Facultatea de Electronică, Telecomunicații și Tehnologia Informației

Controlul QoS prin mecanismul DiffServ

Profesor îndrumător:
Conf. Dr. Ing. Ştefan Stăncescu

Student:
Bosnea Marinela

Master 1: IISC

Cuprins

1. Introducere-QoS

Un şir de pachete de la o sursă la o destinaţie se numeşte flux. Într-o reţea orientată pe conexiune, toate pachetele aparţinând unui flux merg pe aceeaşi rută; într-o reţea neorientată pe conexiune, acestea pot urma rute diferite. Necesităţile fiecărui flux pot fi caracterizate prin patru parametri primari: fiabilitatea, întârzierea, fluctuaţia şi lăţimea de bandă. Împreună, acestea determină QoS (Quality of Service – rom.: Calitatea serviciilor) pe care o necesită fluxul.

Calitatea serviciilor (în engleză, quality of service, scurt **QoS**) este capacitatea de a oferi prioritate pentru diferite genuri de aplicații soft, utilizatori, fluxuri de date, sau garantarea unui anumit nivel de performanță pentru fluxul de date. QoS este utilizat pentru a gestiona cerințele speciale pentru comunicațiile de voce și video, furnizarea datelor cu o rată de întârziere mică (de obicei mai puțin de 250 milisecunde), ameliorarea bruiajelor (de obicei mai puțin de 10 sau 20 de milisecunde), reducerea pierderilor de pachete (de obicei mai puțin de 0,5 % din pachete). Depășirea acestor valori poate provoca calitatea proastă a apelurilor VoIP.

QoS are de asemenea un rol important în rețelele cu receptivitate sporită. Deseori, atunci când unul sau mai multe dispozitive (cum ar fi laptop-uri) din rețea au fost compromise cu software rău intenționat, cum ar fi un virus sau un vierme, ele pot fi folosite ca un punct de lansare pentru atacuri asupra altor părți ale rețelei, cum ar fi dispozitive de rețea sau servere de aplicații. În cazul în care un astfel de atac vine din afara rețelei, de ex. de pe Internet, sau de la un laptop de pe rețeaua internă (intranet), acesta poate fi recunoscut și oprit cu ușurință.

Calitatea serviciilor este frecvent utilizată pentru a stabili prioritățile pentru aplicaţii sensibile la întârzieri, cum ar fi VoIP şi date cu misiune critică. Cu toate acestea, aceeaşi metodă care permite QoS să facă distincţia între prioritate înaltă şi trafic normal poate fi folosită de asemenea pentru a distinge traficul normal de traficul de tip necunoscut sau de altă origine. De asemenea QoS se poate utiliza în combinaţie cu mecanisme de politici de reţea pentru a limita cantitatea de trafic legitim.

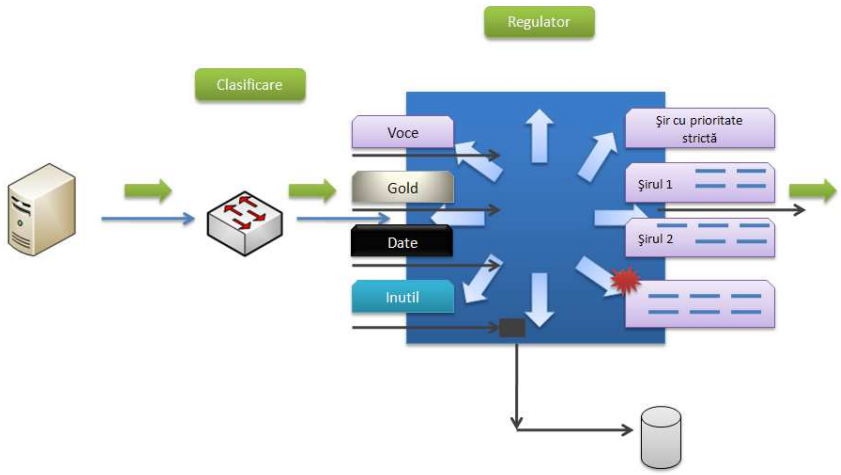

Fig.1 Reprezentarea grafică a mecanismului QoS

2. Tehnici pentru obţinerea unei bune calităţi a serviciilor

Nici o tehnică nu furnizează într-un mod optim o calitate a serviciilor eficientă şi pe care să te poţi baza. În schimb, au fost dezvoltate o varietate de tehnici, cu soluţii practice care adeseori combină mai multe tehnici:

2.1 Supraaprovizionarea

O soluţie uşoară este aceea de a furniza ruterului suficientă capacitate, spaţiu tampon şi lăţime de bandă astfel încât pachetele să zboare pur şi simplu. Problema cu această soluţie este aceea că este costisitoare. Odată cu trecerea timpului şi cu faptul că proiectanţii au o idee mai clară despre cât de mult înseamnă suficient, această tehnică ar putea deveni chiar realistă. Până la un punct: sistemul telefonic este supraapovizionat. Se întâmplă rar să ridici receptorul telefonului şi să nu ai ton de formare instantaneu. Pur şi simplu există atâta capacitate disponibilă încât cererea va fi întotdeauna satisfăcută.

2.2. Memorarea temporară

Fluxurile pot fi reţinute în zone tampon ale receptorului înainte de a fi livrate. Aceasta nu afectează fiabilitatea sau lăţimea de bandă şi măreşte întârzierea, dar uniformizează fluctuaţia. Pentru audio şi video la cerere, fluctuaţiile reprezintă problema principală, iar această tehnică este de mare ajutor. Exemplificare cu un flux de pachete care sunt livrate cu o fluctuaţie considerabilă:

- Pachetul 1 este transmis de către server la momentul *t = 0* sec şi ajunge la client la *t = 1* sec.
- Pachetul 2 acumulează o întârziere mai mare şi îi sunt necesare 2 sec pentru a ajunge. Pe măsură ce pachetele sosesc, ele sunt reţinute în zona tampon pe maşina client.

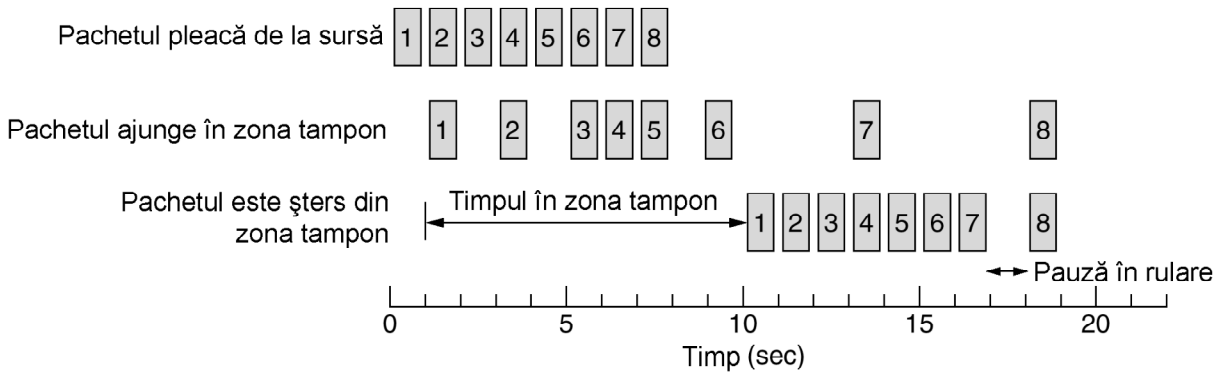

Fig. 2 Uniformizarea fluxului de ieşire prin memorarea temporară a pachetelor.

La *t = 10* sec începe rularea. La acest moment, pachetele de la 1 la 6 au fost introduse în zona tampon aşa că pot fi scoase de acolo la intervale egale pentru o rulare uniformă. Din păcate, pachetul 8 a avut o întârziere aşa de mare încât nu este disponibil când îi vine timpul de rulare,

aşa că rularea trebuie întreruptă până la sosirea acestuia, creând astfel o pauză supărătoare în muzică sau în film. Această problemă poate fi alinată prin întârzierea şi mai mare a momentului începerii, deşi acest lucru necesită şi un spaţiu tampon mai mare.

2.3. Formarea traficului

Neuniformitatea ieşirii este un lucru obişnuit dacă server-ul se ocupă de mai multe fluxuri la un moment dat şi de asemenea permite şi alte acţiuni cum ar fi derularea rapidă înainte sau înapoi, autentificarea utilizatorilor şi aşa mai departe. Totodată memorarea temporară nu este întotdeauna posibilă, de exemplu în cazul videoconferinţelor. Cu toate acestea, dacă s-ar putea face ceva pentru ca serverul (şi calculatoarele gazdă în general) să transmită cu rate de transfer uniforme, calitatea serviciilor ar fi mai bună.

Formarea traficului se ocupă cu uniformizarea ratei medii de transmisie a datelor (atenuarea rafalelor). În contrast, protocoalele cu fereastră glisantă pe limitează volumul de date în tranzit la un moment dat şi nu rata la care sunt transmise acestea. La momentul stabilirii unei conexiuni, utilizatorul şi subreţeaua (clientul şi furnizorul) stabilesc un anumit model al traficului (formă) pentru acel circuit. În unele cazuri aceasta se numeşte înţelegere la nivelul serviciilor (service level agreement). Atât timp cât clientul îşi respectă partea sa de contract şi trimite pachete conform înţelegerii încheiate, furnizorul promite livrarea lor în timp util. Formarea traficului reduce congestia şi ajută furnizorul să-şi ţină promisiunea. Astfel de înţelegeri nu sunt foarte importante pentru transferul de fişiere, însă sunt deosebit de importante pentru datele în timp real, cum ar fi conexiunile audio sau video, care au cerinţe stringente de calitate a serviciilor. Pentru formarea traficului clientul spune furnizorului: „Modelul meu de transmisie arată cam aşa. Poţi să te descurci cu el?” Dacă furnizorul este de acord, problema care apare este cum poate spune furnizorul dacă clientul respectă înţelegerea şi ce să facă dacă nu o respectă.

Monitorizarea fluxului traficului se numeşte supravegherea traficului (traffic policing.) Stabilirea unei forme a traficului şi urmărirea respectării ei se fac mai uşor în cazul subreţelelor bazate pe circuite virtuale decât în cazul subreţelelor bazate pe datagrame. Cu toate acestea, chiar şi în cazul subreţelelor bazate pe datagrame, aceleaşi idei pot fi aplicate la conexiunile nivelului transport.

3. Mecanisme de asigurare a calitatii serviciilor DiffServ.

IntServ era un model care permitea garantarea QoS, dar care necesita **rezervare de resurse pe flux individual** în **fiecare nod** al reţelei de-a lungul întregii căi de la echipamentul terminal sursă până la echipamentul terminal destinație. Spunem deci ca IntServ asigură **End to End QoS.**

DiffServ a apărut aşadar ca o reacție la complexitatea arhitecturii IntServ.

În arhitectura DiffServ:

- Procesările şi deciziile complexe sunt făcute la granița unei rețele de tip DiffServ, în nodurile de margine;

- Controlul QoS este asigurat prin folosirea unui număr redus de clase de îndrumare în nodurile de tip Core (nodurile interne ale domeniului DiffServ).

În contrast cu mecanismul IntServ, controlul QoS prin mecanismul DiffServ se realizează pringruparea pachetelor într-un numar mic de fluxuri aggregate în funcție de tipul serviciului și de destinație. Aceste fluxuri aggregate vor fi clasificate în clasa de îndrumare care vor avea asociat un cod DiffServ dumit DSCP (DS Code Point).

DiffServ se foloseşte de DSCP pentru a decide tratamentul care se aplică unui pachet în fiecare nod al rețelei. Înainte ca un pachet să intre într-un domeniu DS, acesta este marcat prin modificarea campului DSCP de către primul router, după calitatea serviciului dorită și după drepturile acestuia. În interiorul domeniul DS fiecare router va analiza acest camp înainte de a dirija pachetul. Astfel nu mai este necesară realizarea operațiilor de revernare, clasificare şi memorare a informațiilor despre fiecare flux, în fiecare nod al rețelei.

Un modul de condiționare de trafic poate conține următoarele elemente: meter, remarker, shaper dropper.

Meter este modulul de măsurare de trafic care are ca scop să determine dacă traficul este in profile sau out of profile.

Un exemplu simplu de meter este Token Bucket Meter, acesta folosind principiul galeții cu jetoane.

3.1. Algoritmul găleții găurite

Să ne imaginăm o găleată cu un mic orificiu în fundul său. Nu contează cu ce rată curge apa în găleată, fluxul de ieşire va fi la o rată constantă, ρ, dacă există apă în găleată şi zero dacă găleata este goală. De asemenea, odată ce găleata s-a umplut, orice cantitate suplimentară de apă se va revărsa în afara pereţilor şi va fi pierdută (adică nu se va regăsi în fluxul de ieşire de sub orificiu).

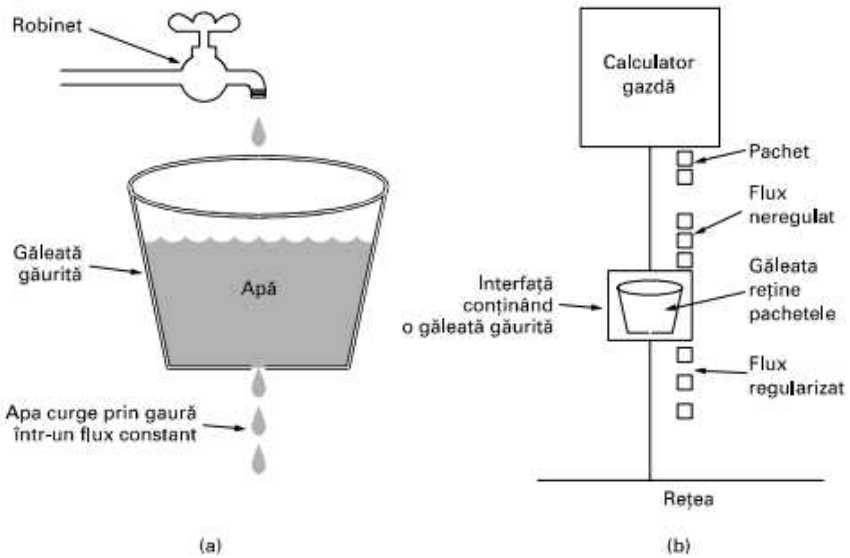

Fig.2 (a) O găleata umplută cu apă (b) O găletă umplută cu pachete

Conceptual, fiecare calculator gazdă este conectat la reţea printr-o interfaţă conţinând o găleată găurită, în fapt o coadă internă cu capacitate finită. Dacă un pachet soseşte în coadă atunci când aceasta este plină, el este distrus.

Cu alte cuvinte, dacă unul sau mai multe procese de pe calculatorul gazdă încearcă trimiterea unui pachet atunci când coada conţine deja numărul maxim de pachete, pachetele noi vor fi distruse fără menajamente. Acest aranjament poate fi implementat în interfaţa hardware sau poate fi simulat de către sistemul de operare gazdă. A fost propus pentru prima dată de către Turner (1986) şi este numit algoritmul găleţii găurite (the leaky bucket algorithm). De fapt nu este altceva decât un sistem de cozi cu un singur server şi cu timp de servire constant.

Calculatorul gazdă poate pune în reţea câte un pachet la fiecare tact al ceasului. Şi acest lucru poate fi realizat de placa de interfaţă sau de către sistemul de operare. Acest mecanism transformă un flux neregulat de pachete de la procesele de pe calculatorul gazdă într-un flux uniform de pachete care se depun pe reţea, netezind rafalele şi reducând mult şansele de producere a congestiei.

Dacă toate pachetele au aceeaşi dimensiune (de exemplu celule ATM), algoritmul poate fi folosit exact aşa cum a fost descris. Dacă se folosesc pachete de lungimi variabile, este adesea mai convenabil să se transmită un anumit număr de octeţi la fiecare tact şi nu un singur pachet. Astfel, dacă regula este 1024 octeţi la fiecare tact, atunci se pot transmite un pachet de 1024 octeţi, două de 512 octeţi, sau patru de 256 octeţi ş.a.m.d.

Dacă numărul rezidual de octeţi este prea mic, următorul pachet va trebui să aştepte următorul tact. Implementarea algoritmului iniţial al găleţii găurite este simplă. Găleata găurită constă de fapt dintr-o coadă finită. Dacă la sosirea unui pachet este loc în coadă, el este adăugat la sfârşitul cozii, în caz contrar, este distrus. La fiecare tact se trimite un pachet din coadă (bineînţeles dacă aceasta nu este vidă). Algoritmul găleţii folosind contorizarea octeţilor este implementat aproximativ în aceeaşi manieră. La fiecare tact un contor este iniţializat la n. Dacă primul pachet din coadă are mai puţini octeţi decât valoarea curentă a contorului, el este transmis şi contorul este decrementat cu numărul corespunzător de octeţi. Mai pot fi transmise şi alte pachete adiţionale, atât timp cât contorul este suficient de mare. Dacă contorul scade sub lungimea primului pachet din coadă, atunci transmisia încetează până la următorul tact, când contorul este reiniţializat şi fluxul poate continua.

Ca un exemplu de găleată găurită, să ne imaginăm un calculator care produce date cu rata de 25 milioane octeţi/sec (200 Mbps) şi o reţea care funcţionează la aceeaşi viteză. Cu toate acestea, ruterele pot să gestioneze datele la această rată doar pentru un timp foarte scurt (practic, până când li se umple spaţiul tampon). Pentru intervale mai mari ele lucrează optim pentru rate care nu depă- şesc valoarea de 2 milioane octeţi/sec. Să presupunem acum că datele vin în rafale de câte 1 milion de octeţi, câte o rafală de 40 msec în fiecare secundă. Pentru a reduce rata medie la 2 MB/sec, putem folosi o găleată găurită având ρ = 2 MB/sec şi o capacitate, C, de 1 MB. Aceasta înseamnă că rafalele de până la 1 MB pot fi gestionate fără pierderi de date şi că acele rafale sunt împrăştiate de-a lungul a 500 msec, indiferent cât de repede sosesc.

3.2. Algoritmul găleţii cu jetoane

Algoritmul găleţii găurite impune un model rigid al ieşirii, din punct de vedere al ratei medii, indiferent de cum arată traficul. Pentru numeroase aplicaţii este mai convenabil să se permită o creştere a vitezei de ieşire la apariţia unor rafale mari, astfel încât este necesar un algoritm mai flexibil, de preferat unul care nu pierde date. Un astfel de algoritm este algoritmul găleţii cu jeton (the token bucket algorithm).

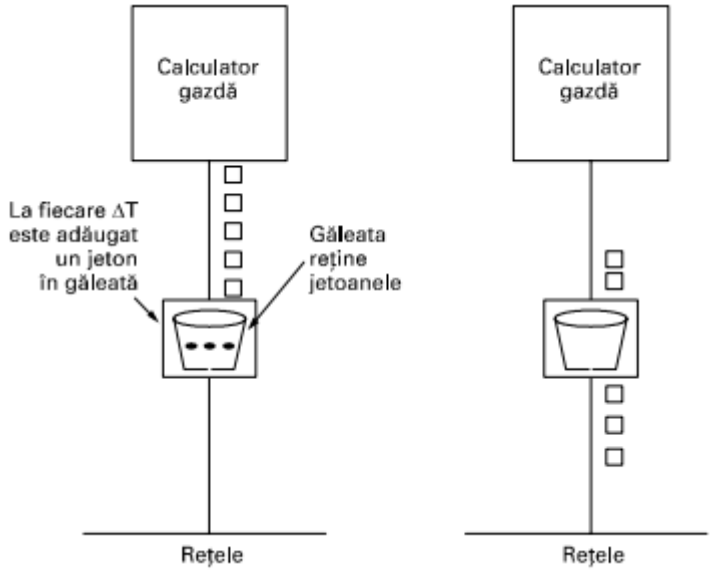

Fig.3 Principiul găleții cu jetoane pentru măsurarea traficului

- Jetoanele sunt adăugate în găleată cu o rată fixă X (token per second) și sunt îndepartate din găleată atunci când se transmite un pachet.

- O găleată are o adâncime finită si poate conține maxim Y jetoane. Când un pachet sosește și găsește cel puțin un jeton în găleată, atunci pachetul este considerat “in profile” și este transmis la ieșire, iar în același timp este îndepărtat un jeton.

- Dacă rata de sosire a pachetelor este R și este mai mică decât X, atunci găleata este umplută cu X-R jetoane pe secundă. Daca R=X, atunci numărul de jetoane rămâne constant. Atunci cand R este mai mare decât X, atât timp cât în găleata există cel puțin un jeton, pachetele sunt considerate a fi “in profile”. Dacă R depășește pentru mult timp X, jetonul se termină și pachetele pot fi remarcate sau aruncate.

- Această tehnică permite depășirea pe termen scurt a valorii de debit prin SLA – se permit rafale mici de pachete.

- Bineînțeles, implementarea pentru token bucket se realizează cu ajutorul unui contor, care de cele mai multe ori măsoară numarul de octeți și numarul de pachete, având în vedere faptul că pachetele pot avea dimensiuni diferite.

3.3. Servicii diferenţiate

Algoritmii bazaţi pe flux au potenţialul de a oferi o bună calitate a serviciilor unuia sau mai multor fluxuri deoarece acestea îşi rezervă resursele necesare de-a lungul rutei. Totuşi ei au şi un dezavantaj: au nevoie să stabilească în avans caracteristicile fiecărui flux, ceea ce nu este optim în cazul în care există milioane de fluxuri. De asemenea, ei memorează caracteristicile fiecărui flux ca date interne ale ruterului, ceea ce le face vulnerabile în cazul căderii ruterului.

Serviciile diferenţiate (DS) pot fi oferite de către un set de rutere care formează un domeniu administrativ (de exemplu un ISP sau telco). Administraţia defineşte un set de clase de servicii cu regulile de rutare corespunzătoare.

Dacă un client se înscrie pentru DS, pachetele clientului care intră în domeniu pot avea un câmp *Type of Service (Tipul serviciului)*, cu servicii mai bine furnizate unor clase (de exemplu serviciu premium) decât altora. Traficului dintr-o clasă i se poate cere să se conformeze unei anumite forme, cum ar fi găleata găurită cu o anumită rată decurgere. Un operator cu fler ar putea să ceară mai mult pentru fiecare pachet premium transportat sau ar putea să permit maximum N pachete premium pe lună pentru o taxă lunară suplimentară fixă. Această schemă nu presupune iniţializarea în avans, nici rezervarea resurselor şi nici negocierea capăt-la capăt pentru fiecare flux, care consumă timp, ca în cazul serviciilor integrate. Aceasta face ca serviciile diferenţiate să fie relativ uşor de implementat.

Pentru a evidenţia diferenţa dintre calitatea serviciilor bazate pe flux şi calitatea serviciilor bazate pe clase, se consideră un exemplu: telefonia Internet. În cazul organizării pe flux, fiecare apel telefonic are propriile resurse şi garanţii. În cazul organizării pe clase, toate apelurile telefonice beneficiază de resursele rezervate pentru clasa telefoniei. Aceste resurse nu pot fi preluate de pachete din clasa transferului de fişiere sau din alte clase, dar nici un apel telefonic nu beneficiază de resurse particulare, rezervate doar pentru acesta.

Serviciile diferențiale, reprezintă o arhintectura bazată pe marcarea pachetelor care permite să fie prioritizate în funcție de cerințele utilizatorului. În timpul unei congestii, ruterele renunță la pachetele cu prioritate scazută în favoarea celor cu prioritate maximă.

Modelul DiffServ este alcătuit din trei component principale:

- **Placa de rețea**: aceasta este specificată de inginerul de sistem şi defineşte nivelul clasei de serviciu pentru traficul care ar trebui să fie transportat într-o rețea;
- **Ruterul marginal al rețelei**: aceasta etichetează pachetele într-un cod în funcție de politica specific rețelei;
- **Ruterul central**: acesta examinează pachetele etichetate şi le transportă spre direcții predefinite;

3.3.1. Clasificarea pachetelor – ToS şi DSCP

Mecanismul DiffServ redefineşte câmpul ToS (Type of Sevice) din pachetul IP pentru a indica comportamentul de îndrumare PHB. Câmpul obținut este denumit DS Field şi conține:

- **DSCP (DiffServ Code Point)** asociat biților 0-5. Aceştia sunt utilizați pentru a indica PHB în vederea definirii modului de îndrumare al pachetului. Bitul 5 este bitul cel mai semnificativ al DSCP.
- **CU** – currently unused – biți 6-7 de rezervă.

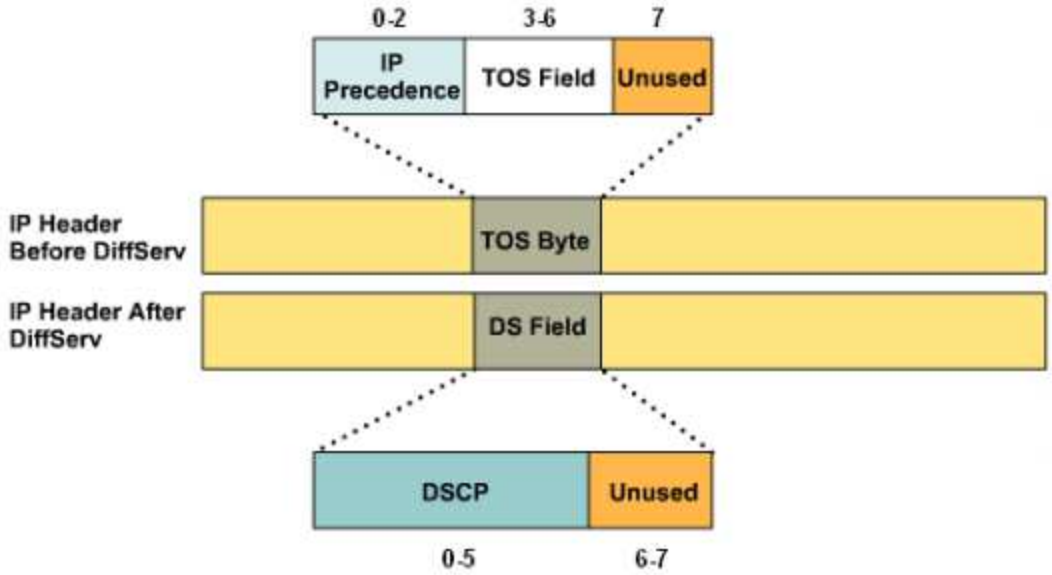

Fig.4 Câmpurile ToS şi DSCP

DS5	DS4	DS3	DS2	DS1	DS0	CU	CU

Fig.5 Câmpul DS din antentul IP

DS5, DS4, DS3 sunt folositi ca selector de clasa care permite compatibilitatea cu clasele definite în IntServ.

DS2, DS1, DS0 sunt folosiți pentru a diferenția şi a prioritize traficul din aceeaşi clasă luând în considerare probabilitatea de pierdere a pachetelor.

Alocarea codurilor s-a decis astfel:

- 32 coduri xxxxx0 – coduri folosite pentru acțiuni de standardizare;
- 16 coduri xxxx11 – coduri folosite experimental;
- 16 coduri xxxx01 – coduri pentru utilizare experimental, cu deosebirea că acestea pot intra în uz (pot fi standardizate) în cazul epuizării primelor 32.

DiffServ defineşte două tipuri de servicii:

- **Expedited Forwarding** – este un serviciu care impune fiecărui ruter de pe traseu să servească întotdeauna pachetele aparţinând acestui serviciu cu o rată foarte apropiată de cea cu care sosesc pachetele. Se asigură astfel un serviciu cu pierderi mici, latenţă scăzută şi bandă asigurată. Serviciul EF poate fi privit ca o linie închiriată, asigurând protecţia faţă de alţi utilizatori ai domeniul DS. Codul DSCP recomandat pentru serviciul EF este 101110;
- **Assured Forwarding** – este un serviciu care defineşte caracteristicile de bandă relativă şi de aruncare a pachetelor. AF defineşte 4 mari clase de îndrumare, iar în interiorul fiecărei clase se defines 3 priorităţi de aruncare (drop precedence).

Când pachetele dintr-o anumită clasă de îndrumare AF depaşesc un anumit prag specificat atunci pachetele cu cea mai mare prioritate în aruncare, vor fi primele aruncate. Nodurle reţelei permit trimiterea pachetelor "out of profile" numai atunci când există bandă în exces disponibilă.

AF asigură deci o flexibilitate mai mare şi o partajare dinamică a resurselor reţelei.

DSCP are structură de tipul *nnnmm0* unde *nnn* codifică clasa şi *mm* codifica prioritatea.

AF DSCP permite identificarea uneia din cele 4 clase, dar nu specifică o dimensiune maximă a cozilor sau intervalul de servire al planificatorului asociat cu fiecare coadă. Operatorul de reţea configurează aceşti parametri în funcţie de calitatea Edge-to-Egde dorită pentru fiecare clasă AF.

3.3.2. Retransmitere expeditivă (Expedited Forwarding)

În retransmiterea expeditiva sunt disponibile două clase de servicii: normale şi rapide (expeditive). Marea majoritate a traficului se presupune că este de tip normal, dar o mică parte se face şi expeditiv. Pachetele din această clasă ar trebui să poată traversa subreţeaua ca şi cum nu ar mai exista şi alte pachete. Cele două canale logice reprezintă o cale de a rezerva lăţime de bandă şi nu o a doua linie fizică.

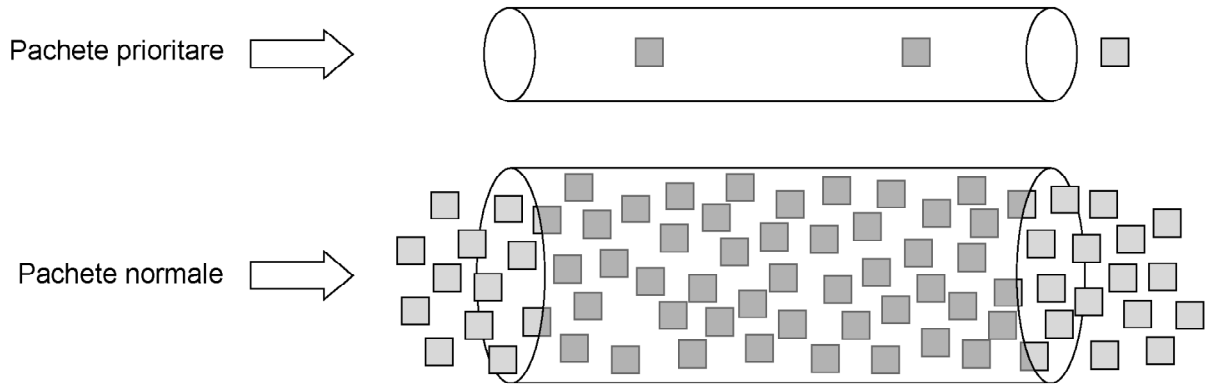

Fig. 6 Pachetele prioritare trec printr-o reţea cu trafic redus.

O modalitate de a implementa această strategie este de a programa ruterele astfel încât să aibădouă cozi de aşteptare pentru fiecare linie de ieşire, una pentru pachete prioritare şi cealaltă pentru pachete normale.

Când soseşte un pachet este introdus în coada corespunzătoare. Planificarea pachetelor ar trebui să utilizeze ceva de genul aşteptării echitabile ponderate. De exemplu, dacă 10% din trafic este de tip expeditiv şi 90% de tip normal, atunci 20% din lăţimea de bandă ar putea fi repartizată traficului expeditiv, iar restul traficului normal. În acest fel traficul expeditiv ar avea de două ori mai multă lăţime de bandă decât îi este necesar pentru a asigura întârziere mică.

Această repartizare poate fi obţinută transmiţând câte un pachet prioritar la fiecare patru pachete normale (presupunând că distribuţia dimensiunilor pachetelor ambelor clase este similară). În acest fel se speră că pachetele prioritare nu văd subreţeaua ca fiind aglomerată, chiar dacă există o încărcare substanţială.

3.3.3. Rutare garantată

O metodă mai elaborată de a administra clasele de servicii este **rutarea garantată (assured forwarding)**. Ea precizează că ar trebui să existe patru clase de priorităţi, fiecare cu propriile resurse. În plus defineşte trei probabilităţi de aruncare a pachetelor care sunt supuse congestiei: mică, medie şi mare. Luate împreună, aceste două criterii defines 12 clase de servicii.

Fig. 5 prezintă o modalitate în care pachetele ar putea fi procesate în cazul rutării garantate. Primul pas este acela de a repartiza pachetele într-una din cele patru clase de priorităţi. Acest pas se poate face pe calculatorul emiţător (după cum se vede şi din figură) sau în primul ruter. Avantajul realizării clasificării pe calculatorul gazdă care realizează transmisia este acela că are la dispoziţie mai multe informaţii despre fiecare pachet şi direcţia în care se îndreaptă.

Al doilea pas este de a marca pachetele în conformitate cu clasa din care fac parte. Pentru aceasta este necesar un câmp antet. Din fericire în antetul IP este disponibil un câmp pe 8 biţi numit *Tipul serviciului*, după cum vom vedea în continuare. Șase dintre aceşti biţi se vor folosi pentru clasa de serviciu, lăsând loc pentru codificarea claselor istorice şi a celor viitoare.

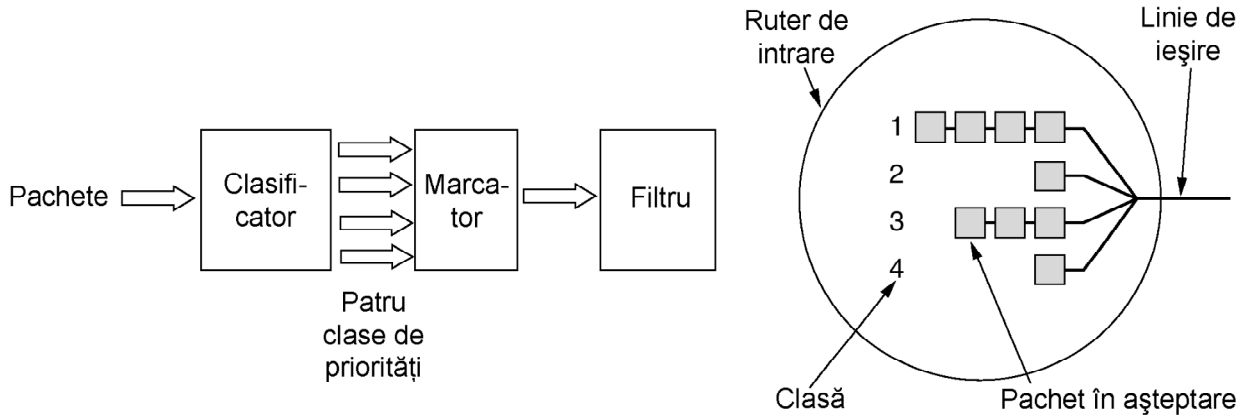

Fig. 7 O posibilă implementare a fluxului de date pentru rutarea garantată.

Pasul al treilea presupune trecerea pachetelor printr-un filtru (eng.: shaper/dropper filter) care ar putea întârzia sau arunca o parte dintre ele pentru a forma cele patru fluxuri într-o manieră acceptabilă, folosind de exemplu algoritmul găleţii găurite sau al găleţii cu jeton. Dacă sunt prea multe pachete, unele dintre ele ar putea fi aruncate în această etapă, în funcţie de categoria în care au fost plasate de filtru. Sunt posibile şi scheme mai elaborate care implică măsurători şi reacţii inverse

În acest exemplu, paşii specificaţi sunt efectuaţi pe calculatorul emiţător, deci fluxul rezultat este trimis primului ruter. Merită observat faptul că aceşti paşi pot fi efectuaţi de un software de reţea specializat sau chiar de către sistemul de operare, pentru a evita modificarea aplicaţiei existente.

4. Concluzii

Calitatea unui serviciu reprezintă aptitudinea unui software, a unei aplicaţii, a unui material, a unei reţele sau a unei interfeţe electronice sau informatice, de a satisface exigenţe implicite sau explicite. Pornind de la această observaţie se poate deduce o definiţie a managementului calităţii unui serviciu:

Managementul calităţii serviciilor reprezintă ansamblul proceselor, instrumentelor şi metodelor prin care se permite stabilirea unei politici şi a unor obiective destinate pentru a gira, ameliora şi garanta calitatea unui software, a unei aplicaţii, a unui material, a unei reţele sau a unei interfeţe electronice sau informatice.

Bibliografie

[1] Andrew S. Tanenbaum

[2] http://www.ietf.org/rfc.html

[3] www.wikipedia.ro

[4] http://www.ietf.org/rfc/rfc2998.txt

[5] Doherty, Jim. *Cisco Networking Simplified* (Jim Doherty, Neil Anderson, Paul Della Maggiora). ed. a II-a. ISBN 978-1-58720-199-8 (pbk.)

[6] Introduction to IP QoS, Cisco Systems, 2001

[7]RFC2474, Definition of the Differentiated Services Field

[8] RFC2475, An architecture for Differentiated Services