

**Universitatea Politehnica Bucuresti, Facultatea de Electronica, Telecomunicatii si  
Tehnologia informatiei**

# **Protocolul UDP**

**Profesor,  
Conf.dr.ing Stefan Stancescu**

**Bizon Cristian 442A  
Simion Bogdan 442A**

## Modelul OSI Bizon Cristian 442A

Modelul de OSI (Open Systems Interconnection) a aparut in anul 1984, fiind produs de ISO, care inseamna Internation Organization for Standardization. Acest model propune criterii generale pentru realizarea comunicatiei sistemelor de calcul pentru ca acestea sa poata schimba informatii, indiferent de particularitatile constructive ale acestora. Modelul de referinta OSI are aplicatii In toate domeniile comunicatiilor de date, nu doar In cazul retelelor de calculatoare.

Modelul OSI este o arhitectura ierarhica alcatuita din 7 nivele. Aceste nivele sunt:

**Aplicatie, Prezenta, Sesiune, Transport, Retea, Legaturi de Date, Fizic** si sunt numerotate de jos in sus, plecand de la 1, cum arata in figura 1<sup>1</sup>.



Figura 1

Fiecare nivel trebuie sa indeplineasca anumite functii:

- fiecare nivel trebuie sa fie diferit din punct de vedere al conceptului de abstracizare al transferului
- nivelele trebuie alese astfel incat functiile sa fie asemanatoare cu aplicaii si implementari reale
- fiecare nivel va exacuta functii bine definite
- intre niveluri traficul trebuie sa fie minim
- numarul de niveluri trebuie sa fie sufficient de mare pentru a nu solicita comansarea in vreun fel si sufficient de mic pentru a fi consistente.

Desi astazi sunt si alte sisteme, cei mai multi distribuitori de echipamente de comunicatie folosesc OSI pentru a educa utilizatorii In folosirea echipamentelor. Se considera ca OSI este cel mai bun mijloc prin care se poate face Inteles modul In care informatia este trimisa si primita.

## Nivelul Transport Bizon Cristian 442A

Nivelul transport are rolul de a face transferal informatiei intre 2 sisteme terminale care comunica. Furnizeaza controlul erorilor si controlul fluxului de date Intre doua puncte terminale, asigurand succesiunea corecta a datelor.

Nivelul de transport include atat modelul TCP/IP, care este fundamentul Internetului, cat si modelul OSI (Open Systems Interconnection) de retelistica generala. Definitiiile pentru nivelul transport sunt usor diferite In aceste modele.

LA nivelul de transport se afla mai multe servicii ce pot fi asigurate de un protocol de nivel transport, si protocoalele diferite le pot implementa sau nu:

- Orientarea Bitilor: este mai usor ca o aplicatie sa proceseze sirul de date ca o secventa de biti. Este folosita pentru lucrul cu formate de mesaje diferite.
- Comunicatie orientata pe conexiuni: este mai usor de manevrat decat modelele fara conexiune, cum ar fi modelul de datagrame al IP(Internet Protocol).
- Pentru ca pachetele sa ajunga in aceiasi ordine se numereaza segmentele, pentru ca apoi la destinatie sa fie aranjate in ordinea initiala. Cu toate ca nivelul de retea nu garanteaza ca pachetele de date vor ajunge In aceeasi ordine In care au fost trimise este un lucru dorit.
- Multiplexarea face parte din nivelul transport, pentru TCP/IP si din nivelul sesiune, pentru OSI.
- Exista posibilitatea ca pachetele sa se piarda In timp ce sunt transportate din cauza unor erori care ar putea sa apara sau datorita congestiei retelei. Protocolul de transport verifica acest lucru prin trimiterea de mesaje ACK sau NACK si prin calcularea CRC-ului.
- Evitarea congestiei: Controlul congestiei poate controla intrarea traficului Intr-o retea de telecomunicatii, pentru a evita prabusirea datorata congestiei prin Incercarea de a evita suprascrierea de procese sau capacitatile link-urilor nodurilor intermediare si retele si realizarea de pasi de reducere a resurselor, cum ar fi rata de reducere de trimitere a pachetelor. De exemplu, cererea de repetare automata poate pastra reseaua Intr-o stare de congestie; aceasta situatie poate fi evitata prin adaugarea evitarii congestiei la controlul fluxului, inclusiv Inceputul lent. Acest lucru pastreaza consumarea largimii de banda la un nivel jos la Inceputul transmisiunii, sau dupa retransmiterea pachetelor.

Unitatea de masura: segmentul

Exemple:

- TCP(Transmission Control Protocol)
- UDP (User Datagram Protocol)
- SPX (Sequenced Packet Exchange)

Internetul are doua protocoale principale in nivelul de transport unul neorientat pe conexiune si unul orientat pe conexiune. Protocolul neorientat pe conexiune se numeste UDP, iar protocolul orientat pe conexiune se numeste TCP.

## Protocolul UDP(User Datagram Protocol) Bizon Cristian 442A

### Introducere in UDP<sup>2</sup>

Setul de protocoale Internet suporta un protocol de transport fara conexiune, UDP (User Protocol – Protocol cu Datagrame Utilizator). UDP ofera aplicatiilor o modalitate de a trimite datagrame IP incapsulate si de a le transmite fara a fi nevoie sa stabileasca o conexiune.

UDP transmite segmente constand intr-un antet de 8 octeti urmat de informatia utila. Antetul este prezentat in figura 2<sup>3</sup>. Cele doua porturi servesc la identificarea punctelor terminale ale masinilor sursa si destinatie. Aceasta atasare are loc atunci cand se foloseste o simpla procedura de nume sau ceva asemanator. Fara campurile portului, nivelul de transport nu ar sti ce sa faca cu pachetul. Cu ajutorul lor, segmentele se livreaza correct.

# Biti	16	16	16	16	16
	Port Sursa	Port Destinatie	Lungime	Checksum	Date

Figura 2

**Portul sursa** este in primul rand necesar atunci cand un raspuns trebuie transmis inapoi la sursa. Prin copierea campului port sursa din segmental care soseste in campul **port destinatie** al segmentului care pleaca, procesul ce trimite raspunsul specificace proces de pe masina de trimitere urmeaza sa-l primeasca.

<sup>2</sup> [ Citate din "Andrew S.Tanenbaum – Retele De Calculatoare 4th Edition" ]

<sup>3</sup> Figura 2 – www.ccna.com

Campul **lungime** include antetul de 8 octeti si datele. Campul suma de control este optional si este stocat ca 0(zero) daca nu este calculate. Dezactivarea acestuia este o prostie, exceptie facand cazul in care calitatea informatiei chiar nu conteaza(de exemplu, transmisia vocala digitalizata).

**Checksum** este reprezentat de un camp de 16 biti care are rolul de a oferi un control al erorii prin compararea valorii checksum-ului cu numarul de biti primiti de statia destinatie.

Campul **data** reprezinta informatia in sine care trebuie transmisa.

Dezavantaje UDP:

- Nu realizeaza controlul fluxului
- Controlul erorii
- Retransmiterea unui segment incorrect primit.

Toate acestea depind de procesele utilizatorului. Ceea ce face este sa ofere protocolului IP o interfata cu facilitati adaugate de demultiplexare a mai multor procese, folosind porturi. Aceasta este tot ceea ce face UDP-ul. Pentru aplicatiile care trebuie sa aiba un control precis asupra fluxului de pachete, controlului erorii sau cronometrarea, UDP-ul furnizeaza doar ceea ce "a ordonat doctorul".

Un domeniu unde UDP-ul este in mod special util este acela al situatiilor client-server. Deseori, clientul trimite o cerinta scurta serverului si asteapta inapoi un raspuns scurt. Daca se pierde ori cererea ori raspunsul, clientul poate pur si simplu sa incerce din nou dupa ce a expirat timpul. Nu numai ca va fi mai simplu codul, dar sunt necesare si mai putine mesaje (cate unul in fiecare directie) decat la un protocol care solicita o initializare initiala.

O aplicatie care foloseste UDP-ul in acest fel este DNS( Domain Name System. Pe scurt, un program care trebuie sa caute adresele de IP ale unor nume gazda, de exemplu www.google.ro , poate trimite un pachet UDP, continand numele gazda, catre un server DNS. Serverul raspunde cu un pachet UDP continand adresa de IP a gazdei. Nu este necesara nici o initializare in avans si nici o inchidere de sesiune. Doar doua mesaje traverseaza reseaua.

## **RPC (Remote Procedure Call) Bizon Cristian 442A**

RPC este o tehnica puternica pentru construirea aplicatiilor distribuite bazate pe modelul client-server. Modelul extinde notiunea de apel local de procedura, diferenta fiind ca

procedura apelata nu se afla in acelasi spatiu de adresare cu procedura apelanta. Cele doua procese implicate pot sa fie pe acelasi calculator sau pot sa fie pe doua calculatoare in retea. Utilizand PRC, programatorii de aplicatii distribuite ocolesc dezvoltarea interfatarii aplicatiei cu retea.

Protocolul RPC se afla la nivelul Prezentare din stiva OSI.

RPC foloseste protocolul XDR ( eXternal Data Representation) – RFC 1832- pentru codarea datelor. Acest protocol se afla tot la nivelul Prezentare din stiva OSI

Intr-un anume sens, trimiterea unui mesaj catre o statie la distanta si primirea inapoi a unui raspuns seaman mult cu realizarea unei functii de apel intr-un limbaj de programare. In ambele cazuri se incepe cu unul sau mai multi parametri si se primeste inapoi un rezultat. De exemplu, imaginati-va doar procedura numita `get_IP_address(host_name)` care functioneaza prin trimiterea unui pachet UDP catre un server DNS si asteptarea raspunsului, cronometrind si incercand inca o data, daca raspunsul nu apare sufficient de rapid.

<sup>4</sup>Efortul cel mai important in acest domeniu a fost depus de catre Birell si Nelson (1984). Rezumand, ce au sugerat Birell si Nelson a fost sa permita programelor sa apeleze procedure localizate pe statii aflate la distanta. Cand procesul de pe masina 1 invoca o procedura de pe masina , procesul apelant de pe prima masina este suspendat si executia procedurii invocate are loc pe cea de-a doua. Informatia poate fi transportata de la cel care apeleaza la cel care este apelat in parametric si se poate intoace in rezultatul procedurii. Nici un transfer de mesaje nu este vizibil pentru programator. Tehnica este cunoscuta sub numele de RPC(Remote Procedure Call) si a devenit baza pentru multe aplicatii de retea. In mod traditional, procedura care apeleaza este cunoscuta ca fiind clientul si procedura apelata ca fiind serverul si vom folosi aceste denumiri si aici.

Ideea din spatele RPC-ului este aceea de a face un apel de procedura la distanta sa arate pe cat posibil ca un local. In forma cea mai simpla, pentru apelarea unei procedure la distanta, programul client trebui sa fie legat cu o mica procedura de biblioteca, numita client stub, care reprezinta procedura serverului in spatial de adresa al clientului. In mod similar, serverul este legat cu o procedura numita server stub. Aceste procedure ascund faptul ca apelul de la client la server nu este local.

---

<sup>4</sup> [ Citate din "Andrew S.Tanenbaum – Retele De Calculatoare 4th Edition" ]

Pasii efectivi ai realizarii unui RPC sunt prezentati in figura 3<sup>5</sup>. Pasul 1 este cel in care clientul apeleaza stub-ul client. Acest apel este un apel de procedura locala, cu parametrii introdusi in stiva in modul obisnuit. Pasul 2 consta in impachetarea parametrului de catre stub-ul client intr-un mesaj si realizarea unui apel de system pentru a trimite mesajul. Impachetarea parametrilor este denumita marshaling. Pasul 3 consta in faptul ca nucleul sistemului de operare trimite un mesaj de la masina client la masina server. Pasul 4 consta in trimiterea de catre nucleu a pachetelor care sosesc la stub-ul server. In sfarsit, pasul 5 consta in faptul ca stub-ul server apeleaza procedura server cu parametrii despachetati. Raspunsul urmeaza aceeași cale și în cealaltă direcție.

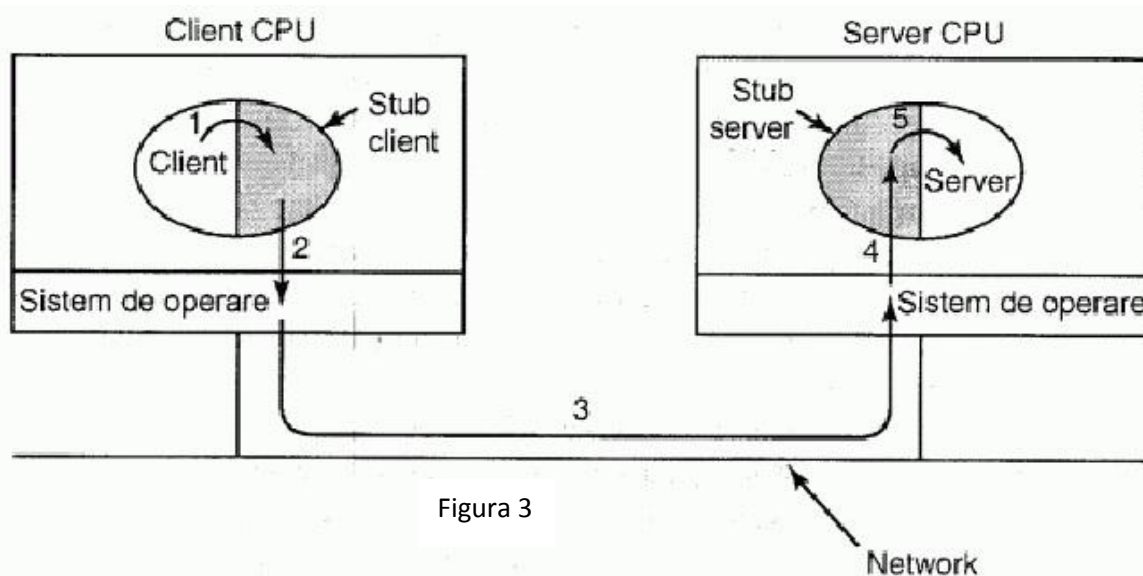


Figura 3

RPC furnizeaza un punct central pentru aplicatiile server sa obtina porturi TCP/UDP pe masina respectiva pe care ruleaza. Aplicatiile nu trebuie « legate » pe anumite porturi specificate pentru ca serviciul de port mapping al RPC-ului furnizeaza porturi libere mai mari de 1024 serverelor RPC. Serviciul de lookup folosit pentru asa ceva include PORTMAPPER-UL (PMAP) si RPCBIND care sunt descrise in RFC 1833. Portmapper are un port fix pe care sta deschis (111) fie TCP, fie UDP. Acest serviciu de lookup trebuie deschis inaintea serverului/clientului si trebuie sa ramana functional pe toata durata executiei aplicatiei RPC.

O problema este introdusa de acest mecanism. Serverele RPC nu utilizeaza porturi rezervate (asa cum sunt cele specificate pentru servicii in fisierul */etc/services*); atunci cand pornesc, utilizeaza portul disponibil dat de serviciul portmapper. Atunci cand un program client doreste sa apeleze un anumit serviciu RPC, el nu stie pe ce port ruleaza acesta pentru a face apelul. Trebuie sa existe o metoda de aflare a acestui port.

<sup>5</sup> Figura 3- Andrew S.Tanenbaum – Retele De Calculatoare 4th Edition

Deamonul portmapper rezolva aceasta problema. Atunci cand un client face un apel RPC, mai intai apeleaza serviciul de portmapper pentru aflarea portului serverului.

## NetBIOS

Network Basic Input/Output System, sau NetBIOS este un API folosit de aplicatii in retelele IBM pentru a solicita servicii de la procesele de retea de nivel scazut. NetBIOS include 3 servicii de baza:

- Serviciul de nume, acest serviciu este in realitate responsabil pentru asigurarea schemei de nume folosita de NetBEUI. Numele sunt difuzate la intrarea in retea sau pot fi interogate explicit prin pachete de difuzare
- Serviciul de sesiuni este un serviciu orientat conexiune, asigurand secventierea mesajelor si gestionind confirmarile pentru acestea.
- Serviciul de datagrame este bazat pe datagrame si este folosit pentru difuzari si mesaje de informare.

## UDP-Lite

O clasa speciala de aplicatii pot beneficia daca date partial deteriorate sunt livrate, decat sa fie aruncate, atunci cand se utilizeaza cai predispuse la eroare. Astfel de aplicatii pot tolera coruptia datelor si pot sa aleaga sa utilizeze protocolul Lightweight User Datagram(UDP-Lite), in loc de UDP de baza.

Formatul antetului este aproape identic cu cel al UDP-ului. UDP-Lite schimba semantica domeniului UDP "lungimea datelor de transport", cea a unui domeniu "lungimii checksum". Altfel, UDP-Lite este identic cu UDP.

Interfata de UDP-Lite difera de cea a UDP prin adaugarea unui singur (socket), optiune care comunica valoarea lungimi checksum.

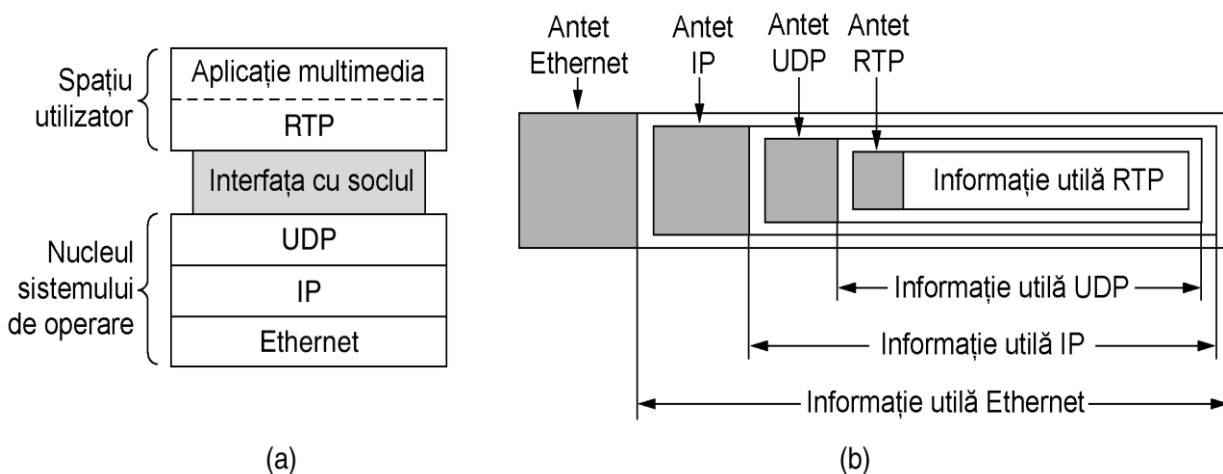
In mod implicit, acoperirea checksum la UDP-Lite se extinde de-a lungul Intregii datagrame. Daca este necesar, o aplicatie poate modifica dinamic aceasta valoare, de exemplu, pentru a oferi o protectie mai mare pentru unele mesaje. UDP-Lite Intotdeauna verifica faptul ca un pachet a fost livrat la destinatie, adica, verifica Intotdeauna campurile de antet. Erorile care apar in partea care nu este importanta nu vor provoca o datagrama UDP care urmeaza sa fie aruncate de destinatie.



Aplicatiile care folosesc UDP-Lite, prin urmare, nu trebuie sa faca presupuneri cu privire la corectitudinea datelor primite. Aplicatie de trimitere, prin urmare, trebuie sa specifice un checksum minim de acoperire, care include toate anteturile de protocol sensibile.

### Protocoale de transport in timp real Simion Bogdan 442A

Una dintre cele mai des intalnite aplicatii ale protocolului UDP este de RPC intre client si server, alta fiind aplicatiile multimedia in timp real cum ar fi video-conferintele, radiourile online, telefonica prin internet, streamurile video si podcasturile au crescut in popularitate odata cu cresterea accesibilitatii la internet a populatiei. Toate aceste aplicatii folosesc in principiu acelasi protocol de transport si acesta este RTP (Real-time Transfer Protocol). Acesta este foarte intalnit in aplicatiile multimedia, el transportand informatia audio si video incapsulata in pachete, iar procesarea se face in mare procentaj la destinatar.



In figura de mai sus avem prezentata pozitionarea protocolului RTP in stiva de protocoale(a) si incapsularea pachetului de date (b). Pozitia RTP in stiva de protocoale este una atipica. El se afla in spatial utilizatorului deasupra protocolului UDP care se afla in nucleul sistemului de operare. O aplicatie multimedia in general lucreaza cu mai multe stream-uri video, audio, semnale, text si eventual altele; acestea fiind introduse ulterior in biblioteca RTP localizata in spatial utilizatorului la fel ca si aplicatia. Biblioteca are rolul de a

<sup>6</sup> Andrew S.Tanenbaum – Retele De Calculatoare 4th Edition

multiplexa stream-urile si de a le coda in pachete RTP, acestea fiind ulterior trimise catre un socket. In partea de sistem de operare a stivei se genereaza pachetele UDP in interiorul carora sunt incluse si cele RTP si sunt expediate catre stratul IP pentru a fi transmise prin Ethernet. Odata pachetul ajuns la receptor procesul are loc in sens invers. Aplicatia multimedia va primi informatia de la biblioteca RTP si va reda informatia media. Datorita acestui mod de proiectare a RTP acesta nu poate fi clasat intr-un anumit strat din stiva. Acesta ruleaza in spatial utilizatorului si are legatura cu aplicatia si aparent este un protocol ce apartine acestui strat dar este totusi un protocol generic care nu are legatura cu aplicatie, el oferind doar facilitate de transport, deci este in categoria protocoalelor de transport sau in alta exprimare RTP este un protocol de transport ce este implementat in stratul de aplicatie.

“Functia de baza a RTP este de a multiplexa mai multe stream-uri de timp real intr-un singur stream de pachete UDP. Acesta din urma poate fi trimis către o singura destinație (unicast) sau mai multe (multicasting). Pentru ca RTP foloseste UDP simplu, pachetele lui nu sunt tratate special de rutere, decat dacă niste caracteristici specifice de QoS (Quality Of Service) ale IP sunt activate. Nu există garantii special ale livrării, deci pachetele pot fi pierdute, corupte, intarziate etc.”<sup>7</sup>

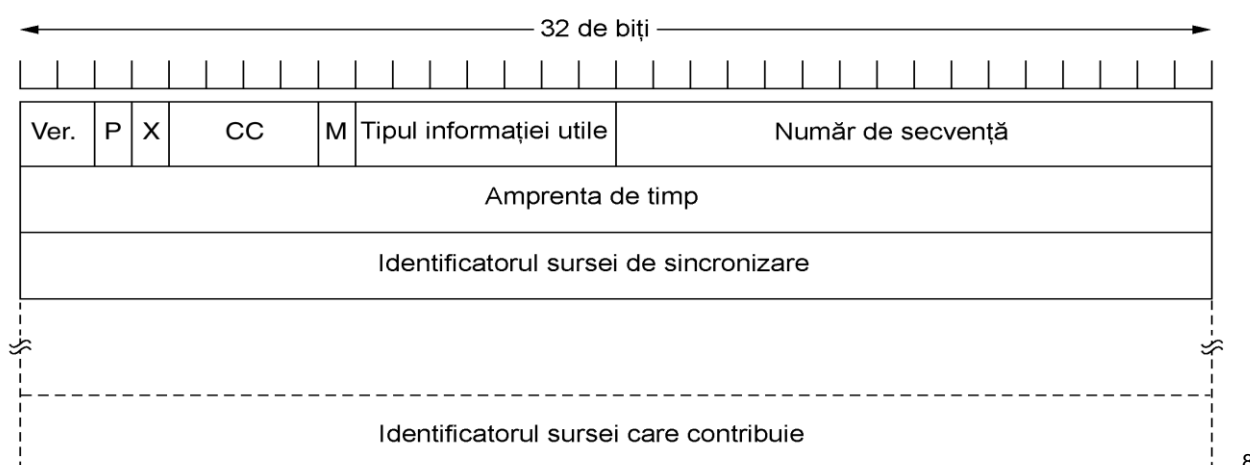
RTP are niste proprietati specific care ajuta dispozitivele receptoare la procesarea informatiei multimedia. In cazul fiecarui pachet ce a fost trimis cu un stream RTP i se da numar egal cu incrementul predecesolui sau. Prin aceasta etichetare masina destinatie determina daca lipsesc unu sau mai multe pachete de date si in acest caz aplicatie poate decide ce se intampla. Pentru un stream video va putea omite un cadru, iar daca informatia transmisa este una audio in locul pachetului lipsa va putea fi interpolate o valoare lipsa. Se fac aceste ajustari deoarece retransmisia nu este o optiune deoarece informatia oricum va ajunge prea tarziu pentru a fi utilizata, de aceea protocolul RTP nu are nici confirmare nici un mecanism de cerere de retransmisie asemenea protocolului UDP.

---

<sup>7</sup> Andrew S.Tanenbaum – Retele De Calculatoare 4th Edition

In interiorul unui pachet de tip RTP se pot gasi mai multe esantioane ce pot fi codate in o multime de feluri acest fapt fiind dependent de aplicatie dar pentru a fi mai usor lucrul emitor-receptor protocolul RTP defineste cateva profile si accepta mai multe formate de codare pentru fiecare dintre ele. Tipul de codare este specificat de catre sursa in campul header al RTP.

O alta facilitate ar putea fi timestamping sau marcarea timpului ceea ce inseamna ca sursa asociaza un marcaj de timp pentru primul esantion din fiecare pachet. Timestamp-urile se refera la inceputul streamului deci valorile lor absolute nu conteaza neaparat ci mult mai importante sunt diferentele intre ele. Cu ajutorul acestuo mecanism receptorul poate sa stocheze esantioanele intr-o memorie tampon(buffer) pentru a fi redate esantioanele la un interval egal de timp pentru a putea fi percepute correct fara a conta daca pachetul care trebui a sosit sau nu. Cu ajutorul timestamping-ului se reduc variatiile datorate intarzierilor din retea si permite si sincronizarea dintre mai multe stream-uri intre ele. Un bun exemplu ar fi televiziunea digitala care este compus din un stream video si doua audio, ori pentru sunet stereo ori unul pentru coloana sonora si celalalt pentru dublare. Fiecare stream are un dispozitiv fizic generator diferit dar datorita faptului ca sunt etichetate de catre acelasi contor ele sunt redate sincronizat chiar daca nu sunt trimise organizat.



<sup>8</sup> Andrew S.Tanenbaum – Retele De Calculatoare 4th Edition

Headerul protocolului RTP contine 3 cuvinte de 32 de biti dar poate avea si extensii. In primul cuvint se afla campul de versiune si momentan avem 2, bitul P arata daca acel pachet a fost bordat la multiplu de 4 octeti. Ultimul octet de bordare ne spune cati octeti au fost adaugati. Bitul X ne spune daca exista o extensie dar formatul si semnificatia ei nu sunt definite, doar lungimea in primul cuvint al extensiei. Campul CC semnaleaza cate surse ce contribuie exista si care pot fi ca numar intre 0 si 15. Bitul M este un marker specific al aplicatiei si poate fi folosit pentru a marca inceputul unui cadru video, inceputul unui semnal audio sau ceva specific aplicatiei. Campul Tipul informatiei utile contine tipul algoritmului de codare folosit. Acest camp este continut de fiecare pachet de aici rezultand ca se poate schimba tipul codarii pe parcursul transmisiei. Numarul de secveta contorizeaza prin incrementare si are scopul de a detecta pierderea de pachete. Rolul timestampului a fost descries anterior.

“Identificatorul sursei de sincronizare spune carui stream apartine acel pachet. Este folosit pentru a multiplexa și demultiplexa mai multe stream-uri de dare intr-un / dintr-un singur stream de pachete UDP. Identificatorii surselor contribuitoare, dacă există, sunt folositi candin studio-ul de inregistrare se folosesc mixere. În acest caz, mixer-ul este sursa de sincronizare, și stream-urile mixate sunt listate aici.”

## **RTCP Simion Bogdan 442A**

RTCP este un protocol care este legat de RTP. RTCP nu transporta date, ocupandu-se de interfata cu utilizatorul si de raspunsuri.

“RTCP se ocupa si de sincronizarea intre stream-uri. Diferite stream-uri pot folosi diferite surse de tact („ceasuri” diferite), cu granularitati si derivate diferite. RTCP poate sa le sincronizeze, in ciuda acestor impedimente.”<sup>9</sup>

Func tia de raspunsuri furnizeaza feedback catre surse in legatura cu diferite proprietati ale retelei, intre care congestia, latimea de banda si intarzierile. Pentru a creste

---

<sup>9</sup> Andrew S.Tanenbaum – Retele De Calculatoare 4th Edition

rata de date poate fi folosita informatia furnizata de acest serviciu in procesul de codare, de asemenea, cand reseaua functioneaza bine creste calitatea, sau rata de date se poate reduce cand exista probleme in retea.

In limitele performantelor curente ale retelei, se poate oferi cea mai buna calitate posibila adaptand algoritmi de codare ca urmare a primirii raspunsurilor continue. Se poate schimba codarea in timpul transmisiei, campul „tip de incarcatura” fiind folosit pentru a indica masinii destinatie ce algoritm de codare s-a folosit pentru pachetul curent.

Pentru o aplicatie cu un numar mare de calculatoare, multicast, pentru a preveni ocuparea bandei cu rapoarte, rata lor este micorata undeva sub 10% din banda media, latimea de banda folosita de RTCP crescand deoarece rapoartele se trimit catre toti participantii la comunicare. Aceasta este problema mecanismului de feedback. Ascultand porturile RTCP, fiecare participant poate determina numarul de participanti. Folosind acesta, poate calcula dimensiunea benzii cu ajutorul emitatorului.

## **Pachetele RTCP Simion Bogdan 442A**

Pachetele RTCP trebuie trimise intr-un pachet compus, existand mai multe tipuri de pachete. Dupa o sectiune fixa, necesara pentru a putea stivui pachetele, urmeaza elemente variabile ca lungime, nu mai mult de 32 de biti. Se pot „alipi” mai multe pachete RTCP pentru a forma acest pachet compus, care poate fi trimis intr-un pachet unic. Deoarece protocoalele din nivelele inferioare tin cont de lungimea totala si determina sfarsitul pachetului compus, nu se numara pachetele componente RTCP.

- SDES – acesta descrie obiectele sursa
- SR (Sender Report): continutul acestuia constand in statistici de transmisie
- RR (Receiver Report): contine statistici primite de la participantii care nu emit activ
- BYE: indica unde se termina participarea unei masini
- APP: pentru functii specifice aplicatiei

Fiecare pachet individual din pachetul compus poate fi separat procesat fara a fi in o anumita ordine. Pentru a putea fi folosit la eficienta maxima protocolul i se impun anumite constrangeri:

- Statisticile de la receptie trebuie trimise cat se poate de des pentru a se maximiza rezolutia statisticilor

- Participantii noi o data cu primirea datelor trebuie sa primeasca si CNAME-ul pentru o sursa pentru a putea sa identifice sursa si tipul unformatiei cat mai repede.

Drept urmare toate pachetele RTCP trebuie livrate intr-un pachet compus in care sunt incapsulate cel putin 2 pachete individuale

“Un participant RTP trebuie să trimită doar un singur pachet compus RTCP pe intervalul de raportare, pentru ca estimarile privind banda pentru fiecare participant să fie corecte. Exceptie: când pachetul compus este impartit pentru criptare partiala. Dacă sunt prea multe surse și nu incap toate pachetele RR intr-un singur ansamblu fără depasirea MTU (maximum transmission unit), atunci doar subsetul care incape intr-un MTU ar trebui inclus în fiecare interval. Subseturile trebuie selectate folosind metoda round-robin, astfel incat toate sursele să fie raportate.”<sup>10</sup>

v=2	p	Nr RR	tip de pachet	Lungime mesaj
SSRC al raportului emitator				
Timestamp NTP(2 cuvinte de 32 biti)				
Timestamp RTP				
Contor cumulativ de pachete al emitatorului				
Contor cumulativ de octeti al emitatorului				
Blocul 1 al raportului de receptie				
Blocul 2 al raportului de receptie				
.....				

11

Pachet RTCP ce include si headerul

<sup>10</sup> RFC3550

<sup>11</sup> <http://www.cs.odu.edu/~cs778/jeffay/Lecture6.pdf>

## Bibliografie

A. Tanenbaum, *Rețele de calculatoare* (editia a patra)

R. Stevens, B. Fenner, A. Rudoff, *UNIX Network Programming Volume 1, Third Edition: The Sockets Networking API*, Addison Wesley, 2003

S. Buraga, G. Ciobanu, *Atelier de programare In rețele de calculatoare*, Polirom, Iasi, 2001

Haugdahl, J. Scott (1990). Inside NetBIOS

D. Comer, D. Stevens, *Internetworking with TCP/IP: vol.III: Client-Server Programming and Applications*, Prentice Hall, New Jersey, 1993

Postel, J. (1980), RFC 768 : User Datagram Protocol; RFC 1591 (1992)

L-A. Larzon Lulea University of Technology, M. Degermark; S. Pink, The University of Arizona; L-E. Jonsson, Ed. Ericsson G. Fairhurst, Ed. University of Aberdeen (2004) RFC 3828

Sollins, K., MIT (1992): RFC 1350: TFTP

Eggert, L. , Nokia (2008); RFC 5405

Flick, J. , Hewlett-Packard Company (2005); RFC 4113

Deering, S. , Cisco (1998); RFC 2460

## Cuprins

Modelul de referinta OSI.....	1
Nivelul transport.....	1
Protocolul UDP.....	3
RPC.....	4
NetBIOS.....	7
UDP-Lite.....	7
RTP.....	9
RTCP.....	12
Pachete RTCP.....	13
Bibliografie.....	15