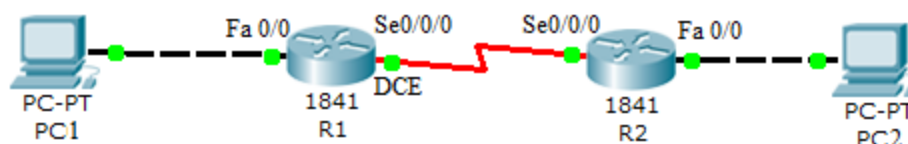


# Laboratórne cvičenie 2

## Ciele

- Konfigurácia a testovanie stratégie FIFO
- Konfigurácia a testovanie stratégie PQ (Priority Queuing)
- Konfigurácia a testovanie stratégie CQ (Custom Queuing)
- Konfigurácia a testovanie stratégie WFQ (Weighted Fair Queuing)

## Graf topológie



## Adresovanie

Zariadenie	Rozhranie	IP Adresa	Maska Subsiete	Východzia Brána
R1	S 0/0/0	172.16.12.33	255.255.255.252	N/A
	Fa 0/0	172.16.12.1	255.255.255.240	N/A
R2	S 0/0/0	172.16.12.34	255.255.255.252	N/A
	Fa 0/0	172.16.12.17	255.255.255.240	N/A
PC1	sieťová karta	172.16.12.10	255.255.255.240	172.16.12.1
PC2	sieťová karta	172.16.12.27	255.255.255.240	172.16.12.17

## Scenár

V tomto cvičení využijete poznatky z laboratórneho cvičenia 1 a naučíte sa konfigurovať 4 typy frontových stratégií podporovaných na smerovačoch Cisco spolu s príkazmi na výpis nakonfigurovanej stratégie a príkazmi na zobrazenie obsahu front. Toto všetko sa bude diať počas streamovania videa prostredníctvom VLC prehrávača a generovania sieťovej prevádzky D-ITG generátorom.

## Príprava

Je potrebné zapojiť topológiu, na ktorej sa bude testovať laboratórne cvičenie, podľa obrázku [Graf Topológie](#). Ďalej potom, ako nastavíme správnu frontovú stratégiu, bude potrebné spustiť streamovanie videa tak, ako to bolo uvedené v predchádzajúcom cvičení a tiež spustiť program na generovanie sieťovej prevádzky D-ITG.

## A. Konfigurácia a testovanie stratégie FIFO

### 1. Krok: Konfigurácia stratégie FIFO

Nastavenie stratégie FIFO na Cisco smerovačoch je veľmi jednoduché, pretože táto stratégia je väčšinou, ale nie vždy, nastavená ako základná. Preto je na smerovači potrebné overiť, ktorá stratégia je aktuálne aktivovaná. Vždy môže byť implementovaná len jedna. Frontové stratégie je vhodné konfigurovať na rozhrania, na ktoré to má zmysel, u nás je to rozhranie serial 0/0/0 na smerovači R1, keďže tam budú pakety čakať na obsluhu aby mohli byť prenesené sériovou linkou k smerovaču R2. Nasledujúcim príkazom je možné overiť aktuálne používanú stratégiu, pridaný je aj neúplný výpis.

```
R1#show interface serial 0/0/0

Serial0/0/0 is up, line protocol is up

  Hardware is GT96K Serial

  Internet address is 172.16.12.33/30

  MTU 1500 bytes, BW 400 Kbit/sec, DLY 20000 usec

  reliability 255/255, txload 39/255, rxload 12/255

  Encapsulation HDLC, loopback not set

  Keepalive set (10 sec)

  Last input 00:00:06, output 00:00:03, output hang never

  Last clearing of "show interface" counters 03:16:26

  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 70327

  Queueing strategy: fifo
```

```
Output queue: 40/40 (size/max)
```

V prípade, že stratégia FIFO nie je aktuálne nakonfigurovaná, je potrebné zadať príkaz, ktorý vypne všetky ostatné frontové stratégie, a tým sa aktivuje stratégia FIFO.

```
R1#configure terminal
```

```
R1(config)#interface serial 0/0/0
```

```
R1(config-if)#no fair-queuing
```

## 2. Krok: Testovanie stratégie FIFO

### 1. Test

Ak ešte nie je spustené streamovanie videa, je potrebné ho zapnúť, a tiež je potrebné vyslať prúd dát generovaných D-ITG generátorom. V tomto teste sa budú generovať UDP pakety. Stratégia FIFO pracuje len s jedným frontom, do ktorého pakety zaraďuje podľa času ich príchodu a nijako ich neusporadúva. Následne ich v rovnakom poradí vysiela rozhraním, na ktorom je implementovaná.

Pri spúšťaných testoch budeme sledovať zmeny na streamovanom videu, ktoré spôsobí generovaná sieťová prevádzka z D-ITG.

Príkazom sa začne odchyťvanie paketov pre potreby štatistík, ktoré sa uložia do súboru na strane príjemcu s názvom testUDPPrev1(program si tento súbor sám vytvorí).

ITGRecv.exe -l testUDPPrev1 ( je potrebné ho zadať v príkazovom riadku v adresári DITG generátora)

Nasledovným príkazom, aplikovanom na strane odosielateľa, vyšle generátor prúd UDP paketov na IP adresu 172.16.12.27, pričom budú mať veľkosť 500 B a bude ich vygenerovaných a vyslaných 100 za sekundu. Test bude trvať 30 sekúnd.

ITGSend.exe -m rttm -a 172.16.12.27 -sp 9400 -rp 9500 -C 100 -c 500 -t 30000 -l testUDPSend1 ( je potrebné ho zadať v príkazovom riadku v adresári DITG generátora)

Sledovaním streamovaného videa môžeme vidieť a počuť, že na obraze a ani na zvuku sme počas testu prakticky nezaznamenali žiadne zmeny (sekanie/vypadávanie obrazu alebo zvuku). Keď na strane príjemcu zadáme príkaz :

```
ITGDec.exe testUDPPrev1
```

```

Flow number: 1 *
From 172.16.12.10:9400
To 172.16.12.27:9500
-----
Total time = 29.995000 s 1.
Total packets = 3000 2.
Minimum delay = 0.289654 s 3.
Maximum delay = 0.309997 s 4.
Average delay = 0.292769 s 5.
Average jitter = 0.002431 s 6.
Delay standard deviation = 0.002741 s 7.
Bytes received = 1500000 8.
Average bitrate = 400.066678 Kbit/s 9.
Average packet rate = 100.016669 pkt/s 10.
Packets dropped = 0 (0.00 %) 11.
-----

***** TOTAL RESULTS *****
Number of flows = 1
Total time = 29.995000 s
Total packets = 3000
Minimum delay = 0.289654 s
Maximum delay = 0.309997 s
Average delay = 0.292769 s
Average jitter = 0.002431 s
Delay standard deviation = 0.002741 s
Bytes received = 1500000
Average bitrate = 400.066678 Kbit/s
Average packet rate = 100.016669 pkt/s
Packets dropped = 0 (0.00 %)
Error lines = 0 12.
-----

```

Otvorí sa súbor obsahujúci štatistiku týkajúcu sa prvého testu vygenerovaných paketov (výsledky testov sa môžu meniť). Z jeho výpisu je možné vidieť počet generovaných prúdov (\*), celkový čas generovania (1.), celkový počet vygenerovaných paketov (2.), minimálne (3.), maximálne (4.) a priemerné (5.) oneskorenie, priemerné kolísanie oneskorenia (6.), odchýlku oneskorenia (7.), počet prijatých bajtov (8.), priemernú rýchlosť prenášania bitov (9.), priemernú rýchlosť prenášania paketov (10.), počet zahodených paketov (11.) a počet chýb na linke (12.).

Nielen to, že sme nezaregistrovali chyby videa a zvuku, ale aj výpis štatistík hovorí, že danou linkou možno preniesť väčší tok dát, preto vrámci ďalšieho testu zvýšime počet a veľkosť generovaných paketov cez D-ITG.

## 2. Test

V druhom teste zväčšíme počet generovaných paketov na hodnotu 300 paketov za sekundu (tiež je potrebné mať stále zapnuté streamovanie videa). Ostatné parametre nemeníme. Na strane príjemcu zadáme príkaz

ITGRecv.exe -l testUDPrecv2

Na strane odosielateľa zadáme príkaz

```
ITGSend.exe -m rttm -a 172.16.12.27 -sp 9400 -rp 9500 -C 300 -c 500 -t 30000 -l  
testUDPSend2
```

V priebehu generovania sieťovej prevádzky badáme zhoršenie kvality streamovaného videa na strane príjemcu v podobe kociek a tieňov na obraze a taktiež zhoršenie zvuku videa v podobe jeho sekania a oneskorenia. Zhoršená kvalita bola spôsobená nadmerným počtom paketov prichádzajúcich na sériové rozhranie 0/0/0 a s tým spojené zahadzovanie (strata) paketov, medzi ktorými bolo aj streamované video. Po ukončení generovania sieťovej prevádzky D-ITG generátorom prenos streamovaného videa pokračuje v kvalite, v akej bolo vysielané pred generovaním paketov.

Preťaženie sériovej linky a tým spôsobené zahadzovanie paketov vidíme aj vo výsledných štatistikách, ktoré nám poskytuje generátor sieťovej prevádzky na strane príjemcu. Je dobré všimnúť si aj zvýšenie časov oneskorenia oproti predchádzajúcemu testu. Na strane príjemcu zadáme ITGDec.exe testUDPrecv2

```
Flow number: 1  
From 172.16.12.10:9400  
To 172.16.12.27:9500  
-----  
Total time = 30.508000 s  
Total packets = 7944  
Minimum delay = 0.292664 s  
Maximum delay = 0.915503 s  
Average delay = 0.672809 s  
Average jitter = 0.002167 s  
Delay standard deviation = 0.187138 s  
Bytes received = 3972000  
Average bitrate = 1041.562869 Kbit/s  
Average packet rate = 260.390717 pkt/s  
Packets dropped = 1056 <11.73 %>  
-----  
***** TOTAL RESULTS *****  
-----  
Number of flows = 1  
Total time = 30.508000 s  
Total packets = 7944  
Minimum delay = 0.292664 s  
Maximum delay = 0.915503 s  
Average delay = 0.672809 s  
Average jitter = 0.002169 s  
Delay standard deviation = 0.187138 s  
Bytes received = 3972000  
Average bitrate = 1041.562869 Kbit/s  
Average packet rate = 260.390717 pkt/s  
Packets dropped = 1056 <11.73 %>  
Error lines = 0  
-----
```

### 3. Test

V tomto teste zvýšime počet generovaných paketov na 400 za sekundu, a aj ich veľkosť na hodnotu 800 B. Ostatné veličiny ponecháme nezmenené. Príkaz na strane príjemcu je nasledovný

ITGRecv.exe testUDPrecv3

Na strane odosielateľa príkaz vyzerá nasledovne

```
ITGSend.exe -m rttm -a 172.16.12.27 -sp 9400 -rp 9500 -C 400 -c 800 -t 30000 -l  
testUDPsend3
```

Po tom, ako sa začnú generovať pakety opäť sledujeme streamované video na strane príjemcu. Môžeme si všimnúť značné znehodnotenie obrazu v podobnej miere ako v 2. teste, alebo až zastavenie obrazu. Zvuk sa ihneď po začatí generovania paketov cez D-ITG stratí. Keď zadáme príkaz „show interface serial 0/0/0“ (v priebehu testu) na smerovači R1 vidíme, že maximálna veľkosť frontu je 40 paketov a aktuálna veľkosť frontu je tiež 40 paketov. Toto značí, že je front plný a pakety ktoré sa do neho nezmestia, smerovač zahodzuje.

```
R1#show interface serial 0/0/0  
  
Serial0/0/0 is up, line protocol is up  
  
  Hardware is GT96K Serial  
  Internet address is 172.16.12.33/30  
  
  MTU 1500 bytes, BW 400 Kbit/sec, DLY 20000 usec,  
    reliability 255/255, txload 39/255, rxload 12/255  
  
  Encapsulation HDLC, loopback not set  
  
  Keepalive set (10 sec)  
  
  Last input 00:00:06, output 00:00:03, output hang never  
  
  Last clearing of "show interface" counters 03:16:26  
  
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 70327  
  
  Queueing strategy: fifo  
  
  Output queue: 40/40 (size/max)
```

Aj výsledný súbor vytvorený generátorom na strane príjemcu nám ukazuje, že sériová linka bola v priebehu generovania paketov značne preťažená, čo môžeme vidieť na počte zahodených paketov, ktorý je prirodzene ešte väčší ako pri 2. teste. Takisto aj zvýšené časy oneskorenia sú dôkazom preťaženia linky. Príkazom sa zobrazí štatistika pre generované pakety

ITGDec.exe testUDPrecv3

```

Flow number: 1
From 172.16.12.10:9400
To 172.16.12.27:9500
-----
Total time = 30.603000 s
Total packets = 6206
Minimum delay = 0.306384 s
Maximum delay = 1.004509 s
Average delay = 0.936429 s
Average jitter = 0.002060 s
Delay standard deviation = 0.080054 s
Bytes received = 4964800
Average bitrate = 1297.859687 Kbit/s
Average packet rate = 202.790576 pkt/s
Packets dropped = 5794 (48.28 %)
-----

***** TOTAL RESULTS *****
Number of flows = 1
Total time = 30.603000 s
Total packets = 6206
Minimum delay = 0.306384 s
Maximum delay = 1.004509 s
Average delay = 0.936429 s
Average jitter = 0.002060 s
Delay standard deviation = 0.080054 s
Bytes received = 4964800
Average bitrate = 1297.859687 Kbit/s
Average packet rate = 202.790576 pkt/s
Packets dropped = 5794 (48.28 %)
Error lines = 0
-----

```

## B. Konfigurácia a testovanie stratégie Priority Queuing

Stratégia Priority Queuing umožňuje rozdeliť sieťovú prevádzku do jedného zo štyroch frontov výstupnej linky smerovača na základe rôznych kritérií: použitého protokolu, prichádzajúceho rozhrania, zdrojovej alebo cieľovej adresy, zdrojového alebo cieľového portu. Fronty sa nazývajú podľa svojej priority: High, Medium, Normal, Low. Smerovač ich obsluhuje podľa ich priorít, to znamená, že neobsluží front s menšou prioritou, pokiaľ sa nachádzajú nejaké pakety vo fronte s vyššou prioritou. A taktiež, keď príde paket do prázdneho frontu High, ak smerovač obsluhuje aktuálne iný front, ihneď prácu ukončí a prechádza na obsluhovanie frontu High. Hovoríme, že fronty sú obsluhované s absolútnou prioritou. Stratégia PQ sa môže použiť aj v kombinácii s ACL (Access list).

### 1. Krok: Konfigurácia stratégie PQ

Pri konfigurácii PQ je potrebné najskôr vytvoriť tzv. zoznam s prioritnými frontami (priority-list), v ktorom definujeme globálne triediace metódy. Podľa nich sa bude sieťová prevádzka triediť do jednotlivých front.

Ďalej je možné nastaviť veľkosti frontov, toto je však voliteľné, pretože fronty majú základne nastavené svoje veľkosti nasledovne: high 20, medium 40, normal 60, low 80.

Následne je potrebné globálne vytvorený zoznam s prioritnými frontami aplikovať na niektoré rozhranie. V našom prípade na sériové rozhranie 0/0/0 na smerovači R1.

Na testovanie stratégie PQ by sme mohli použiť generovanie sieťovej prevádzky v podobe TCP paketov, popri streamovanom videu, pričom prevádzka videa by bola zaradená do frontu s prioritou high a D-ITG generovaná sieťová prevádzka by bola zaradená do frontu s mešou prioritou, čiže napríklad do frontu medium. Výsledok takéhoto testovania, by však priniesol výsledky, ktoré by nám neukázali v štatistických súboroch vytvorené D-ITG generátorom, že niektoré TCP pakety boli zahodené, a tiež video by zostalo neporušené (keďže tieto pakety sme predtým zaradili do fronty high). Avšak vzhľadom na to, že TCP pakety vyžadujú potvrdenie o doručení zo strany príjemcu (TCP protokol zabezpečuje, že poslané dáta sa dostanú k príjemcovi v rovnakom poradí a bez chýbajúcich častí), znamená to, že prakticky každý paket bude doručený, pretože keď odosielateľovi nepríde potvrdenie o doručení paketu, zopakuje jeho zaslanie.

My budeme testovať prácu PQ nasledovne. Vytvoríme zoznam s prioritnými frontami, v ktorom definujeme, že do frontu High bude zaraďovaná prevádzka prichádzajúca z portu 1234 (streamované video) a do frontu Medium bude zaraďovaná prevádzka generovaná D-ITG generátorom (prichádzajúca z portu 9400). Upravíme tiež veľkosť frontu Medium. Potom aplikujeme vytvorený zoznam s prioritnými frontami na sériové rozhranie 0/0/0. Príkazy sú nasledovné.

```
R1(config)#priority-list 2 protocol ip high udp 1234
R1(config)#priority-list 2 protocol ip medium udp 9400
R1(config)#priority-list 2 queue-limit 20 100 60 80
R1(config)#interface serial 0/0/0
R1(config-if)#priority-group 2
```

V týchto testoch, ktoré sa týkajú testovania frontovej stratégie PQ budeme okrem paketov nesúcich streamované video používať aj UDP pakety generované D-ITG generátorom.

### Test

Vykonáme rovnaký test ako v 3. teste v časti testovanie FIFO stratégie a budeme sledovať v čom sa dané testy odlišujú. Počas toho ako sa streamuje video, zadáme nasledovný príkaz na strane príjemcu.



ITGRecv.exe -I PQrecvUDP1

Na strane odosielateľa zadáme príkaz

ITGSend.exe -m rttm -a 172.16.12.27 -sp 9400 -rp 9500 -C 400 -c 800 -t 30000 -I PQsendUDP1

Počas toho ako sú generované a prenášané pakety nebedať na streamovanom videu žiadne zmeny, čo sme docielili vhodným uprednostnením streamovaného videa pred prevádzkou generovanou generátorom D-ITG. Vytvorený súbor nám poskytuje informácie o tom, že smerovač pakety generované generátorom nestíhal obsluhovať. Je to vidieť na číslach, ktoré percentuálne vyjadruje, koľko týchto paketov bolo zahodených.

```
Flow number: 1
From 172.16.12.10:9400
To 172.16.12.27:9500
-----
Total time = 29.994000 s
Total packets = 8850
Minimum delay = 0.498596 s
Maximum delay = 0.570998 s
Average delay = 0.563647 s
Average jitter = 0.001296 s
Delay standard deviation = 0.003340 s
Bytes received = 7080000
Average bitrate = 1888.377676 Kbit/s
Average packet rate = 295.059012 pkt/s
Packets dropped = 3122 (26.08 %)
-----
***** TOTAL RESULTS *****
-----
Number of flows = 1
Total time = 29.994000 s
Total packets = 8850
Minimum delay = 0.498596 s
Maximum delay = 0.570998 s
Average delay = 0.563647 s
Average jitter = 0.001296 s
Delay standard deviation = 0.003340 s
Bytes received = 7080000
Average bitrate = 1888.377676 Kbit/s
Average packet rate = 295.059012 pkt/s
Packets dropped = 3122 (26.08 %)
Error times = 0
-----
```

Tento výpis sa zoprazí po aplikovaní nasledovaného príkazu na strane príjemcu

ITGDec.exe PQrecvUDP1

Výsledky testov nám naznačujú, že pri vhodne nastavenej stratégii Priority Queuing dáva táto stratégia lepšie výsledky ako stratégia FIFO.

## C. Konfigurácia a testovanie stratégie Custom Queuing

Stratégia Custom Queuing dovoľuje rozdeliť sieťovú prevádzku do niekoľkých rôznych frontov. Jej rozdelenie je založené na type informácie, akú daný paket prenáša. Je ju možné použiť aj v kombinácii so zoznamom na kontrolu prístupu (ACL – access control list). Vlastnosti na základe ktorých je možné rozdeliť sieťovú prevádzku sú transportný, alebo aplikačný protokol, čísla portov, tzv. differentiated service code point (DSCP) alebo tzv. IP precedence, vstupné rozhranie.

Hlavnou výhodou stratégie CQ je, že vie proporcionálne alokovať šírku pásma pre rôzne triedy sieťovej prevádzky. Môže použiť až 16 frontov pre zradovanie paketov pred samotným ukladaním paketov na výstupnú linku, ktoré obsluhuje spôsobom Round – Robin. Znamená to, že jednotlivé fronty sa obsluhujú cyklicky, pričom každý jeden front môže vyslať v jednom kole len predkonfigurovaný počet bytov (túto veličinu je možné meniť). Taktiež je možné flexibilne konfigurovať veľkosť a počet použitých frontov. Toto nám umožňuje efektívne a proporcionálne stanoviť koľko šírky pásma chceme priradiť ktorému frontu.

### 1. Krok: Konfigurácia stratégie CQ

Pri konfigurácii tejto stratégie je potrebné najskôr vytvoriť zoznam frontov a definovať globálne pravidlá, podľa ktorých sa bude rozdeľovať sieťová prevádzka do jednotlivých frontov.

Ďalej môžeme využiť možnosť upraviť veľkosti jednotlivých frontov a tiež počet bajtov, ktoré budú v každom cykle z frontu vyňaté a spracované (odoslané príjemcovi).

Nakoniec je potrebné aplikovať daný zoznam frontov na konkrétne rozhranie (u nás serial 0/0/0). Nesmieme zabudnúť, že na jednom rozhraní môže byť aktuálne implementovaná len jedna frontová stratégia, a len jeden zoznam frontov.

Cisco IOS dovoľuje vytvoriť 1 – 16 frontových zoznamov a v jednom zozname frontov definovať až 16 (1 - 16) rôznych frontov, pričom front 0 sa používa napríklad na kontrolu prevádzky (ICMP).

My nakonfigurujeme zoznam front 1, ktorý bude obsahovať dva fronty. Do prvého frontu sa na základe portu 1234 budú ukladať pakety streamovaného videa a do druhého frontu sa budú na základe portu 9400 ukladať pakety generovanej sieťovej UDP prevádzky. Príkazy aplikujeme na sériové rozhranie 0/0/0 na smerovači R1.

```
R1(config)#queue-list 1 protocol ip 1 udp 1234
```

```
R1(config)#queue-list 1 protocol ip 2 udp 9400
```

Následne vytvorený zoznam front priradíme na spomínané rozhranie.

```
R1(config)#interface serial 0/0/0
```

```
R1(config-if)#custom-queue-list 1
```

## 1. Test

Počas streamovania videa zadáme na strane príjemcu príkaz

ITGRecv.exe -I CQsendUDP1

Následne môžeme začať generovať pakety D-ITG generátorom príkazom

ITGSend.exe -m rttm -a 172.16.12.27 -sp 9400 -rp 9500 -C 100 -c 500 -t 30000 -I CQsendUDP1

Pri generovaní sieťovej prevádzky si všimame, že nemá žiadny vplyv na streamované video. Znamená to, že technika CQ zvláda spracovať obidva typy sieťovej prevádzky, čo nám ukazuje výstupný súbor generátora v počte nula zahodených paketov.

```
Flow number: 1
From 172.16.12.10:9400
To 172.16.12.27:9500
-----
Total time = 29.990000 s
Total packets = 3000
Minimum delay = 12.443883 s
Maximum delay = 12.464997 s
Average delay = 12.447651 s
Average jitter = 0.002385 s
Delay standard deviation = 0.002799 s
Bytes received = 1500000
Average bitrate = 400.133378 Kbit/s
Average packet rate = 100.033344 pkt/s
Packets dropped = 0 (0.00 %)
-----

***** TOTAL RESULTS *****
-----
Number of flows = 1
Total time = 29.990000 s
Total packets = 3000
Minimum delay = 12.443883 s
Maximum delay = 12.464997 s
Average delay = 12.447651 s
Average jitter = 0.002385 s
Delay standard deviation = 0.002799 s
Bytes received = 1500000
Average bitrate = 400.133378 Kbit/s
Average packet rate = 100.033344 pkt/s
Packets dropped = 0 (0.00 %)
Error times = 0
-----
```

Dôležité je si všimnúť ako sa zmenilo priemerné oneskorenie !

## 2. Test

Pri nasledovanom teste začneme generovať sieťovú prevádzku, pričom zvýšime počet generovaných paketov. Príkaz pre príjemcu

ITGRecv.exe -I CQsendUDP2

A nasledovným príkazom pre odosielateľa

ITGSend.exe -m rttm -a 172.16.12.27 -sp 9400 -rp 9500 -C 300 -c 500 -t 30000 -I CQsendUDP2

Pri prebiehajúcom teste sledujeme striedanie okamihov, v ktorých sa video prehráva správne a okamihov, kedy je jeho kvalita poškodená (kocky a tieň). Je to spôsobené cyklickou obsluhou zoznamu frontov. To, že v priebehu testu nebolo niečo v poriadku, potvrdí aj výstupný súbor generátora D-ITG v podobe zahodených paketov.

```
Flow number: 1
From 172.16.12.10:9400
To 172.16.12.27:9500
-----
Total time = 30.077000 s
Total packets = 7500
Minimum delay = 12.455330 s
Maximum delay = 12.568534 s
Average delay = 12.518955 s
Average jitter = 0.003233 s
Delay standard deviation = 0.039059 s
Bytes received = 3750000
Average bitrate = 997.439904 Kbit/s
Average packet rate = 249.359976 pkt/s
Packets dropped = 1500 (16.67 %)
-----

***** TOTAL RESULTS *****
-----
Number of flows = 1
Total time = 30.077000 s
Total packets = 7500
Minimum delay = 12.455330 s
Maximum delay = 12.568534 s
Average delay = 12.518955 s
Average jitter = 0.003233 s
Delay standard deviation = 0.039059 s
Bytes received = 3750000
Average bitrate = 997.439904 Kbit/s
Average packet rate = 249.359976 pkt/s
Packets dropped = 1500 (16.67 %)
Error times = 0
-----
```

### 3. Test

Následne spustíme rovnaký test ešte raz, ale najskôr zmeníme parametre prvého frontu v zozname frontov nasledovnými príkazmi

```
R1(config)# queue-list 1 queue 1 limit 100
```

```
R1(config)# queue-list 1 queue 1 byte-count 3000
```

Týmito príkazmi sme zabezpečili, že 1. front v zozname frontov bude maximálne môcť obsahovať 100 paketov, pričom v každom cykle sa z neho odošle 3000 bytov dát.

Na strane príjemcu spustíme príkaz

```
ITGRecv.exe -I CQsendUDP3
```

A na strane odosielateľa príkaz

```
ITGSend.exe -m rttm -a 172.16.12.27 -sp 9400 -rp 9500 -C 300 -c 500 -t 30000 -I CQsendUDP3
```

Výsledok tohto testu je taký, že nebadat' známky poškodenia na obraze ani na zvuku videa, no výstupný súbor z D-ITG generátora nám ukáže, že niektoré generované pakety museli byť zahodené z dôsledku naplnenia frontu.

Flow number: 1	
From 172.16.12.10:9400	
To 172.16.12.27:9500	
-----	
Total time	= 30.095000 s
Total packets	= 6906
Minimum delay	= 12.464339 s
Maximum delay	= 12.614369 s
Average delay	= 12.535268 s
Average jitter	= 0.003236 s
Delay standard deviation	= 0.045511 s
Bytes received	= 3453000
Average bitrate	= 917.893338 Kbit/s
Average packet rate	= 229.473334 pkt/s
Packets dropped	= 2094 (23.27 %)
-----	
***** TOTAL RESULTS *****	
-----	
Number of flows	= 1
Total time	= 30.095000 s
Total packets	= 6906
Minimum delay	= 12.464339 s
Maximum delay	= 12.614369 s
Average delay	= 12.535268 s
Average jitter	= 0.003236 s
Delay standard deviation	= 0.045511 s
Bytes received	= 3453000
Average bitrate	= 917.893338 Kbit/s
Average packet rate	= 229.473334 pkt/s
Packets dropped	= 2094 (23.27 %)
Error lines	= 0
-----	

## D. Konfigurácia a testovanie stratégie Weighted Fair Queuing

WFQ je jednou zo základných frontových stratégií na Cisco smerovačoch. Je nastavená ako základná stratégia na rozhraniach pomalších ako 2048 Mbps (sériová linka), v inom prípade je ako základná frontová stratégia použitá stratégia FIFO (napr. FastEthernet rozhranie). Princíp, na ktorom táto technika funguje, je založený na príslušnosti paketov k jednotlivým tokom, pričom tok existuje len vtedy, ak existuje paket, ktorý do neho patrí a nachádza sa v ňom. V inom prípade tento tok a aj front pre neho vytvorený zaniká. Tok môže byť identifikovaný na základe zdrojovej/ cieľovej IP adresy, zdrojového/ cieľového portu, alebo tzv. IP precedence. WFQ poskytuje šírku pásma tak, že menší paket s vyššou prioritou dostane výhodnejšiu alokáciu šírky pásma, ako väčší paket s nižšou prioritou a tým sa vyhne blokovaniu paketov. Maximálny počet vytvorených frontov je 4096. WFQ dynamicky vytvorí konverzný front ihneď ako prijme paket patriaci do toku, ktorému nebol doposiaľ priradený žiadny front na danom rozhraní. A naopak tento front zruší ihneď ako je z neho poslaný posledný paket.

### 1. Krok: Konfigurácia WFQ stratégie

Pred nasledujúcimi testami je potrebné odstrániť konfiguráciu z predchádzajúcej úlohy príkazmi

```
R1(config)#interface serial 0/0/0  
R1(config-if)#no custom-queue-list 1
```

Základná konfigurácia WFQ stratégie v podstate ani nie je potrebná z dôvodu, že WFQ je základná stratégia použitá na linkách s rýchlosťou menšou ako 2048 Mbps. Z toho dôvodu by mala byť táto stratégia aktívna na našom sériovom rozhraní 0/0/0. Overíme to príkazom

```
R1# show interface serial 0/0/0
```

#### Test

V tomto teste si ukážeme, že metóda WFQ je schopná dynamického vytvorenia frontu pre ICMP pakety a tieto pakety uprednostniť pred paketmi, generovanými D-ITG generátorom ako aj pred paketmi streamovaného videa. Vzhľadom na to, že WFQ uprednostňuje menšie pakety s vyššou IP prioritou, obraz videa a aj zvuk by mali zostať neporušené. Video by nemali pokaziť ani pakety ICMP, ktoré bez problémov „prekĺžu“ pred D-ITG generovanými paketmi.

Na strane príjemcu spustíme nasledovný príkaz

ITGRecv.exe -I WFQsendUDP1

A na strane odosielateľa otvoríme ďalší príkazový riadok, pričom si do neho nachystáme nasledovný príkaz, ktorý vygeneruje 30 ICMP paketov a následne zobrazí štatistiku o doručení.

ping 172.16.12.27 -n 30

Na strane odosielateľa tiež zadáme príkaz

ITGSend.exe -m rttm -a 172.16.12.27 -sp 9400 -rp 9500 -C 400 -c 1000 -t 30000 -I CQsendUDP3

Následne spustíme aj pripravený príkaz ping a sledujeme zmeny na obraze a zvuku.

Výsledkom je nezmenená kvalita videa a výstupný súbor D-ITG generátora, v ktorom vidíme, že niektoré pakety generované z D-ITG boli zahodené. Zahodenie paketov je spôsobené preťažením linky, teda nedostatkom šírky pásma. Aplikovaním frontového mechanizmu na preplnenú linku nemôžeme očakávať zázračné zlepšenie služieb. Problémy preplnenej linky by úplne vyriešilo zvýšenie šírky pásma.

```
Flow number: 1
From 172.16.12.10:9400
To 172.16.12.27:9500
-----
Total time = 30.420000 s
Total packets = 4093
Minimum delay = 12.536999 s
Maximum delay = 13.078159 s
Average delay = 12.963857 s
Average jitter = 0.003363 s
Delay standard deviation = 0.065074 s
Bytes received = 4093000
Average bitrate = 1076.397107 Kbit/s
Average packet rate = 134.549638 pkt/s
Packets dropped = 7906 (65.89 %)
-----

***** TOTAL RESULTS *****
-----
Number of flows = 1
Total time = 30.420000 s
Total packets = 4093
Minimum delay = 12.536999 s
Maximum delay = 13.078159 s
Average delay = 12.963857 s
Average jitter = 0.003363 s
Delay standard deviation = 0.065074 s
Bytes received = 4093000
Average bitrate = 1076.397107 Kbit/s
Average packet rate = 134.549638 pkt/s
Packets dropped = 7906 (65.89 %)
Error lines = 0
-----
```

Následne môžeme vidieť vo výslednom výpise po vykonaní príkazu ping, že všetky ICMP pakety boli poslané a aj odozvy na ne boli prijaté.

Výsledky týchto testov potvrdili našu teóriu, že ICMP pakety budú uprednostnené.