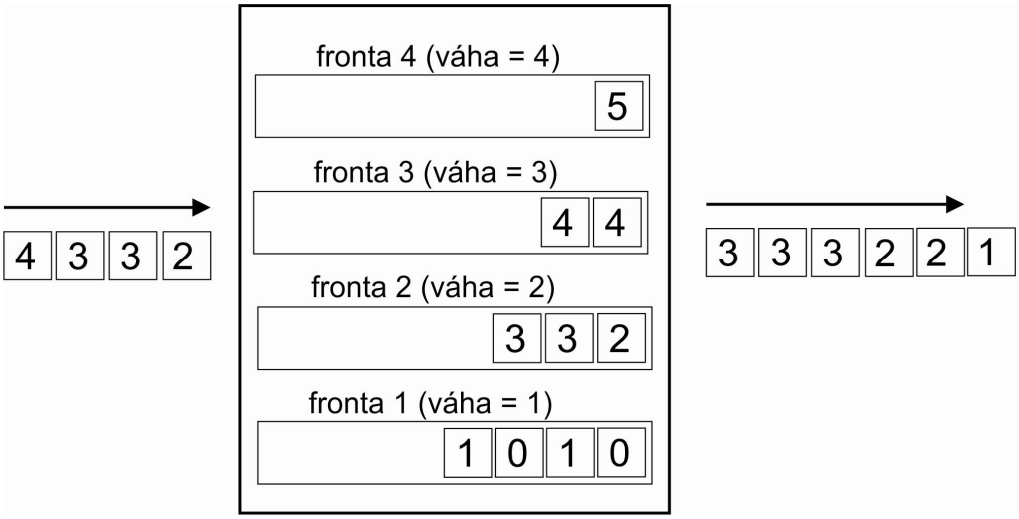
Obr. 10.8: Mechanismus Mechanismus PQ (Priority Queuing).

10.8.3 WFQ (Weighted Fair Queuing)

Snaží se o rozdělování pásma s přihlédnutím k jejich váze (důležitosti). Jeho výhodou je, že zahazuje pakety nejvíce agresivních toků.

Od verze 12.0(5)T je váha počítána podle vztahu `váha = 32768 / (IP Prec. + 1)`

Indexy front, do kterých je paket umístěn, jsou vypočteny hashováním ze zdrojové a cílové IP adrey, zdrojového a cílového portu, čísla transportního protokolu a hodnoty pole ToS (Type of Service). Důležité je, že počet nakonfigurovaných front musí být větší než očekávaný počet toků. Problémem je, že do hashe nelze zasáhnout a že zde nejsou poskytnuty žádné pevné garance, jde jen o cyklickou obsluhu (viz obr. 10.9).



Obr. 10.9: Cyklická obsluha a přidělování pásma podle vah u mechanismu WFQ.

Syntax je:

`fair-queue [cdt [dynamic/queues [reservation/queues]]]`

kde `cdt` (Congestive Discard Threshold) je počet zpráv, od kterého směrovač začne zahazovat nové pakety té nejdelší frontě.¨

10.8.4 IP RTP

Tento mechanismus se kombinuje s WFQ nebo CBWFQ (přidává jednu speciální prioritní frontu) a je použitelný prouze pro UDP provoz s předpovědnutelnými čísly portů (obvykle VoIP). Zabraňuje požírání provozu nějakým provozem, má však malé možnosti klasifikace a je proto nahrazován mechanismem CBLLQ.

10.8.5 CBWFQ (Class-Based Weighted Fair Queuing)

CBWFQ je mechanismus, který jednotlivým třídám garantuje provozu šířku pásma. Neboli funkčnost WFQ je zde doplněna o podporu uživatelsky definovaných tříd provozu.

Mechanismus WFO na rozdíl od PQ zajišťuje, že žádný provoz nevyhladoví, čili není opomíjen. Ani jeden však nezajistí dostupnost pásma pro definované typy provozu. Pomocí mechanismu CB-WFQ lze naopak určit minimální šířku pásma, a to pro 64 tříd provozu. Nevyhladoví ani provoz s nižší prioritou, jako je tomu u PQ.

Příklad 10.15

Zadání

V rámci mechanismu CBWFQ omezte provoz protokolu Citrix na polovinu pásma.

Řešení

```
Pepa> enable
Pepa# configure terminal
Pepa(config)# class-map citrix
Pepa(config-cmap)# match protocol citrix
Pepa(config-cmap)# end
Pepa> enable
Pepa# configure terminal
Pepa(config)# policy-map pol
Pepa(config-pmap)# class citrix
Pepa(config-pmap-c)# bandwidth percent 50
Pepa(config-pmap-c)# end
Pepa> enable
Pepa# configure terminal
Pepa(config)# interface ethernet 2/4
Pepa(config-if)# service-policy input pol
Pepa(config-if)# end
```

Příklad 10.16

Zadání

Omezení provozu P2P (předtím blokován) na pásmo 8 kb/s.

Řešení

Byl zvolen mechanismus QoS CBWFQ (Class-Based Weighted Fair Queue).

```
class-map match-any p2p
 match protocol bittorrent
 match protocol edonkey
 match protocol fasttrack
 match protocol gnutella
 match protocol kazaa2
!
policy-map QoS-inbound-policy
 class p2p
   police cir 8000
     conform-action drop
     exceed-action drop
!
interface FastEthernet0
 description Facing LAN
 ip address 192.168.1.1 255.255.255.0
 ip nat inside
 service-policy input QoS-inbound-policy
 speed 100
 full-duplex
!
```

10.8.6 LLQ

Jediná nevýhoda mechanismu CBWFQ je nedostatek mechanismů pro prioritní řízení, což řeší jeho drobná úprava, která se nazývá LLQ. LLQ může jedné nebo více třídám provozu nařídit provoz směrovat do prioritní fronty. Je si ale třeba uvědomit, že umístěním paketu do prioritní fronty nepřidělujeme tomuto provozu pouze šířku pásma, ale také policing (omezení dostupné šířky pásma), aby provoz s nižší prioritou nevyhladověl.

LLQ je typ řazení preferovaný pro provoz citlivý na zpoždění.

Příklad 10.17

```
!Nejprve vytvoříme mapu tříd pro provoz VoIP
Pepa(config)#class-map match-all voice-traffic
Pepa(config-cmap)#match access-group 102
Pepa(config)#access-list 102 permit udp any any range 16384 32776
! Uvedené porty slouží pro VoIP přenos s využitím protokolu H.323
access-list 102 permit udp any any precedence critical
! Vyfiltrujeme kritický provoz nebo
access-list 102 permit udp any any dscp ef
! Vyfiltrujeme pakety s hodnotou pole dscp rovnou ef
Access-list 102 permit udp host 192.10.1.1 host 192.20.1.1
class-map voice
match ip rtp 16384 16383
```

```
class-map voice
match ip precedence 5
! nebo
class-map voice
match ip dscp ef
! Nyní se budeme věnovat signalizaci
class-map voice-signaling
match access-group 103
!
access-list 103 permit tcp any eq 1720 any
access-list 103 permit tcp any any eq 1720
! VoIP  přenos můžeme realizovat pomocí protokolů H.323, SIP, MGCP
! nebo Skinny. Jednotlivé protokoly používají tyto porty:
! H.323/H.225 = TCP 1720
! H.323/H.245 = TCP 11xxx (Standard Connect)
! H.323/H.245 = TCP 1720 (Fast Connect)- naše volba
! H.323/H.225 RAS = TCP 1719
! Skinny = TCP 2000-2002 (CM Encore)
! ICCP = TCP 8001-8002 (CM Encore)
! MGCP = UDP 2427, TCP 2428 (CM Encore)
! SIP= UDP 5060, TCP 5060 (configurable)
! Create a Policy Map and Associate to the VoIP Class-Maps
Pepa(config)#policy-map VOICE-POLICY
Pepa(config-pmap)#class voice-traffic
Pepa(config-pmap-c)#priority ?
<8-2000000>  Kilo Bits per second
! Je třeba vybrat přenosovou rychlost
Pepa(config-pmap)#class voice-signaling
Pepa(config-pmap-c)#bandwidth 8
Pepa(config-pmap)#class class-default
Pepa(config-pmap-c)#fair-queue
Zbývající provoz je typu WFQ
Pepa(config)#interface multilink 1
Pepa(config-if)#service-policy output VOICE-POLICY
! O multilinku později
```

10.8.7 WRR (Weighted Roud Robin)

Tento mechanismus kombinuje váhy a cyklickou obsluhu. Používáse pouze u přepínačů Catalyst.

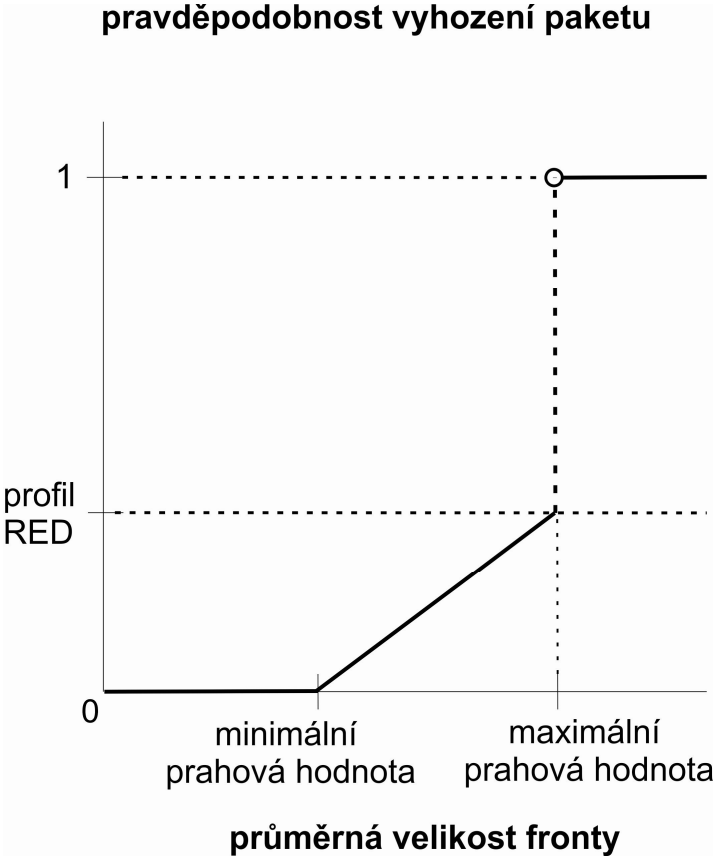
10.9 Předcházení zahlcení

K tomuto účelu slouží několik mechanismů:

- RED (Random Early Detection)
- WRED (Weighted RED)
- CBWRED (Class-Based WRED)

10.9.1 RED (Random Early Detection)

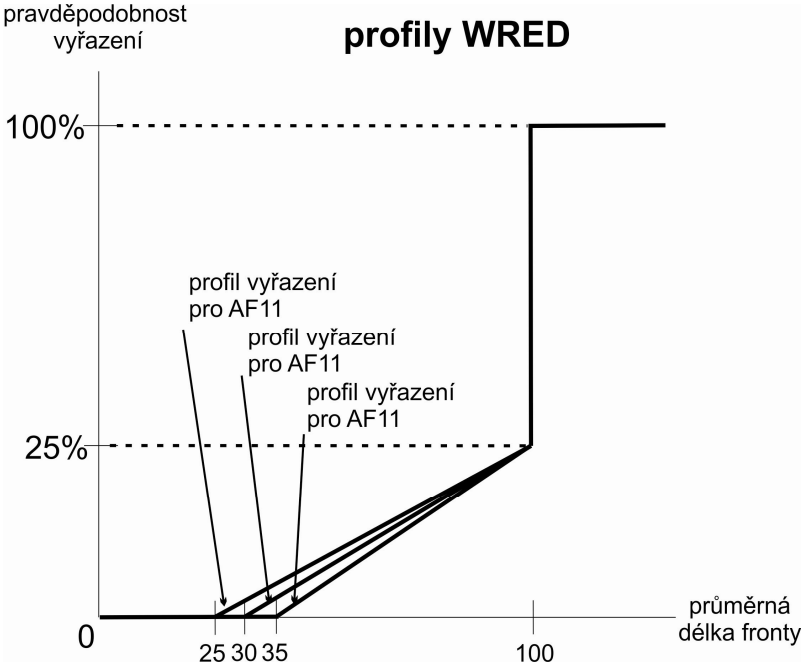
RED je mechanismus, který náhodně zahazuje pakety, dokonce dříve, než se fronta zaplní. Tento mechanismus je určen třemi hodnotami: minimální prahová hodnota, maximální prahová hodnota a pravděpodobnost, při které se skokem přechází na plné zahazování paketů (viz obr. 10.10).



Obr. 10.10: Ukázka použití mechanismu RED (jediný profil).

10.9.2 WRED (Weighted RED) a CBWRED (Class-Based WRED)

Na rozdíl od mechanismu RED mechanismus WRED vytváří profil pro každé značení priority (viz obr. 10.11).



Obr. 10.11: Ukázka použití mechanismu WRED (profil vyřazení je různý pro různé třídy provozu).

Příklad 10.18

Paket s precedencí 0 má minimální limit 20 paketů (v případě přetížení se začnou zahazovat dříve).

Paket s precedencí 1 má minimální limit 25 paketů.

Class-Based WRED může být použit v kombinaci s CBWFQ.

10.10 Fragmentace a prokládání

U příliš dlouhých datových paketů dochází k jevu zvanému „starving“, kdy hlasové pakety příliš dlouho čekají na jejich zpracování, což vde k příliš velkému rozptylu jejich dob doručení. Proto jsou příliš dlouhé pakety na pomalých linkách (pod 2 Mb/s) fragmentovány. V praxi se používají tři mechanismy fragmentace a prokládání (viz obr. 10.12), angl. LFI (Link Fragmentation and Interleaving):

- MLPP (také označení MP, MPL, MPPP či Multilink),
- FRF.12 – VoIPoFR (Voice over Frame Relay),
- FRF.11 Annex C – u linek VoFR.

Při konfiguraci mechanismu LFI musí být zpoždění serializace v rozsahu od 10 do 35 ms.

Obr. 10.12: Příklad prokládání dlouhého paketu D krátkými pakety V.

Příklad 10.19

Rámec 512 B na lince o 128 kb/s má zpoždění serializace (512 *8)/128 = 32 ms, neboli nevyhovuje požadavku. Je ho třeba fragmentovat.

MLP lze povolit příkazem `Pepa(config-if) # ppp multilink interleave`

Závěrem zajímavý komplexní příklatý z [Alawadhi].

Příklad 10.20

Zadání

Implentace QoS pro domácí ADSL připojení do Internetu.

Řešení

Vytvoříme si třídu protokolů typu P2P a do ní zařadíme protokoly BitTorrent, FastTrack, Gnutella, Kazaa, and Napster. Pojmenujeme třídu jako p2p.

```
pepa (config)# class-map match-any p2p
pepa (config-cmap)# match protocol bittorrent
pepa (config-cmap)# match protocol fasttrack
pepa (config-cmap)# match protocol gnutella
pepa (config-cmap)# match protocol kazaa2
pepa (config-cmap)# match protocol napster
```

Konfigurujeme ostatní třídy:

```
pepa (config)# class-map match-any voip
pepa (config-cmap)# match protocol sip
pepa (config-cmap)# exit
pepa (config)# class-map match-any important
pepa (config-cmap)# match protocol icmp
pepa (config-cmap)# exit
pepa (config)# class-map match-any high-priority
pepa (config-cmap)# match protocol secure-http
pepa (config-cmap)# match protocol ssh
pepa (config-cmap)# match protocol dns
pepa (config-cmap)# match protocol ntp
pepa (config-cmap)# exit
pepa (config)# class-map match-any normal
pepa (config-cmap)# match protocol http
pepa (config-cmap)# match protocol ftp
pepa (config-cmap)# match protocol pop3
pepa (config-cmap)# match protocol smtp
pepa (config-cmap)# exit
```

Definujeme politiku s využitím IP precedencí:

```
pepa (config)# policy-map ma_politika
pepa (config-pmap)# class voip
pepa (config-pmap-c)# set ip precedence 6
pepa (config-pmap-c)# exit
pepa (config-pmap)# class important
pepa (config-pmap-c)# set ip precedence 6
pepa (config-pmap-c)# exit
pepa (config-pmap)# class high-priority
```

```
pepa (config-pmap-c)# set ip precedence 5
pepa (config-pmap-c)# exit
pepa (config-pmap)# class normal
pepa (config-pmap-c)# set ip precedence 3
pepa (config-pmap-c)# exit
pepa (config-pmap)# class p2p
pepa (config-pmap-c)# police 10000 5000 5000 conform-action set-prec-
transmit 2 exceed-action set-prec-transmit 1
```

U P2P přenosů je do 10 kb/s přidělována priorita 2, při překročení je snížena na 1.

Přiřadíme politiky na rozhraní

```
pepa (config)# interface FastEthernet0
pepa (config-if)# service-policy input ma_politika
pepa# wr mem
```

Ověříme správnost nastavení pomocí stahování většího souboru pomocí BitTorrent a sledování použitého pásma, pak dáme ping anebo zavoláme a měla by rychlost stahování souboru poklesnout.

Použité zdroje

[Alawadhi] Alawadhi, A., *Implementing QoS for home ADSL using Cisco routers.* http://knol.google.com/k/cisco-and-qos#.

[Avaya2006] *Avaya IP Telephony implementation Guide*. Communication Manager 3.1. Avaya Labs, May 2006. http://www.scribd.com/doc/55784679/75/Appendix-F-Sample-QoS-Configurations.

[Cisco1] *Applying QoS Features Using the MQC*. Materiály Cisco. http://www.cisco.com/en/US/docs/ios/qos/configuration/guide/qos_mqc_ps6441_TSD_Products_Configuration_Guide_Chapter.html.

[Cisco2] *Enterprise Quality of Service*. Prezentace Cisco Live Networkers. TECRST-2500.

[CiscoDoc2006] *Cisco documentation*, 2006, http://www.cisco.com/univercd/cc/td/doc/solution/vobbsols/vob1/vbdig/vbappc1.htm.

[CodeRed06] *Using Network-Based Application Recognition and ACLs for Blocking the "Code Red" Worm*. Document ID: 27842. Aug 02, 2006, http://www.cisco.com/en/US/products/hw/routers/ps359/products_tech_note09186a00800fc176.shtml.

[custom2006] *Creating a Custom Protocol*. Last Updated: April 4, 2006. http://www.cisco.com/en/US/docs/ios/qos/configuration/guide/nbar_cust_protcl.pdf.

[ITU-TE.800] *ITU-T Recommendation E.800. Terms and definitions related to Quality of Service and Network Performance Including Dependability*. 08/94. http://wapiti.telecom-lille1.eu/commun/ens/peda/options/ST/RIO/pub/exposes/exposesrio2008-ttnfa2009/Belhachemi-Arab/files/IUT-T%20E800.pdf.

[Ethernet2010] *Ethernet Network Switches*. Miercom Lab Testing Summary Report 100827. September 2010. http://www.cisco.com/en/US/prod/collateral/switches/ps5718/ps6406/miercom_report_cisco_catalyst_2960_3750_switches_qos.pdf.

[Garcia] Garcia, A. QOS For IP Video Conference. Cisco Systems 11/14/2001.

[ITU-TY.1541] *Series Y: Global Information Infrastructure, Internet Protocol Aspects and Next-Generation Networks*. Internet protocol aspects – Quality of service and network performance. ITU-T Y.1541. (02/2006).

[RFC791] *Defense Advanced Research Projects Agency*. InternetProtocol Darpa Internet Program Protocol Specification. RFC 791, September 1981.

[RFC1946] Jackowski, S. *Native ATM Support for ST2+*. RFC 1946. 05/96. http://tools.ietf.org/html/rfc1946.

[RFC2474] Nichols, K., Blake, S., Baker, F., Black, D. *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*. RFC 2474. December 1998. http://www.ietf.org/rfc/rfc2474.txt.

[RFC2475] Blake, S, Black, D., Carlson, M., Davies, E., Wang, Z., Weiss W. *An Architecture for Differentiated Services*. RFC 2475, December 1998. http://www.ietf.org/rfc/rfc2475.txt.

[RFC4594] Babiarz, J., Chan, K., Baker, F. *Configuration Guidelines for DiffServ Service Classes*. RFC 4594. August 2006.

[RFC5127] Chan, K, Babiarz, Baker, F. *Aggregation of Diffserv Service Classes*. RFC 5127. http://www.rfc-archive.org/getrfc.php?rfc=5127.

[RFC5865] Baker, F., Dolly, M. *A Differentiated Services Code Point (DSCP) for Capacity-Admitted Traffic*. RFC 5865, May 2010. http://tools.ietf.org/html/rfc5865.

[RTP2011] *Real-Time Transport Protocol (RTP) Parameters* (last updated 2011-05-18, http://www.iana.org/assignments/rtp-parameters).

[Satrapa] Satrapa, P. *Internetový protokol IPv6*. CZ NIC. Praha, 2008. ISBN: 9788090424807. Internet : http://www.root.cz/knihy/internetovy-protokol-ipv6/