UNIVERSITATEA "POLITEHNICA" TIMIŞOARA
FACULTATEA DE AUTOMATICĂ ŞI CALCULATOARE

LUCRARE DE DIPLOMĂ

Coordonator ştiinţific,
Prof. Mircea VLĂDUŢIU

Candidat,
Mihai RUSOAIE

TIMIŞOARA
Iunie 2006

UNIVERSITATEA "POLITEHNICA" TIMIŞOARA
FACULTATEA DE AUTOMATICĂ ŞI CALCULATOARE

STUDIU EXPERIMENTAL PENTRU CALITATEA SERVICIILOR (QUALITY OF SERVICE) ÎN TEHNOLOGIA VOICE OVER IP

– LUCRARE DE DIPLOMĂ –

Coordonator ştiinţific,
Prof. Mircea VLĂDUŢIU

Candidat,
Mihai RUSOAIE

TIMIŞOARA
Iunie 2006

CUPRINS

Capitolul 1

INTRODUCERE

1.1. Generalități

Voice over IP, sau telefonia pe Internet, este o tehnologie care permite transmiterea vocii după un anumit standard, peste protocolul IP (Internet Protocol). Astfel se nasc rețele de telefonie combinate între standardul PSTN[1] şi VoIP. Un abonat VoIP poate apela alt abonat VoIP, iar dacă are un abonament ce îi permite acest lucru, poate apela un număr de telefon clasic dintr-o rețea de telefonie fixă sau mobilă.

În cazul VoIP, apelantul poate folosi un calculator obişnuit cu un software specializat şi o pereche de căşti cu microfon (headset), sau poate dispune de un aparat hardware specializat pentru VoIP.

Vocea se codează/decodează cu ajutorul unui codec[2] şi se transmite prin legătura de date (protocolul IP) existentă între cele două părți.

---

[1] **PSTN** – *Public Switched Telephone Network* este rețeaua publică globală de telefonie bazată pe comutație de circuit, spre deosebire de Internet care este rețeaua publică globală bazată pe comutație de pachete. La origini, PSTN a fost o rețea de telefonie analogică, în ziua de azi ea fiind aproape în întregime digitală, incluzând atât telefonia fixă, cât şi pe cea mobilă.

[2] **codec** – dispozitiv hardware sau program software capabil să efectueze codarea şi decodarea unui flux de date sau a unui semnal. Cuvântul codec provine din prescurtarea şi alăturarea cuvintelor **Co**mpressor–**Dec**ompressor, **Co**der–**Dec**oder sau algoritm de **Co**mpresie/**Dec**ompresie.

1.2. Avantajele şi dezavantaje VoIP faţă de telefonia clasică

Referindu-ne la costuri, telefonia clasică necesită resurse financiare mai mari deoarece este nevoie de o reţea dedicată, centrale de comutaţie, adaptoare între circuite şi alte astfel de echipamente, pe când VoIP foloseşte, în cele mai multe cazuri, reţeaua publică de date (Internet), ajutată de câteva echipamente specifice (routere, telefoane speciale).

Din punct de vedere al funcţionalităţii, VoIP are, din nou, un avantaj major: flexibilitatea în adăugarea unor servicii adiţionale (agendă telefonică, transmisii video, conferinţe, transmisii simultane de date), precum şi uşurinţa cu care se poate reloca locaţia abonaţilor, avantaj evident pentru furnizorii de astfel de servicii.

În ceea ce priveşte mobilitatea, un abonat VoIP poate să intre în reţea de pe orice terminal conectat la Internet care îndeplineşte câteva condiţii elementare (lăţime de bandă minim garantată, timpi de răspuns cât mai mici).

Ca prim dezavantaj, şi cel mai important, al VoIP, trebuie să amintim de standardele QoS[1], care, în cazul VoIP fiind aproape imposibil de implementat, deoarece conexiunile de date traversează reţele publice, la care nu se poate impune un QoS de către furnizorul final de servicii. În cazul telefoniei clasice, furnizorii folosesc reţelele proprii şi astfel pot asigura o calitate a vocii, ecoului, pierderilor şi stabilităţii conexiunii. Alte dezavantaje ar fi: încetarea funcţionării serviciilor în cazul unei pane de curent, integrarea aproape imposibilă a serviciilor de urgenţă (112 sau 911). Pe lângă acestea, majoritatea serviciilor VoIP nu includ criptare.

[1] **QoS** – *Quality of Service* (calitatea serviciului) în reţelele cu comutaţie orientată pe pachete se referă la faptul că reţeaua de telecomunicaţii dată trebuie să se încadreze în anumiţi parametrii de calitate (în cele mai multe cazuri este vorba ca un pachet să reuşească să traverseze reţeaua între două puncte ale sale)

Capitolul 2

TEHNOLOGIE

2.1. Protocoale şi codec-uri

Există multe dezbateri în jurul celor mai populare tipuri de VoIP: SIP şi H.323. Inițial, H.323 a fost cel mai popular protocol, dar popularitatea sa a scăzut datorită adaptabilității sale scăzute la NAT[1] şi firewall-uri. Din acest motiv, în dezvoltarea de soluții end-user[2] VoIP a fost folosit tot mai mult SIP[3]. Cu toate acestea, în rețele de voce de mare capacitate, unde totul trebuie să fie sub control, se alege H.323, în cele mai multe cazuri. Astfel, într-o convorbire, cu toate că clientul final foloseşte SIP pentru a accesa rețeaua VoIP, aceasta este convertită pe tronsoanele mari de comunicații în H.323. Cu toate acestea, recent, s-au adus câteva noi modificări la protocolul H.323, astfel ca acesta să poată traversa cu uşurință NAT şi echipamente firewall, deschizând noi posibilități pentru acest protocol.

În zilele noastre, multe dintre convorbirile internaționale sunt rutate prin VoIP, fără ca utilizatorul final să realizeze. Acest lucru se poate observa şi prin ieftinirea semnificativă, din ultimii ani, a tarifelor operatorilor la convorbirile internaționale.

Semnalizarea într-o rețea VoIP, ca şi în rețelele clasice de telefonie, se face prin intermediul unuia din următoarele protocoale: SIP, H.323, Megaco (H.248), Skinny

---

[1] **NAT** – *Network Address Translation* implică rescrierea sursei şi/sau a destinației pachetelor într-o rețea IP, când se face trecerea printr-un router sau firewall.

[2] **End-user** – persoana care foloseşte produsul sau serviciul.

[3] **SIP** – *Session Initiation Protocol.*

Client Control Protocol (Cisco), MiNET (Mitel), CorNet-IP (Siemens), IAX, Skype, Jajah, Jingle.

Vocea umană, ca şi orice altă mărime din natură, este o mărime analogă; pentru a putea fi trimisă printr-o rețea orientată pe comutație de pachete (rețeaua IP) ea trebuie digitalizată. Cele mai comune standarde de codare-decodare (CODEC) sunt prezentate în tabelul 2.1.

Tabelul 2.1

Standarde de codare–decodare (CODEC)

Standard de codare	Algoritm de codare	Lățime de bandă folosită
G.711	PCM (Pulse Code Modulation)	64 kbps
G.726	ADPCM (Adaptive Differential Pulse Code Modulation)	16, 24, 32, 40 kbps
G.728	LD–CELP (Low Delay Code Excited Linear Prediction)	16 kbps
G.729	CS–ACELP (Conjugate Structure Algebraic CELP)	8 kbps
G.723.1	MP–MLQ (Multi-Pulse Maximum Likelihood Quantization)	6,3 kbps, 5,3 kbps
	ACELP (Algebraic Code Excited Linear Prediction)	6,3 kbps, 5,3 kbps

Pentru identificarea terminalelor dintr-o rețea VoIP, se folosesc adresele IP, similar cum se folosesc numerele de telefon într-o rețea de telefonie normală.

2.2. Structura unei rețele VoIP

În figura 2.1 se arată interacțiunea dintre principalele componente ale unei rețele VoIP. Gateway-ul realizează conversia de la interfețele tradiționale de telefonie (POTS[1], T1/E1, ISDN, E&M) spre VoIP. Un telefon IP este terminalul capabil să susțină un apel VoIP şi care se poate conecta direct la o rețea IP. Terminalul se poate referi la un telefon IP sau la un PC cu o interfață VoIP conectată.

[1] **POTS** – *Plain Old Telephone Service* – serviciile disponibile de la telefoanele analogice până la introducerea centralelor telefonice electronice în rețeaua publică de telecomunicații (PSTN).

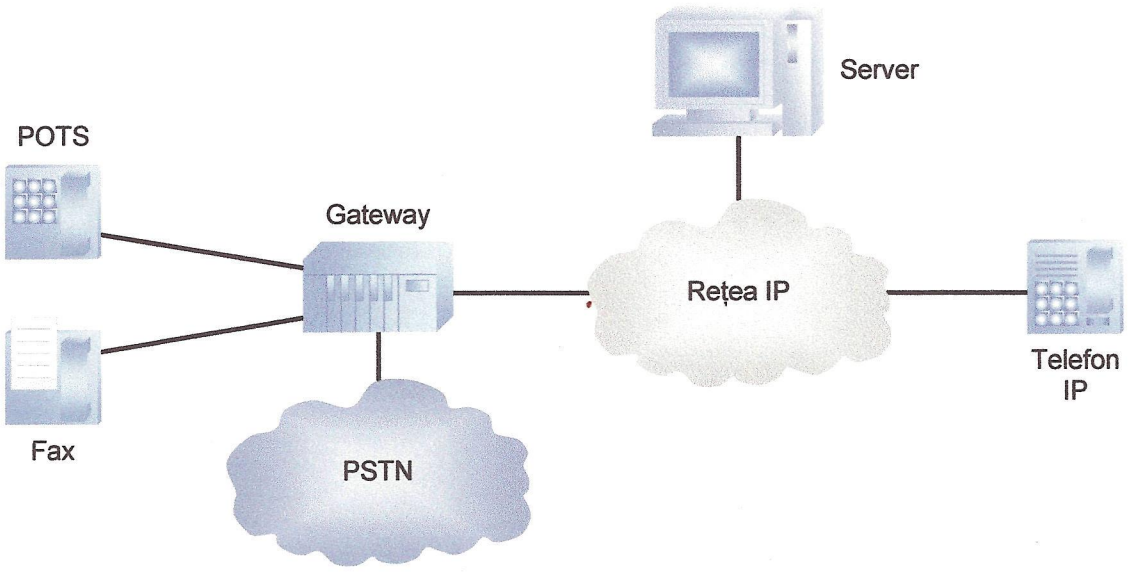

Fig. 2.1 – **Interacțiunea dintre principalele componente ale unei rețele VoIP**

Serverul oferă funcții administrative pentru a gestiona rutarea apelurilor prin rețea. Într-un sistem bazat pe H.323, serverul este cunoscut şi sub denumirea de gatekeeper, iar în cazul SIP/SDP serverul este cunoscut sub denumirea de SIP server. Într-un sistem bazat pe MGCP sau MEGACO, serverul este un agent de apeluri (call agent).

Rețeaua IP realizează conexiunile dintre terminale. Această rețea poate fi, după caz, o rețea privată, un Intranet, sau Internetul.

Odată ce apelul a fost inițiat, vocea va fi digitalizată şi apoi transmisă prin intermediul rețelei în cadre IP. Eşantioanele de voce sunt încapsulate în RTP (Real-time Transport Protocol) şi UDP (User Datagram Protocol) înainte de a fi transmise într-un cadru IP.

În figura 2.2 de observă un exemplu de cadru (frame) VoIP în LAN[1] şi în WAN[2].

---

[1] **LAN** – *Local Area Network* – este o rețea de comunicații date ce acoperă o suprafață mică, de exemplu o casă, un birou sau un grup mic de clădiri.

[2] **WAN** – *Wide Area Network* – este o rețea de comunicații de date ce acoperă o suprafață geografică mai mare şi face legătura între mai multe rețele de tip metropolitan.

De exemplu, dacă codecul folosit este G.711 şi timpul de realizare a pachetelor este de 20 ms, atunci informaţia utilă va avea 160 de octeţi. Aceasta va rezulta într-un cadru de date de 206 octeţi în WAN şi de 218 octeţi în LAN.

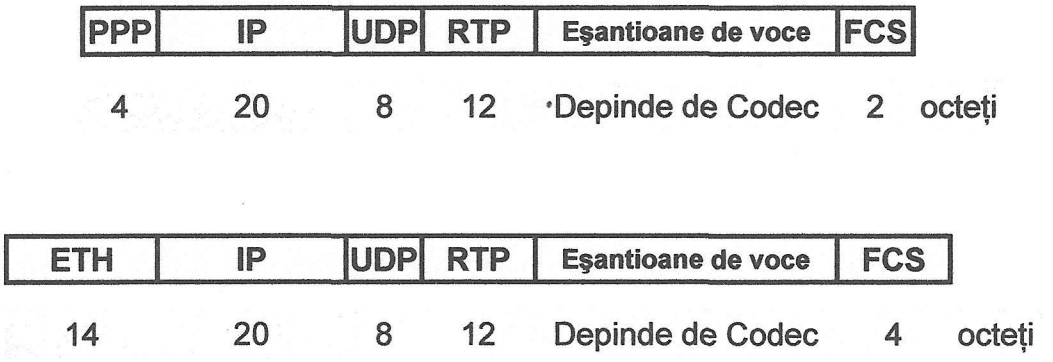

Fig. 2.2 – **Încapsularea cadrelor VoIP în LAN (sus) şi în WAN (jos)**

2.2.1. Serverul PBX pentru procesarea convorbirilor

Private Branch Exchange, sau Private Business Exchange este un echipament ce se comportă într-o reţea VoIP ca şi o centrală telefonică.

Serverul PBX îndeplineşte 3 mari funcţii:

- Stabileşte conexiunile dintre doi utilizatori ai reţelei (mapează numărul apelat spre telefonul fizic corespunzător, dacă acesta nu este ocupat);
- Menţine convorbirea atâta timp cât cei doi utilizatori doresc acest lucru (face schimb de semnale de voce între părţi);
- Furnizează informaţii despre costuri.

Pe lângă principalele funcţii, serverul PBX are şi alte capabilităţi, pentru a oferi operatorului o anumită flexibilitate în serviciile oferite clienţilor finali:

- Posibilitate de a suna în interiorul reţelei locale;
- Transfer automat al apelului;
- Speed dialing;
- Căsuţă vocală;

- Redirecționare automată către un post telefonic liber;
- Redirecționare în caz de ton ocupat;
- Muzică pentru apelul în așteptare;
- Robot telefonic;
- Serviciu de noapte;
- Apel în așteptare;
- Parcarea apelului;
- Apel în conferință;
- Gestionarea contului de client;
- Agendă telefonică.

2.2.2. **Terminalele**

Terminalele într-o rețea VoIP pot fi hardware (telefon fizic) sau software (aplicații care simulează comportamentul unui telefon). Acestea se comportă ca și un echipament obișnuit dintr-o rețea IP: are alocată o adresă IP și folosește stiva de protocoale TCP-IP. Pentru a simplifica instalarea terminalelor VoIP, în general se folosește DHCP[1] pentru autoconfigurarea lor. Serverul DHCP informează telefonul și despre locația serverului de configurare VoIP, care este în majoritatea cazurilor același cu serverul de procesare al apelului.

Aparatele telefonice software (softphones) sunt aplicații software ce rulează, în general, pe dispozitive mobile (laptop, pocket PC etc.) sau chiar și pe calculatoarele personale. Ele au aceleași caracteristici de bază ca și telefoanele VoIP.

Consolele, pe de altă parte, sunt aplicații cu anumite caracteristici de control. Consolele de obicei includ un softphone, dar pot interacționa și cu un telefon normal, prin intermediul unui gateway. Consolele sunt aplicații speciale menite pentru a controla distribuția apelurilor. Acestea includ console pentru recepționist, care au abilitatea că conecteze apeluri, console executive, care pot observa statusul diferitelor grupuri de telefoane și console pentru relații cu clienții, cu abilitatea de a suporta

[1] **DHCP** – *Dynamic Host Configuration Protocol* este un protocol de rețea client–server care furnizează informațiile necesare unui echipament nou pentru a comunica într-o rețea IP.

distribuția apelurilor. Distincția între diferitele tipuri de console nu este prea clară. Toate consolele VoIP au în comun folosirea aceloraşi seturi de protocoale.

2.2.3. **Termenii Gateway şi Gatekeeper**

Cei doi termeni, gateway şi gatekeeper sunt deseori folosiți ca termeni interschimbabili. În mod tradițional, gatekeeper-urile sunt, în principal, folosite pentru recepția apelurilor şi managementul controlului şi al lățimii de bandă. Dar acest lucru s-a schimbat recent, deoarece tehnologia a permis ca aceste funcționalități să coexiste cu gateway-urile clasice.

Principala diferență dintre un ruter şi un gateway este aceea că ruterul face legătura între două rețele de acelaşi tip (de exemplu, IP–IP), iar gateway-ul face legătura între rețele diferite (de exemplu, IP–PSTN). Funcția principală a gateway-urilor este conversia analog–digitală a vocii şi crearea de pachete IP (funcții ale CODEC-ului). În plus, gateway-urile au funcționalități opționale, cum ar fi compresia vocii, anularea ecoului, suprimarea zgomotului de fond şi adunarea de date statistice.

Gateway-ul formează interfața pe care vocea o foloseşte pentru a fi transportată într-o rețea IP. În mod normal, fiecare conversație reprezintă o singură sesiune IP transportată de un protocol RTP peste UDP sau TCP.

Gateway-urile există în mai multe forme. De exemplu, gateway-urile pot fi echipamente de telecomunicații dedicate sau chiar calculatoare PC pe care rulează software VoIP.

În funcție de caracteristicile şi serviciile oferite putem avea:

- *Gateway-uri de trunchiere*, care interfațează între rețeaua telefonică şi rețeaua VoIP. Astfel de gateway-uri de obicei conduc un număr mare de circuite digitale.
- *Gateway-uri rezidențiale* care asigură o interfață tradițională analogică unei rețele VoIP. Exemple de astfel de gateway-uri sunt: modem-uri de cablu, dispozitive DSL şi dispozitive wireless.

- *Gateway-uri de acces* care oferă o interfață PBX analogică sau digitală unei rețele VoIP.
- *Gateway-uri din clasa „business"* care oferă o interfață digitală tradițională PBX sau un soft integrat PBX pentru a interfața o rețea VoIP.
- *Servere de acces la rețea* care, prin atașarea unui modem la un circuit telefonic, asigură acces la Internet.

2.2.4. **Rețeaua IP**

Rețeaua IP poate fi văzută ca un switch logic. Oricum, acest switch logic poate fi văzut ca un sistem distribuit, mai mult decât o singură entitate. În funcție de protocoalele IP folosite, acest sistem văzut ca şi un întreg este referit ca o arhitectură „softswitch".

Infrastructura IP trebuie să asigure o distribuție liniară a vocii şi a pachetelor de date către elementele VoIP. Datorită disimilarităților, rețeaua IP trebuie să trateze fluxul de voce şi fluxul de date diferit. Dacă o rețea IP este folosită pentru a transporta atât voce cât şi date, trebuie să poată prioritiza tipurile diferite de trafic, deoarece traficul VoIP este extrem de sensibil la latență.

În timp ce există diferite similarități între VoIP şi circuitele de comutare, există de asemenea şi câteva diferențe. Una dintre ele este în transportul traficului de voce rezultat. Telecomunicațiile tradiționale pot fi clasificate ca şi rețele TDM[1] care dedică canale, rezervă lățime de bandă necesară interconectând switch-urile. De exemplu, o conversație telefonică rezervă un singur canal DS-0[2], şi conexiunea este folosită doar pentru o singură conversație. Aceasta nu este o modalitate eficientă de utilizare a resurselor.

Rețelele IP sunt diferite de infrastructura tradițională. Aceste resurse ale rețelei nu sunt legate pe întreaga durată a convorbiri, spre deosebire de rețelele tradiționale. Clasa de servicii (CoS) asigură că pachetele unei aplicații anume au o anumită

[1] **TDM** – *Time-division multiplexing* – metodă de a trimite semnale digitale multiple (prin multiplexare în timp) pe o singură linie de telecomunicații

[2] **DS-0** – *Digital Signal 0* – este capacitatea unei linii de bază în telecomunicații, capabilă să transfere o capacitate de 64 kbps, echivalentul unui singur canal de voce

prioritate. Această prioritizare este necesară pentru aplicațiile VoIP în timp real pentru a asigura faptul că serviciile de voce nu sunt afectate de alte fluctuații ale traficului.

2.3. Implementarea VoIP

Deoarece rețeaua IP nu furnizează un mecanism care să asigure ca pachetele de date să fie transmise în ordine secvențială, sau să furnizeze garantări ale calității serviciului (QoS), implementările VoIP se confruntă cu probleme legate de latență şi jitter. Acest lucru se întâmplă mai ales atunci când în circuit apar şi conexiuni prin intermediul sateliților. Nodul de recepție trebuie să restructureze pachetele IP – care s-ar putea să nu fie primite în ordine, să fie întârziate sau să lipsească – pentru a ne asigura că fluxul audio rămâne într-o proporție cât mai mare întreg. Acest lucru este, de obicei, realizat cu ajutorul unui buffer.

Un alt aspect important este rutarea traficului VoIP prin firewall-uri şi translatoare de adrese (NAT[1]). SBC[2]-urile private sunt folosite, alături de firewall-uri pentru a permite apelurilor VoIP să pătrundă în rețelele private ale companiilor. De exemplu, Skype foloseşte un protocol propriu pentru a ruta apelurile prin intermediul altor noduri Skype din rețea, reuşind să treacă de NAT-uri şi firewall-uri. Alte metode de a traversa firewall-uri implică folosirea unor protocoale ca STUN[3] sau ICE[4].

---

[1] **NAT** – *Network Address Translation* – proces ce implică rescrierea adreselor (sursă şi/sau destinație) din interiorul pachetelor IP, în momentul în care acestea trec printr-un ruter sau firewall.

[2] **SBC** – *Session Border Controller* – dispozitiv folosit în unele rețele VoIP cu rolul de a prelua controlul asupra fluxului de semnalizare şi date.

[3] **STUN** – *Simple Traversal of UDP over NATs* – este un protocol de rețea ce permite clienților de după NAT să îşi descopere adresa publică, tipul de NAT şi portul asociat de NAT cu un port local.

[4] **ICE** – *Interactive Connectivity Establishment* – furnizează un mecanism de traversare al NAT, folosit mai ales pentru a permite clienților de voce SIP să treacă de diverse firewall-uri.

Capitolul 3

PROTOCOALE VoIP

3.1. Setul de protocoale H.323

3.1.1. Generalități

H.323 este un set de protocoale, recomandate de ITU-T, ce definesc comunicații de tip audio-video într-o rețea orientată pe comutație de pachete (figura 3.1).

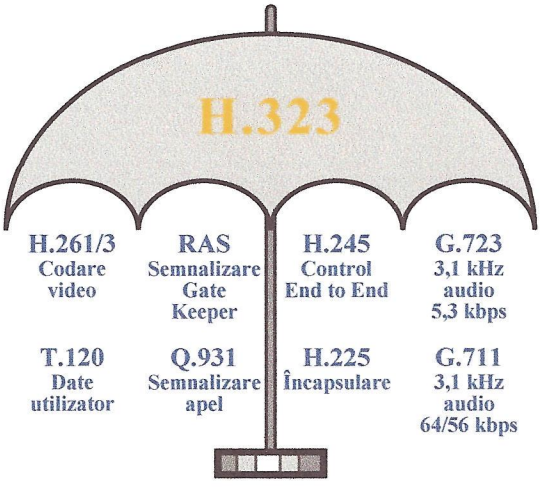

Fig. 3.1 – **Setul de protocoale H.323**

În prezent, H.323 este implementată în multe aplicații în timp real, de exemplu NetMeeting sau Ekiga. H.323 face parte din seria de protocoale H.32X care definesc și relaționarea unei rețele VoIP cu rețele de telecomunicații clasice: ISDN, PSTN, wireless sau SS7 (*Signalling System 7* – sistem de semnalizare în PSTN). H.323 este folosit în mod curent în VoIP, precum și în video conferințe bazate pe rețeaua IP.

H.323 a fost inițial creat pentru a furniza un mecanism de transport a aplicațiilor multimedia în rețelele LAN, dar a evoluat rapid și a ajuns să acopere majoritatea nevoilor unei rețele VoIP.

Ca și părți componente ale unui sistem VoIP bazat pe protocoalele H.323, putem aminti: terminalele, proxy-ul, gateway-ul și gatekeeper-ul (figura 2.3), entități detaliate în continuare.

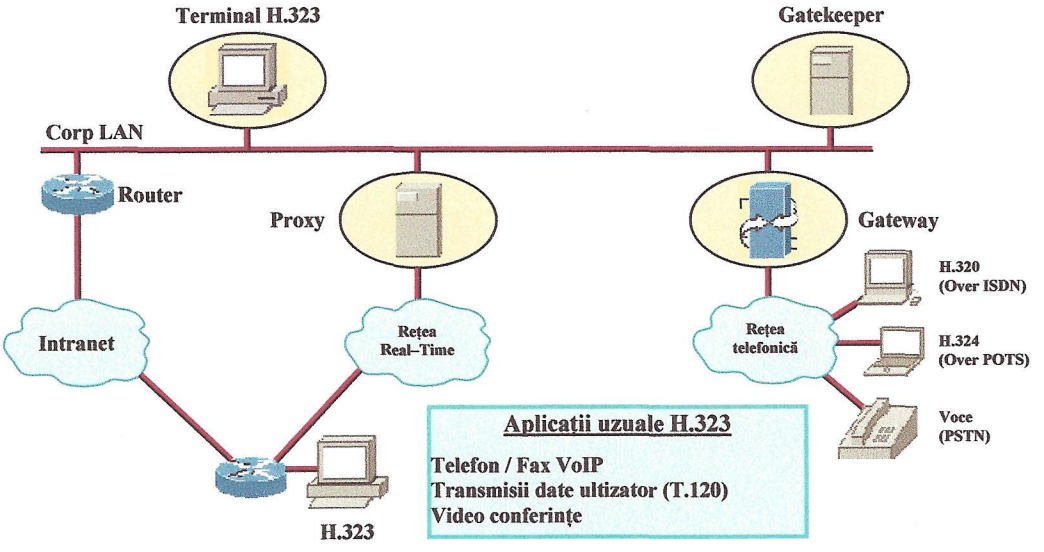

Fig. 3.2 – **Infrastructura H.323** (după CISCO Systems)

3.1.2. **Terminalele H.323**

Terminalele H.323 sunt terminale de tip LAN pentru transmisia vocii. Exemple comune de astfel de terminale sunt calculatoare personale ce rulează Microsoft NetMeeting și au un microfon de rețea.

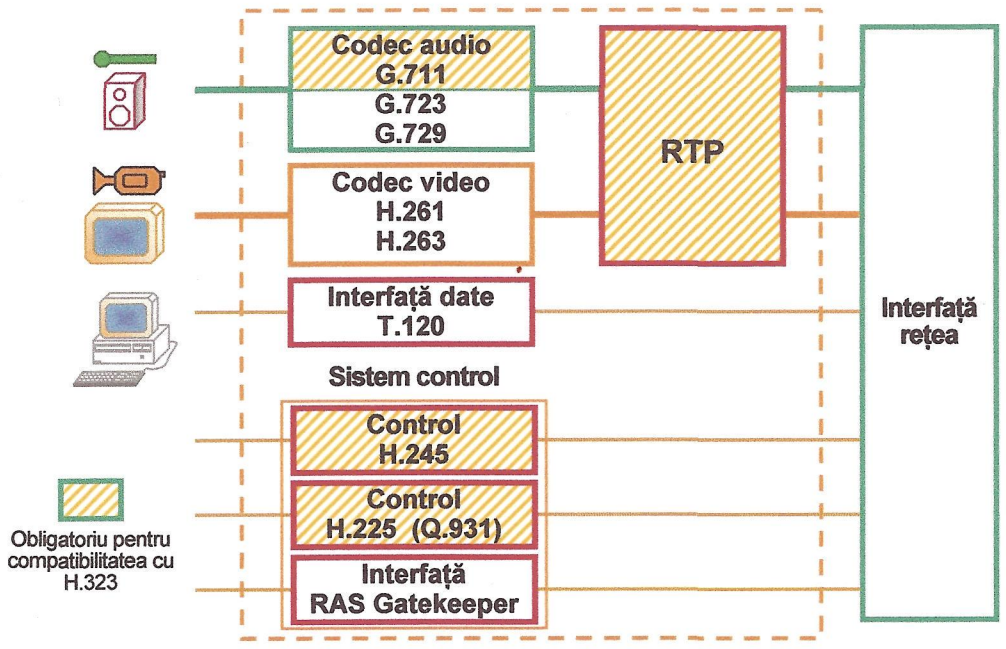

Fig. 3.3 – **Configurația unui terminal H.323**

Terminalele H.323 implementează funcții de transmitere a vocii şi includ cel puțin un CODEC de voce (Compressor/Decompressor) care trimite şi recepționează voce pachetizată. Codec-uri mai întâlnite sunt: ITU–T G.711 (PCM), G.723 (MP–MLQ), G.729A (CA–ACELP) şi GSM. Codec-urile diferă prin cerințele CPU-lui, prin calitatea vocii rezultate şi prin inerenta întârziere de procesare.

Terminalele, de asemenea, trebuie să suporte funcții de semnalizare. Standardele care se aplică aici sunt: semnalizările H.225.0 care sunt un subset al semnalizărilor din ISDN (Q.931); H.245 care este utilizat pentru schimbul de informații între terminale ca de exemplu standardele de compresie care pot fi diferite; RAS (Registration, Admission, Status) care conectează un terminal la un gatekeeper. Terminalele mai pot implementa capabilități de comunicații video şi de date, însă nu fac obiectul studiului nostru.

3.1.3. Gateway şi gatekeeper în H.323

Un gateway H.323 este un „punct final" al reţelei care oferă comunicaţie în timp real, în ambele sensuri, între terminale H.323 din reţeaua IP şi alte terminale ITU dintr-o reţea de comutaţie sau cu un alt gateway H.323. Gateway-ul realizează funcţia de translaţie între diferite formate de transmisie, de exemplu din H.225 în H.221. De asemenea poate face translaţia între codec-uri audio şi video. Gateway-ul este interfaţa între PSTN şi Internet: preia vocea din reţeaua PSTN şi o plasează pe Internet, sau invers. Într-o reţea simplă LAN, în care terminalele comunică direct între ele, gateway-urile sunt opţionale. Atunci când terminalele dintr-o reţea trebuie să comunice cu alte terminale din altă reţea, se folosesc gateway-ul şi protocoalele H.245 şi Q.931.

Gateway-ul poate avea de la 2 porturi (vezi figura 3.4) – folosit, de obicei pentru aplicaţii personale (acasă) – până la mii de porturi (vezi figura 3.5) – cu aplicaţie industrială.

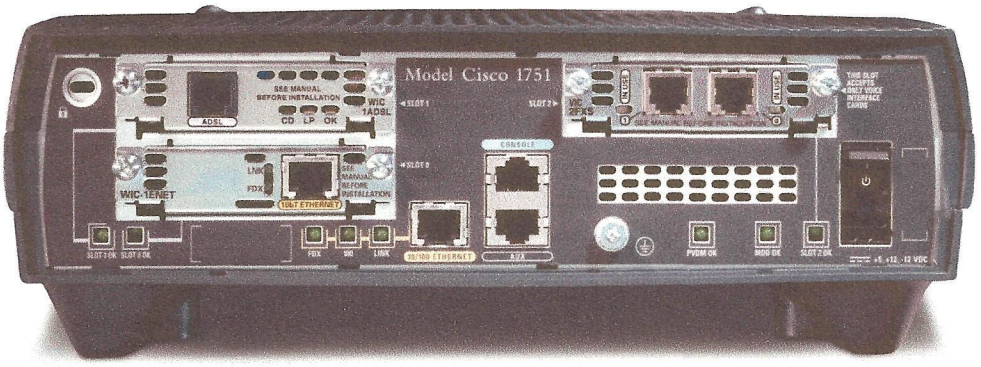

Fig. 3.4 – **Gateway cu două porturi VoIP (Model Cisco 1751)**

Gatekeeper-ul din specificaţiile protocolului H.323 îndeplineşte funcţiile de translatare a adreselor, controlul admisiei, semnalizarea apelului, autorizarea apelului, managementul lăţimii de bandă, managementul apelului. Gatekeeper-ul este răspunzător pentru realizarea conexiunilor H.323 într-o reţea orientată pe comutaţia de pachete.

Acesta poate, de asemenea, interzice accesul sau să limiteze numărul de conexiuni simultane pentru a evita aglomerarea rețelei.

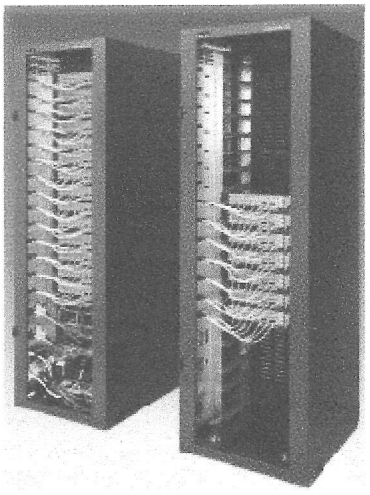

Fig. 3.5 – **CISCO AccessPath-VS3, soluție VoIP pe scară largă, realizat cu gateway-uri de voce Cisco AS5300**

MCU (Multipoint Control Unit) permite funcții de conferință între trei sau mai multe terminale. Un MCU conține 2 părți:

- Multipoint Controller (MC) care se ocupă cu semnalizările şi mesajele de control necesare conferințelor;
- Multipoint Processor (MP) care primeşte semnalele de la terminale, le multiplică şi le trimite apoi către participanții la conferință.

Un MCU poate implementa ambele funcții atât ale MP cât şi ale MC, caz în care este denumit MCU centralizat. Alternativ, un MCU descentralizat implementează doar funcțiile MC, lăsând funcția de multipoint processor pentru terminalele participante.

Este important de reținut că definirea tuturor părților unei rețele H.323 este pur logică. Nici o specificație nu a fost dată pentru divizarea fizică a unităților. De exemplu, MCU poate fi un echipament de sine stătător sau poate fi integrat într-un terminal, gateway sau gatekeeper.

3.1.4. Stiva de protocoale

În ceea ce priveşte protocoalele folosite, care sunt prezentate în figura 3.6, pentru transferul fluxului audio şi a celui video se foloseşte protocolul UDP (User Datagram Protocol, iar pentru date se foloseşte protocolul TCP (Transmission Control Protocol).

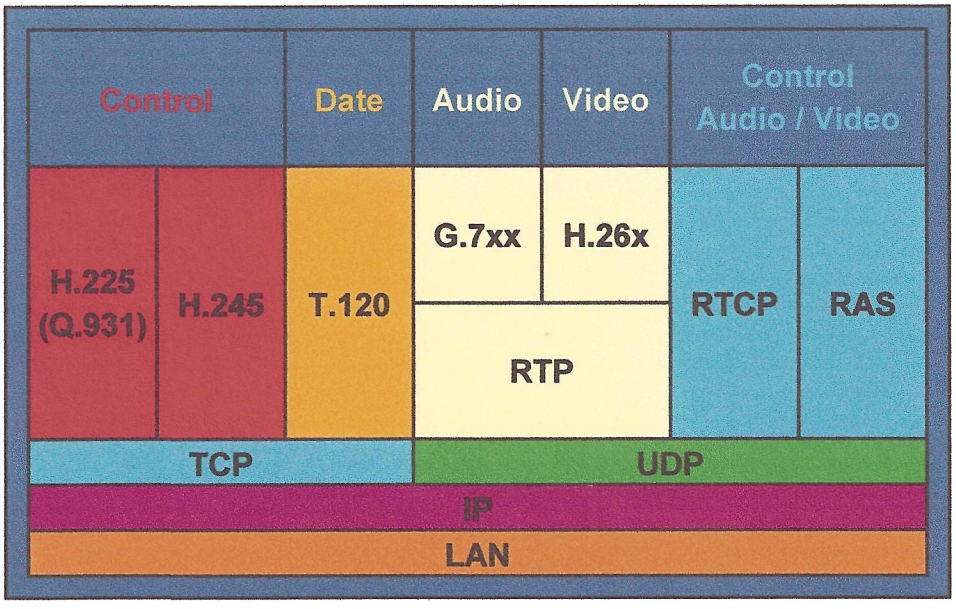

Fig. 3.6 – **Stiva de protocoale H.323**

Mesajele de control (semnalizarea Q.931, negocierea parametrilor H.245 şi protocolul RAS) se transportă peste nivelul TCP. Aceasta asigură ca mesajele importante să fie transmise cu siguranţă. Traficul de media este transportat de către mai puţin sigurul protocol UDP şi include 2 protocoale definite de ITU-T în RFC 1889: RTP (Real-time Transport Protocol) care transportă chiar informaţia şi RTCP (RTP Control Protocol) care transmite mesaje periodice de control şi de stare. Informaţia este transportată de UDP deoarece nu ar avea sens să fie retransmisă (aşa cum ar proceda TCP de exemplu): pierderea unui fragment de sunet dacă ar fi retransmis, el probabil ar ajunge prea târziu şi nu ar mai putea fi folosit la reconstruirea vocii. Mesajele RTP sunt transmise pe porturile UDP impare, iar cele RTCP pe cele pare imediat alăturate.

3.1.5. **Controlul şi semnalizarea în H.323**

Standardul H.323 oferă trei protocoale de control H.225/Q.931 Call Signaling, H.225.0 RAS şi H.245 Media Control. H.225/Q.931 este folosit în conjuncţie cu H.323 şi oferă semnalizare pentru controlul apelului. Pentru stabilirea unui apel de la o sursă la un destinatar, este folosit canalul H.225 RAS (Registration, Admission and Signaling). După ce s-a stabilit legătura, H.245 este folosit pentru a negocia fluxurile media.

Canalul RAS este folosit pentru comunicaţia între terminale şi gatekeeper. Deoarece mesajele RAS sunt transmise peste UDP (canal nesigur), se recomandă monitorizări ale timpului de transmisie şi, dacă este cazul, reîncercări de transmitere a mesajelor. Procedurile definite de canalul RAS sunt următoarele:

– Determinarea Gatekeeper-ului este procesul prin care un terminal îl foloseşte pentru a descoperi gatekeeper-ul care îl va folosi. În mod normal, terminalul transmite o cerere către Gatekeeper (GRQ – Gatekeeper Request Message). Unul sau mai multe gatekeeper-uri pot răspunde cu un mesaj de confirmare (GCF – Gatekeeper Confirmation Message) prin care îşi exprimă disponibilitatea de a prelua terminalul respectiv. Acest răspuns include adresa de transport a canalului RAS al gatekeeper-ului. Gatekeeper-urile care nu doresc să-şi înregistreze terminalul pot trimite un mesaj de respingere (GRJ – Gatekeeper Reject Message). Dacă mai mult de un Gatekeeper răspunde cu GCF, atunci gatekeeper-ul poate fi ales de către terminal. Dacă nici un gatekeeper nu răspunde într-un anumit interval de timp, terminalul poate retransmite mesajul GRQ.

– Înregistrarea gatekeeper-ului este procesul prin care un terminal se alătură unei zone şi informează gatekeeper-ul de adresa sa de transport şi de alias. Toate terminalele se înregistrează la gatekeeper-ul care a fost identificat în urma procesului gatekeeper discovery şi foloseşte binecunoscutul canal RAS TSAP. Gatekeeper-ul răspunde ori cu o Registration Confirmation (RCF), sau cu Registration Reject (RRJ). Gatekeeper-ul trebuie să se asigure că fiecare adresă alias translatează în mod unic către o singură adresă de transport. Un terminal poate să-şi anuleze înregistrarea unui terminal trimiţând un mesaj Unregister Request (URQ) către terminal. Terminalul ar trebui să răspundă cu un mesaj Unregisrter Confirmation (UCF).

– Un terminal sau un gatekeeper care are o adresă alias pentru un terminal şi ar dori să determine informaţii referitoare la contactul acestuia, poate transmite un mesaj Location Request (LRQ). Gatekeeper-ul la care este înregistrat terminalul răspunde cu un mesaj Location Confirmation (LCF) care conţine informaţii de contact ale terminalului sau ale gatekeeper-ului. Toate gatekeeper-ele care nu au înregistrat terminalul respectiv şi au primit mesaj LRQ pe canalul RAS vor răspunde cu un mesaj Location Reject (LRJ).

– Canalul RAS este folosit de asemenea pentru transmiterea admisiei, schimbării lăţimii de bandă, a statusului şi dezactivării prin intermediul mesajelor. Aceste mesaje care sunt schimbate între terminale şi gatekeeper-uri sunt folosite pentru a oferi controlul admisiei şi funcţii de management al lăţimii de bandă. Mesajele Admission Request (ARQ) specifică lăţimea de bandă necesară apelului. Gatekeeper-ul poate reduce lăţimea de bandă cerută prin mesajul ACF. Un terminal sau chiar gatekeeper-ul poate încerca să modifice lăţimea de bandă în timpul apelului printr-un mesaj Bandwidth Change Request (BRQ).

Canalul de semnalizare este folosit pentru a transporta mesaje de control H.225. În reţelele care nu conţin un gatekeeper, mesajele de semnalizare a apelului sunt direcţionate între terminalele sursă şi destinaţie folosind Call Signaling Transport Addresses. Se presupune că terminalul sursă cunoaşte adresa terminalului apelat, şi de aceea poate comunica în mod direct. În reţelele care conţin gatekeeper, schimbul de mesaje de admisie are loc între terminalul apelant şi gatekeeper-ul care foloseşte adresa RAS a canalului de transport. Semnalizarea apelului este efectuată utilizând TCP (canal sigur).

Mesajele de semnalizare pot fi împărţite în două categorii. Prima categorie cuprinde mesajele de semnalizare care sunt rutate prin gatekeeper printre terminale. Cealaltă alternativă reprezintă mesajele care sunt transmise direct la terminale. Mesajele de admisie sunt schimbate cu gatekeeper-ul folosind canalul RAS, urmate de un schimb de mesaje de semnalizare.

Când este folosit apelul de semnalizare, sunt două metode de a ruta Canalul de Control H.245: prima alternativă stabileşte canalul H.245 direct între terminale, iar al în doilea caz, stabilirea canalului de control H.245 se face prin gatekeeper.

3.1.6. Arhitecturi H.323

Vom exemplifica un apel de voce între telefonul A şi telefonul B din figura 3.7, folosind o rețea ce respectă protocoalele H.323 şi realizează conexiuni cu un grad de securitate ridicat, precum şi cu sistem de taxare. Gateway-urile stabilesc un canal de comunicație sigur cu gatekeeper-urile.

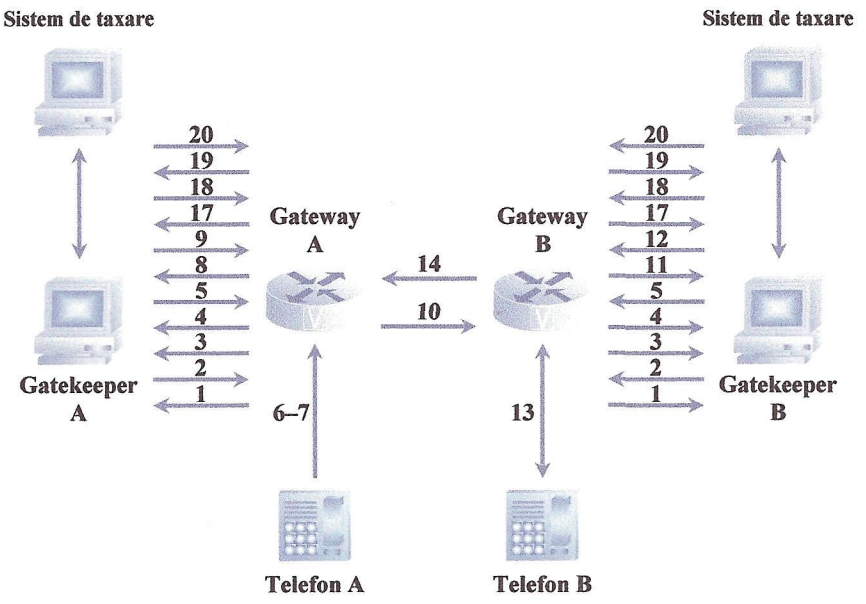

Fig. 3.7 – **Exemplu de apel într-o rețea H.323** (după CISCO)

1. Gateway-urile A şi B trimit requestul GRQ spre gatekeeper-urile corespunzătoare. Mesajul GRQ include capabilități de autentificare şi identificatorul algoritmului (algoritm object ID).

2. Gatekeeper-urile A şi B răspund gateway-urilor corespunzătoare cu mesajul gatekeeper confirmation (GCF). Mesajul GCF include capabilități de autentificare şi identificatorul algoritmului (algoritm object ID).

3. Dacă valorile pentru parametrii de securitate ai H.323 sunt sub aşteptări, atunci gatekeeper-ul răspunde cu un mesaj de refuz (GRJ) care conține şi motivul refuzului. Acest lucru obligă gateway-ul să retrimită GRQ.

4. Gateway-urile A şi B trimit semnalul de înregistrare (RRQ) către gatekeeper-ele corespunzătoare. Acest mesaj conţine şi informaţii de autentificare encriptate.

5. Gatekeeper-urile A şi B răspund gateway-urior corespunzătoare cu un mesaj de confirmare al autentificării (RCF).

Dacă se produce o eroare a autentificării, gatekeeper-ul răspunde cu un mesaj de refuz al autentificării (RRJ).

Se iniţiază un canal de telecomunicaţii sigur.

6. Telefonul A realizează o conexiune cu Gateway-ul A

7. Gateway-ul A iniţiază un răspuns interactiv de voce (IVR) pentru a obţine numărul contului şi PIN-ul utilizatorului, precum şi numărul de telefon destinaţie.

8. Gateway-ul A trimite o cerere de admisie (ARQ) către gatekeeper-ul A. Gateway-ul trebuie să includă în mesajul ARQ informaţii adiţionale pentru a permite gatekeeper-ului să autentifice apelul. Informaţia inclusă în mesajul ARQ variază dacă acesta este trimis de gateway-ul sursă sau de cel destinaţie. În acest moment, gateway-ul sursă iniţiază admisia. Astfel, mesajul ARQ include contul şi PIN-ul userului. Această informaţie este criptată folosind MD5[1].

9. Gatekeeper-ul A validează informaţiile de autentificare, detaliază numărul de telefon destinaţie şi determină gateway-ul potrivit pentru acest apel (în acest caz – gateway-ul B). Gatekeeper-ul A trimite un mesaj de confirmare a admisiei (ACF) către gateway-ul A. Mesajul ACF conţine informaţii de taxare a utilizatorului.

10. Gateway-ul A trimite un mesaj de iniţiere către Gateway-ul B.

11. Gateway-ul B trimite un mesaj ARQ către gatekeeper-ul B.

12. Gatekeeper-ul B validează informaţia de autentificare şi răspunde gateway-ului B cu un mesaj ACF.

Dacă informaţia de autentificare nu este corectă, gatekeeper-ul B trimite un mesaj de refuz (ARJ) către gateway-ul B cu motivul de security Denial.

13. Gateway-ul B iniţiază apelul către telefonul destinaţie.

14. În momentul în care telefonul de la destinaţie răspunde, gateway-ul B trimite un mesaj de conectare către gateway-ul A.

---

[1] MD5 – *Message-Digest Algorithm 5* – algoritm criptografic pe 128 de biţi utilizat pe scară largă

15. Gateway-urile A şi B îşi pornesc cronometrele pentru a măsura durata apelului. Dacă apelantul foloseşte servicii de tip "prepaid[1]" atunci timpul vorbit se compară în permanenţă (real-time) cu creditul disponibil în cont, care a fost inclus în mesajul ACF de la pasul 9.

Comunicaţiile telefonice sunt încheiate

16. Apelul este încheiat când una dintre părţi închide sau, în cazul serviciilor prepaid, unul din gateway-uri decide că limita de credit a utilizatorului a fost depăşită.

17. Gateway-urile A şi B trimit mesaje DRQ către gatekeeper-ul corespunzător. Mesajul DRQ conţine rezultatul informaţiei de taxare.

18. Gatekeeper-ele A şi B trimit informaţii legate de deconectare (DCF) către gateway-urile corespunzătoare.

Comunicaţia dintre gateway-uri şi gatekeeper-uri este terminată

19. Gateway-urile A şi B trimit mesaje URQ către gatekeeper-urile corespunzătoare.

20. Gatekeeper-urile A şi B trimit informaţii de dezînregistrare (UCF) către gateway-urile corespunzătoare.

3.2. **Session Initiation Protocol (SIP)**

3.2.1. **Generalităţi**

Protocolul SIP (Session Initiation Protocol) este un protocol de semnalizare folosit la iniţierea, menţinerea şi încheierea sesiunilor de voce şi video într-o reţea orientată pe pachete. Într-o sesiune SIP poate exista comunicare unicast[2] sau multicast[3].

---

[1] Servicii "Prepaid" – servicii plătite în avans

[2] **Unicast** – transmiterea pachetelor de date către o singură destinaţie.

[3] **Multicast** – transmiterea pachetelor de date către mai multe destinaţii simultan, folosind cea mai eficientă strategie.

Protocolul SIP este dezvoltat de către SIP Working Group, din cadrul IETF (Internet Engineering Task Force). Protocolul a fost publicat în RFC[1] 2543.

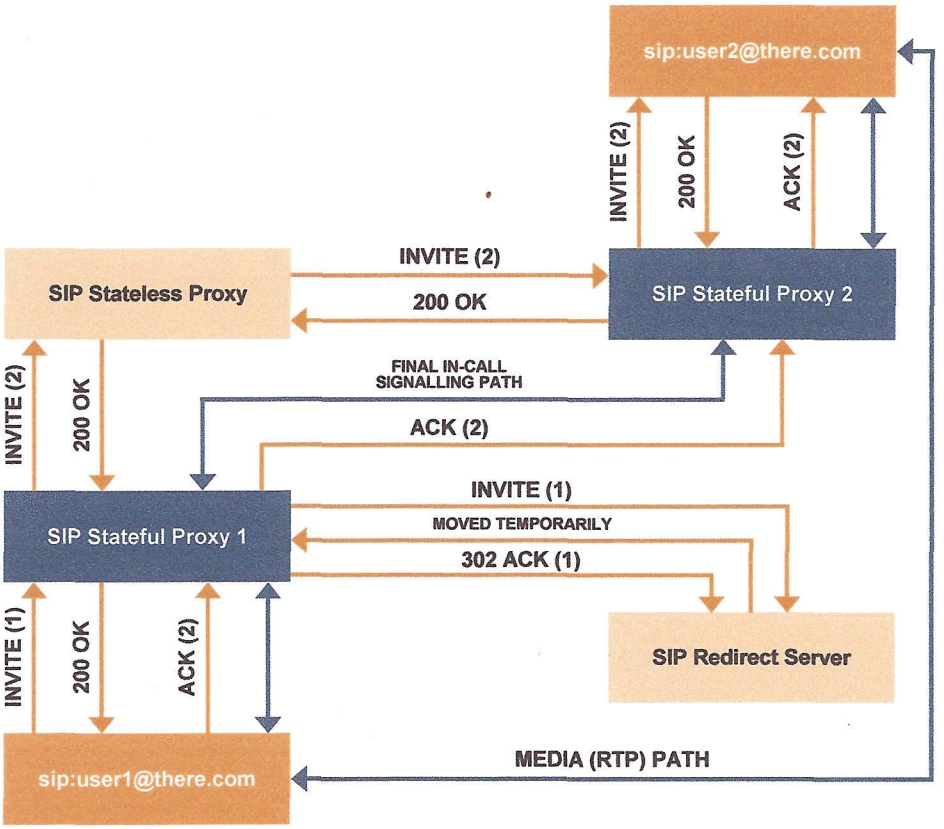

Fig. 3.8 – **Exemplu de semnalizare SIP (Session Initiation Protocol)**

3.2.2. Entităţi SIP

O reţea SIP este compusă din patru tipuri de entităţi SIP. Fiecare entitate SIP are funcţii specifice şi participă în comunicare ca un client (iniţiază cereri), ca un server (răspunde la cereri), sau ca şi ambele. Un singur dispozitiv hardware poate oferi funcţionalităţile unei sau mai multor entităţi SIP. De exemplu, un server care se comportă ca un Proxy, poate să preia şi funcţia de Registrer în acelaşi timp.

---

[1] Documentele **RFC** (*Request for Comments*) sunt documente referitoare la noile tehnologii Internet.

Prima entitate SIP este **clientul** (user agent) care este o aplicație formată din două module principale: user agent client (inițiază cereri SIP) şi user agent server (un server care aşteaptă cereri SIP de la alt client şi informează utilizatorul când se primeşte o astfel de cerere). Dispozitivele care pot avea rol de client într-o rețea SIP sunt: stațiile de lucru, telefoanele IP, gateway-urile de telefonie, roboți telefonici etc.

A doua entitate SIP este **serverul proxy** care este se comportă ca un server, dar şi ca un client pentru a putea realiza cereri din partea altor clienți. Cererile se pot procesa intern, sau, după o translatare corespunzătoare, se trimit la alte servere. Serverul proxy interpretează, şi, la nevoie, rescrie o cerere înainte să o trimită mai departe.

A treia entitate SIP este **serverul de redirectări** (redirect server) care are rolul de a accepta cereri SIP, de a mapa adresa SIP a apelantului şi de a le întoarce clientului. Spre deosebire de serverele proxy, serverele de redirectări nu trimit cererea spre alte servere.

Ultima entitate a SIP-ului este **registar**-ul; acesta acceptă cereri de înregistrare, cu scopul de a actualiza locația unui utilizator din rețea.

3.2.3. Mesaje

Într-o rețea SIP există două tipuri de mesaje: cerere şi răspuns. Cererile sunt trimise de la client la server (tabelul 3.1), iar răspunsurile sunt trimise de către server spre client (tabelul 3.2).

Tabelul 3.1

Tipuri de cereri SIP

Cerere	Descriere
INVITE	Inițiază un apel, schimbă parametrii apelului (re-INVITE)
ACK	Confirmă un răspuns final pentru INVITE
BYE	Termină un apel
CANCEL	Anulează căutările după destinație sau apelarea (când sună)
OPTIONS	Interoghează capabilitățile celeilalte părți
REGISTER	Se înregistrează în serviciul de localizare (Location Service)
INFO	Trimite informații intermediare care nu schimbă parametrii sesiunii

Tabelul 3.2

Exemple de tipuri de răspuns SIP

100	Continuă	**408**	Cerere expirată (time-out)
180	Sună	**480**	Indisponibil
200	OK	**481**	Tranzacţia nu există
300	Alegeri multiple	**482**	Ciclu infinit
301	Mutat definitive	**5xx**	Eroare din server
302	Mutat temporar	**600**	Ocupat
400	Cerere greşită	**603**	Destinatarul respinge apelul
401	Neautorizat	**604**	Destinatarul nu există
403	Interzis	**606**	Cererea nu este acceptată

Mesajele de răspuns conţin coduri numerice de răspuns. Răspunsul codificat SIP este bazat în mare parte pe răspunsurile standardizate HTTP. Există două tipuri de răspuns şi şase clase:

Tipuri de răspuns:

- Parţiale (clasa 1xx) – sunt folosite de către SIP pentru a indica statusul tranzacţiilor, dar nu pentru a le termina;
- Finale (clasele 2xx, 3xx, 4xx, 5xx şi 6xx) – sunt folosite pentru a termina tranzacţii SIP.

Clase:

- 1xx – iniţiator, realizează acţiuni de căutare, cerere apel, sunat;
- 2xx – succes;
- 3xx – redirectare, trimitere mai departe (forward);
- 4xx – cerere eşuată din vina clientului;
- 5xx – eroare din cauza serverului;
- 6xx – eroare generală (sună ocupat, destinatarul respinge apelul, destinatarul nu este disponibil niciunde în reţea).

Părţile componente ale unui mesaj sunt:

- Linia de start – conţine tipul mesajului şi versiunea protocolului şi poate fi o linie de status sau o linie de cerere

- Antete (headers) – câmpurile din antetul SIP este folosit pentru a transmite atribute ale mesajului sau pentru a-i modifica scopul; sunt similare cu elementele din antetul HTTP:

<nume>:<valoare>

- Conținut – include descrierea sesiunii care urmează să fie inițiată, de exemplu într-o sesiune multimedia aceasta poate include tipurile de codec-uri audio şi video, ratele de eşantionare, etc.) Tipuri de date adesea întâlnite în „body": SDP (Session Description Protocol), Multipurpose Internet Mail Extensions (MIME).

În tabelele 3.3 şi 3.4 sunt prezentate exemple de mesaje: utilizatorul SIP mihai@rusoaie.com îl invită pe andrei@cs.utt.ro la o convorbire despre prânz.

Tabelul 3.3

Exemplu de mesaj cerere SIP

Mesajul cererii	Descriere
INVITE sip:andrei@cs.utt.ro SIP/2.0	Tipul cererii, adresa SIP a destinatarului, versiunea SIP folosită
Via: SIP/2.0/UDP	Adresa nodului anterior
mihai_ws.rusoaie.com	
From: Mihai RUSOAIE <sip:mihai@rusoaie.com>	Adresa celui care adresează această cerere
To: Andrei POPESCU <sip:andrei@cs.utt.ro>	Utilizatorul invitat
Call-ID: 2386390012@mihai_ws.rusoaie.com	Număr unic de identificare al acestui apel
CSeq: 1 INVITE	Ordinea comenzilor – identifică tranzacția
Subject: Masa de prânz	Subiectul sau natura apelului
Content-Type: application/SDP	Tipul conținutului, în acest caz: SDP
Content-Length: 182	Lungimea conținutului în bytes
	Linia goală semnifică terminarea antetului SIP şi începerea conținutului
v = 0	Versiunea SDP
o = Mihai 53655765 2353687637 IN IP4 81.181.24.88	Inițiatorul sesiunii, versiunea sesiunii şi adresa
s = Apel de la Mihai.	Subiectul sesiunii
c = IN IP4 mihai_ws.rusoaie.com	Informații despre conexiune
M = audio 3456 RTP/AVP 0 3 4 5	Informații despre capabilitățile apelantului: tip, port, formate posibile pe care apelantul le suportă

Tabelul 3.4

Exemplu de mesaj răspuns SIP

Mesajul răspunsului	Descriere
SIP/2.0 200 OK	Status: versiune SIP, codul de răspuns, motiv
Via: SIP/2.0/UDP mihai_ws.rusoaie.com	Copiat din cerere
From: Mihai RUSOAIE <sip:mihai@rusoaie.com>	Copiat din cerere
To: Andrei POPESCU <sip:andrei@cs.utt.ro>;tag=17462311	Copiat din cerere. Include un cod unic de identificare
Call-ID: 2386390012@mihai_ws.rusoaie.com	Copiat din cerere
CSeq: 1 INVITE	Copiat din cerere
Content-Type: application/SDP	
Content-Length: 200	
	Linia goală semnifică terminarea antetului SIP şi începerea conţinutului
v=0	Versiunea SDP
o=Andrei 4858949 4858949 IN IP4 192.1.2.3	Iniţiatorul sesiunii, versiunea sesiunii şi adresa
s=Prânz	Subiectul sesiunii
c=IN IP4 workstation.cs.utt.ro	Informaţii despre conexiune
m=audio 5004 RTP/AVP 0 3	Descrierea fluxurilor de sunet/imagine pe care destinatarul este pregătit să le accepte

3.3. Comparaţie între H.323 şi SIP

Cei care au creat standardul SIP susţin că, deoarece H.323 a fost iniţial conceput pentru reţelele ATM şi ISDN, nu se potriveşte sută la sută pentru sistemele VoIP, deşi acesta are multe caracteristici în plus. De asemenea H.323 are lipsuri în extensibilitatea necesară protocolului VoIP. SIP a fost creat pentru a fi folosit în Internet, şi pentru a evita complexitatea, având nevoie şi de o mai mare extensibilitate.

Protocolul SIP reutilizează majoritatea header-elor folosite, reguli de codificare, coduri şi mecanisme de eroare ale HTTP. H.323 defineşte sute de elemente, în timp ce SIP are doar 37 de headere, fiecare cu un număr mic de parametri.

H.323 foloseşte o reprezentare binară a mesajelor sale, care sunt pe standardul ASN.1, în timp ce SIP foloseşte codificarea mesajelor în text, similară cu http.

H323 nu este foarte scalabil şi a fost proiectat pentru o singură reţea şi, deci, are unele probleme de în scalare, cu toate ca în unele versiuni mai noi se folosesc tehnici care ocolesc aceasta problemă.

H.323 este limitat când se execută un loop detection în căutările multi-domain. Această problemă se poate rezolva păstrând mesajele, însă această tehnică nu este scalabilă. Pe de altă parte, SIP foloseşte o metodă de detecţie căutând în history-ul headerelor mesajelor.

Avantajele SIP sunt susţinute de IETF, care reprezintă unul dintre cele mai importante standarde, iar avantajul H323 este dat de o mai largă răspândire a sa. Tabelul 3.5 afişează diferenţele dintre cele două protocoale.

Tabelul 3.5

Diferenţele dintre protocoalele H.323 şi SIP

H.323	SIP
Protocol complex	Relativ simplu
Reprezentare binară pentru mesaje	Reprezentare text
Necesită compatibilitate cu versiunea precedentă	Nu necesită compatibilitate cu versiunea precedentă
Nu este foarte modular	Foarte modular
Nu foarte scalabil	Foarte scalabil
Semnalizare complexă	Semnalizare simplă
Mare răspândire pe piaţa	Susţinut IETF
NU suportă IM	Suportă IM
Inter domain routing: static	Inter domain routing: dinamic
Fără identificare de nume	Identificare de nume
Adresare flexibilă	Recunoaşte doar tipul URL de adresare
Sute de elemente	Doar 37 de headere

3.4. Alte protocoale

3.4.1. Media Gateway Control Protocol (MGCP)

Media Gateway Control Protocol (MGCP) este un protocol utilizat în cadrul implementărilor de tip Voce pe IP (VoIP). MGCP este definit în mod informațional (nestandardizat) în documentul IETF RFC 3435, care actualizează definiția anterioară din RFC 2709. MGCP înlocuiește Simple Gateway Control Protocol (SGCP).

Protocolul standard care servește aceluiași scop este Megaco (H.248), definit în RFC 3015. Totuși, Megaco nu este un protocol foarte răspândit în acest moment (Marie 2006), dar pare să câștige din ce în ce mai mult teren în cadrul arhitecturii NGN.

Utilizarea cea mai întâlnită pentru MGCP se găsește în cadrul arhitecturii de televiziune prin cablu, pentru servicii de Voce pe IP sau Video la Cerere (VoD).

MGCP este un protocol folosit în cadrul sistemelor de Voce pe IP distribuite, care văzute din exterior par un singur dispozitiv. Spre deosebire de alte protocoale VoIP, precum SIP sau H.323, MGCP-ul are o arhitectură de tip stăpân – sclav (în eng.: master – slave).

Sistemul este compus din două entități:

- Media Gateway Controller (MGC), numit și Call Agent (CA), care joacă rolul de stăpân;
- Media Gateway (MG) care joacă rolul de sclav.

Media Gateway (MG) – Element de rețea care transformă informațiile între rețelele cu comutație de circuite (PSTN) – trunchiuri sau bucle locale – și Internet (ori alte tipuri de rețele de date cu comutație de pachete).

Asigură, de asemenea, conversia semnalelor audio între cele două tipuri de rețele, sau, medierea transmisiilor între dispozitive care nu au un codec comun.

Câteva exemple de gateway-uri:

- gateway-uri de trunchiuri;
- gateway-uri de voce pe ATM;

- gateway-uri rezidențiale;
- gateway-uri de acces.

Media Gateway Controller (MGC) – Element de rețea care gestionează înregistrarea, administrarea cât şi controlul funcționalității resurselor unuia sau mai multor Media Gateway. Colectează informațiile despre desfăşurarea evenimentelor şi le le pune la dispoziția sistemelor de administrare şi plată.

3.4.2. **Session Announcement Protocol (SAP)**

Acest protocol este folosit pentru a anunța conferințele multicast şi alte operațiuni multicast. Un emițător SAP transmite periodic un pachet de date unei binecunoscute adrese multicast şi port (9875). Un receptor SAP foloseşte protocolul multicast Scope Zone Announcement Protocol şi ascultă adresa cunoscută pentru multicast şi port. Nu există mecanisme de întâlnire. Astfel un emițător nu ştie de existența sau absența unui receptor. Mesajele multiple pot anunța aceeaşi sesiune. Perioada de timp dintre repetările unui anunț se aleg astfel încât lățimea de bandă totală folosită de toate mesajele pentru un singur grup SAP să rămână sub limita prestabilită. Fiecare emițător va asculta celelalte mesaje pentru a determina numărul de sesiuni amânate într-un singur grup. Pentru a reduce delay-ul dintr-un grup se recomandă folosirea unui server de proxy pentru cache.

Serverul de proxy SAP ascultă toate grupurile SAP cu scopul de a menține o listă cu toate sesiunile prezente precum şi cu momentul în care fiecare mesaj s-a recepționat. SAP conține, de asemenea, mecanisme pentru asigurarea integrității mesajelor pentru anunțul sesiunilor, pentru autentificarea originii anunțurilor şi pentru criptarea acestora.

Capitolul 4

CALITATEA SERVICIILOR (QoS) ÎN VoIP

Şi în acest caz, ca de altfel în cazul oricărui serviciu în timp real, VoIP necesită ca reţeaua să ofere performanţe previzibile în parametrii de transport prevăzuţi de furnizorul de servicii.

4.1. Măsurarea calităţii vocii

4.1.1. Mean Opinion Score (MOS)

Mean Opinion Score (MOS) este cel mai răspândit mod de testare a calităţii vocii. Este o metodă subiectivă pentru determinarea calităţii. Există două metode: testul de conversaţie şi testul de ascultare. Subiecţii testului judecă calitatea vocii, fie dintr-o conversaţie, fie ascultând eşantioane de voce, iar apoi evaluează calitatea vocii folosind următoarea scală:

5 – excelent, 4 – bună, 3 – acceptabilă, 2 – slabă, 1 – foarte slabă

MOS este obţinut calculând media notelor date de subiecţii testului. Folosind acest sistem, la o medie de 4 sau superioară, calitatea este considerată ca fiind acceptabilă.

MOS a fost creat inițial pentru a compara diverşi algoritmi de codare (vezi Tabelul 4.1).

Tabelul 4.1

MOS pentru diverse CODEC-uri

Standard de codare	MOS
G.711	4,3 – 4,4 (64 kbps)
G.726	4,0 – 4,2 (32 kbps)
G.728	4,0 – 4,2 (16 kbps)
G.729	4,0 – 4,2 (8 kbps)
G.723.1	3,8 – 4,0 (6,3 kbps)
	3,5 (5,3 kbps)

MOS este considerat cel mai relevant test pentru că reţeaua de voce este folosită de oameni, şi, deci, părerea lor este prioritară față de orice alte teste. Cu toate acestea, fiind un test subiectiv, ce foloseşte subiecți umani, poate fi un mare consumator de timp în administrare.

4.1.2. **Perceptual Speech Quality Measure (PSQM)**

PSQM (Perceptual Speech Quality Measure) foloseşte un model psihoacustic pentru a determina matematic diferențele dintre semnalul de intrare şi cel de ieşire (vezi figura 4.1).

Folosind această metodă, dacă semnalele de intrare şi de ieşire sunt identice, scorul PSQM va fi zero. Cu cât sunt mai mari diferențele dintre aceste semnale cu atât este mai mare scorul PSQM, maximul fiind de 6,5. Cu toate acestea, spre deosebire de măsurătorile tradiționale în cazul raportului semnal–zgomot, PSQM pune accent pe diferențele din voce care vor afecta percepția umană asupra calității vocii.

Una dintre criticele aduse la adresa PSQM este că a fost proiectat inițial pentru a măsura calitatea standardelor de codare. Astfel, el nu ia în considerare efectele anumitor erori de transmisie. PSQM+ a fost propus în decembrie 1997 şi poate gestiona

percepții diferite datorită volumului sau a unor distorsiuni gălăgioase precum şi voce cu întreruperi. Cu ajutorul PSQM+ corelația dintre scorul obiectiv obținut şi MOS este îmbunătățită.

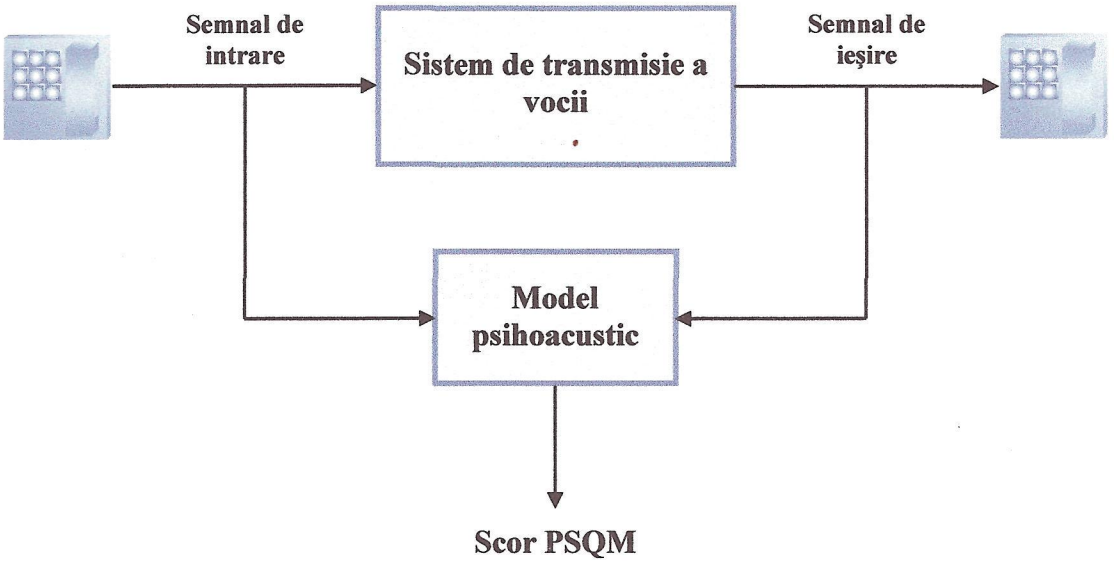

Fig. 4.1 – **Scorul PSQM**

4.1.3. Alte metode de măsurare a calității vocii

Există şi alte măsurători obiective care au fost propuse sau chiar se folosesc. Dintre acestea, putem aminti: MNB (Measuring Normalizing Blocks), PAMS (Perceptual Analysis Measurment System) dezvoltat de British Telecom, precum şi PESQ (Perceptual Evaluation of Speech Quality).

4.1.4. Caracteristici de transmisie

Într-o rețea VoIP slăbiciunile transmisiei joacă un rol foarte important în determinarea calității vocii. Aceste slăbiciuni cuprind pierderea de pachete, latența şi jitter-ul. Altă perspectivă asupra testării calității vocii este de a măsura direct aceste slăbiciuni ale liniilor de transmisie şi de a prevedea ce măsuri trebuie luate.

Modelul „E" (E-Model), aşa cum este el descris în G.107 de către ITU-T, furnizează un model matematic pentru o analiză predictivă. Ecuaţia de bază este:

R = Ro – Is – Id – Ie + A,

unde:

R – factorul ratei de transmisie (*transmission rating factor*);

Ro – raportul semnal-zgomot (*basic signal-to-noise ratio*) este obţinut din toate sursele de zgomot din circuit;

Is – factorul de slăbiciune simultan (*simultaneous impairment factor*) apare în cazul în care sidetone-ul[1] nu e optim, precum şi a distorsiunilor cuantice;

Id – factorul de slăbiciune la întârzieri (*delay impairment factor*) este cauzat de delay-uri în reţea;

Ie – factorul de slăbiciune al echipamentelor (*equipment impairment factor*) este cauzat de codoare care lucrează cu o bandă îngustă, ca, de altfel, şi efectul pierderilor de cadre de partea codorului;

A – factor de expectanţe (*expectation factor*) este un factor de corecţie care ajustează calitatea percepută în funcţie de dorinţele utilizatorului. De exemplu, dacă utilizatorii sunt conştienţi că comunică cu o locaţie greu de atins prin multe conexiuni via satelit, vor fi mult mai cooperativi în a tolera slăbiciunile reţelei datorate întârzierilor mari.

De exemplu, o dată ce slăbiciunile unei reţele IP au fost măsurate, modelul „E" poate fi folosit pentru a calcula factorul ratei de transmisie. Acesta poate fi transformat în MOS folosind următoarele ecuaţii:

- **For R < 0:**
  MOS = 1

- **For 0 < R < 100:**
  MOS = 1 + 0.035R + 7R(R-60)(100-R) x 10 - 6

- **For R > 100:**
  MOS = 4.5

[1] **Sidetone,** în telefonie, este efectul sunetului preluat din microfon şi reprodus în casca aceluiaşi receptor pentru a da impresia că receptorul funcţionează.

4.1.5. **Metoda optimă de măsurare a calității vocii**

După ce am parcurs metodele de măsurare a calității, putem spune că, în practică unele din aceste metode pot fi folosite combinate. După cum am menționat şi mai sus, MOS este metoda cea mai relevantă pentru că în cazul ei factorul uman contează cel mai mult. MOS trebuie folosit întotdeauna ca şi o verificare, pentru că este cel mai apropiat de realitate. În loc să se testeze folosind testul MOS, este mai bine să se testeze rețeaua VoIP direct cu un grup selectat de useri, aceştia furnizând rezultate mult mai apropiate de realitate. După acestea, când se configurează sistemul, se mai pot face ajustări pe baza feedback-ului utilizatorilor. În aceste cazuri, un test obiectiv ca de exemplu PSQM, PAMS sau PESQ poate fi mai favorabil. În VoIP, QoS este o componentă foarte importantă. În măsurarea efectivității QoS, măsurarea slăbiciunilor transmisiei (pierderi de pachete, întârzieri şi jitter) este mult mai utilă, deoarece ajută în mod direct la a răspunde la mai multe întrebări:

- Dacă rețeaua este aglomerată, pachetele ce conțin voce au prioritate față de pachetele de date?
- Care este întârzierea medie a pachetelor de voce?
- Care este media jitter-ului al pachetelor de voce?

În testarea unei rețele VoIP este necesară crearea unui mediu realist. De obicei, acest lucru înseamnă că există mai multe apeluri simultane şi că fluxul de voce este în permanentă competiție cu fluxul de date pe aceeaşi lățime de bandă.

4.2. **Minimalizarea întârzierilor (latenței)**

Întârzierile (în eng. delay), adesea numite şi latență, este perioada de timp necesară unui pachet să traverseze rețeaua, de la terminalul sursă până la terminalul de la destinație. Cu alte cuvinte, latența este timpul necesar ca vocea celui care vorbeşte să ajungă la urechea ascultătorului. O latență mare nu deteriorează neapărat calitatea vocii

dintr-un apel telefonic, dar poate cauza o desincronizare între vorbitori. Acest lucru poate genera ezitări în timpul convorbirii şi poate duce la dificultăţi în a interacţiona.

Tipurile de delay care pot apărea într-o reţea ce oferă servicii multiple sunt prezentate în figura 4.2.

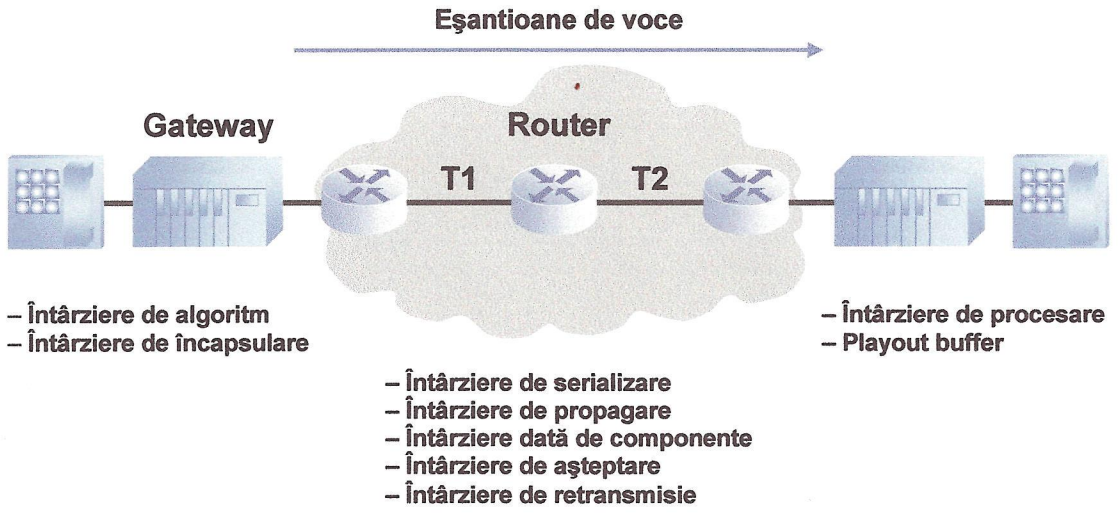

Fig. 4.2 – **Sursele întârzierilor dintr-o reţea VoIP**

Întârzierile introduse de codec-curi, numite *întârzieri de algoritm* (*Algorithmic Delay*), sunt datorate algoritmului de codare a vocii. În tabelul 4.2 se prezintă delay-urile introduse de cele mai comune codec-uri.

Tabelul 4.2

Întârzierile introduse de codec-urile folosite în mod curent

CODEC	Întârziere introdusă (ms)
G.711	0,125
G.726	1
G.728	3–5
G.729	15
G.723.1	37,5

Întârzierile datorate timpului necesar ca terminalele să creeze pachetele folosite în serviciile de voce sunt cunoscute sub numele de *întârzieri de încapsulare* (*packet creation latency* sau *packetization delay*). În RTP, eşantioanele de voce sunt, de obicei, acumulate înainte de a fi puse într-un cadru de transmisie pentru a reduce dimensiunea datelor suplimentare necesare la nivelul fizic al reţelei. Standardul RFC 1890 specifică că timpul de creare a pachetelor ar trebui să fie de 20 ms. În cazul G.711 acest lucru se traduce prin faptul că 160 de eşantioane de voce sunt acumulate şi mai apoi transmise într-un singur cadru de date. Pe de altă parte, G.723.1 generează un cadru de voce o dată la 30 ms şi fiecare cadru este transmis de obicei ca un singur pachet RTP.

Întârzierile generate de timpul necesar pentru a serializa informaţia digitală pentru a fi transmisă pe legătura fizică de interconectare a echipamentelor se numesc *întârzieri de serializare* (*serialization delay*). În tabelul 4.3 sunt redate întârzierile de serializare, în funcţie de lăţimea de bandă şi dimensiunea pachetului

Tabelul 4.3

Întârzieri de serializare în funcţie de lăţimea de bandă şi dimensiunea pachetului

Lăţime de bandă	Dimensiunea pachetului					
	64 octeţi	128 octeţi	256 octeţi	512 octeţi	1024 octeţi	1500 octeţi
56 kbps	9 ms	18 ms	36 ms	72 ms	144 ms	214 ms
64 kbps	8 ms	16 ms	32 ms	64 ms	128 ms	187 ms
128 kbps	4 ms	8 ms	16 ms	32 ms	64 ms	93 ms
256 kbps	2 ms	4 ms	8 ms	16 ms	32 ms	46 ms
512 kbps	1 ms	2 ms	4 ms	8 ms	16 ms	23 ms
768 kbps	0,64 ms	1,28 ms	2,56 ms	5,12 ms	10,24 ms	15 ms

De exemplu, dacă este folosit codec-ul G.711 şi perioada de pachetizare este de 20 ms (sunt 160 bytes în încărcătura RTP), atunci, în cazul încapsulării PPP, întregul cadru va avea 206 bytes. Pentru a-l transmite, va avea nevoie de 1,1 ms în cazul unei linii T1[1], 3,2 ms în cazul unei lăţimi de bandă de 512 kbps şi 25,8 ms la 64 kbps.

[1] **T1** este o schemă de semnalizare în telecomunicaţii, folosită în cazul circuitelor DS1 (Digital Signal 1) în America de Nord şi Japonia (corespondentul său fiind E1). Capacitatea suportată de circuitele DS1 este de 1.544 Mbps, canal de transmisie sincron.

Întârzierea la serializare apare de fiecare dată când informaţia trece printr-un echipament de tip „stochează şi trimite" (store–and–forward) ca de exemplu un router sau un switch. Astfel, un cadru de date ce trece prin 10 routere va acumula 10 întârzieri.

Timpul necesar ca un semnal electric (sau fotonic) să parcurgă lungimea unui conductor este cunoscut sub numele de *întârziere de propagare* (*propagation delay*). Acest timp de întârziere depinde de distanţa geografică dintre sursă şi destinaţie. Viteza de propagare într-un cablu este între 4 şi 6 milisecunde pe kilometru. În cazul transmisiilor prin sateliţi, întârzierile se situează în jurul a 110 ms la un satelit de la 14.000 km altitudine şi, respectiv, de 260 ms la un satelit de la 36.000 km altitudine. Exemplu: pentru a calcula delay-ul de propagare a unei fibre de 6.000 de Km, se foloseşte formula:

delay de propagare = 6.000 km / (299.300 km/s x 0,6) = 0,0334 secunde

Întârzierile date de componentele reţelei (*component delay*) sunt cauzate de diversele echipamente din sistemul de transmisie. De exemplu, un cadru de date ce trebuie să treacă printr-un router, trebuie să ajungă de la portul de intrare la cel de ieşire prin intermediul plăcii de interconectare (*backplane*). Există un delay minim din cauza acestei plăci, precum şi datorită cozilor de aşteptare din routere.

Întârzierea cauzată de timpul în care un pachet rămâne în zona tampon a unui echipament de reţea, aşteptând retransmiterea, este cunoscută sub denumirea de *întârziere de aşteptare* (*queuing delay*).

Timpul necesar unui echipament de reţea (router, switch, firewall etc.) pentru a procesa un anumit pachet şi a lua o decizie de trimitere mai departe, generează *întârzierea de retransmitere* (*packet forwarding delay*).

La proiectarea unei reţele care trebuie să furnizeze mai multe servicii, delay-ul total acumulat de semnal sau pachet este suma tuturor latenţelor. În general, este acceptat că pentru convorbiri telefonice de calitate, latenţa totală dintr-o reţea trebuie să fie mai mică de 150 ms.

Să presupunem că dorim să nu depăşim o întârziere de 150 ms. Pentru a realiza acest lucru, calculăm întârzierile fixe ce apar pe reţea:

G.723.1 (întârziere de algoritm)..... 37,5
G.723.1 (întârziere de procesare)..... 30,0
Întârziere de serializare (2 T1)..... 2,0
Întârziere de propagare (5.000 km de fibră optică)..... 25,0
Alte întârzieri..... 2,0
Total întârzieri fixe:..... 96,5

În acest caz, limita întârzierilor variabile ce pot să apară în rețea este de:

150 ms – 96,5 ms = 53,5 ms

Cel mai important efect al întârzierilor este ecoul. Acesta poate apărea într-o rețea de voce datorită legăturii defectuoase dintre cască şi microfon. Acesta este cunoscut sub numele de ecou acustic. Mai poate apărea şi când o parte din energia electrică este reflectată înapoi spre ascultător de către circuitul hibrid din cadrul rețelei PSTN. Acesta este cunoscut sub denumirea de ecou hibrid.

Când delay-ul de la un capăt la celălalt capăt al rețelei este scurt, orice ecou generat de circuitul de voce, se va întoarce spre vorbitor foarte repede şi nu va fi observat. De fapt, anularea ecoului nu este necesară dacă întârzierile de propagare pe un sens al comunicației nu depăşesc 25 ms. Cu alte cuvinte, dacă ecoul se întoarce în 50 ms, el nu va fi observat. Cu toate acestea, deoarece într-o rețea VoIP întârzierile aproape întotdeauna depăşesc 25 ms, este necesar să fie luate măsuri de anulare a ecoului.

4.3. Impactul jitter-ului în calitatea vocii

Jitter-ul (sau diferența de întârziere) este diferența de timp dintre momentul când un pachet este aşteptat să ajungă şi momentul în care chiar ajunge la destinație. Jitterul se manifestă, în general prin sacadări sau întreruperi ale fluxului de date de intrare. Cu alte cuvinte, dat fiind un flux constant de pachete la fiecare 20 ms, noile pachete sunt aşteptate să ajungă la destinație exact la 20 ms. Din păcate, după cum se vede şi din figura 4.3, de cele mai multe ori nu se întâmplă aşa. În figură pachetul unu (P1) şi pachetul trei (P3) ajung la destinație la momentul potrivit, dar pachetul doi (P2) ajunge cu 12 ms întârziere față de momentul dorit, iar pachetul patru (P4) ajunge cu 5 ms întârziere.

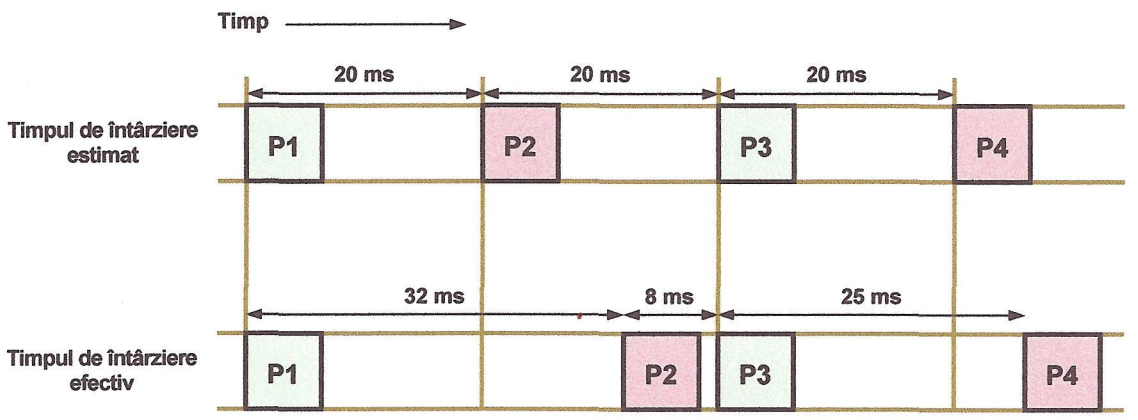

Fig. 4.3 – **Diferențe de întârziere (jitter)**

Într-o rețea de servicii complexe, jitter-ul este cauzat, de cele mai multe ori, de variații ale cozii de așteptare, din cauza schimbărilor dinamice ale încărcării rețelei. Jitter-ul apare și atunci când unul sau mai multe pachete urmează o cale diferită – de cost egal, dar care nu este de aceeași lungime – de calea parcursă de celelalte pachete de voce.

Majoritatea gateway-urilor de voce au un buffer (*playout buffer*) care păstrează pachete din fluxul de date, astfel încât vocea reconstituită să nu fie afectată de jitter. Cu cât jitter-ul este mai mare cu atât mai mult timp unele cadre vor fi reținute în buffer, ceea ce mărește întârzierea.

Pentru întârzierile din exemplul prezentat în subcapitolul 4.2, jitter-ul maxim tolerat va fi de 53,5 ms. Pentru a rămâne în parametrii propuși, jitterul trebuie să fie eliminat de un buffer care trebuie să întârzie cadrele cu 53,5 ms, eliminând în acest fel jitter-ul.

Cu toate că un anumit jitter este normal să fie prezent, dacă acesta depășește anumite limite, poate afecta calitatea vocii pentru că gateway-ul de voce poate scăpa unele pachete care nu ajung la timp. În aceste condiții gateway-ul poate introduce spații în vocea reconstituită.

4.4. Calculul lățimii de bandă necesară

Operatorii pot calcula lățimea de bandă necesară pentru a face față traficului de voce cu ajutorul câtorva calcule simple. Decizia asupra lățimii de bandă alocate pentru fiecare serviciu într-o rețea de date şi voce necesită o atenție sporită din partea furnizorului de servicii. Alocarea unei lățimi de bandă prea mici poate genera o calitate inacceptabilă.

Printre factorii care influențează decizia împărțirii benzii sunt:

- Impactul lățimii de bandă: traficul de voce primeşte, de obicei, prioritate față de traficul de date datorită sensibilității ridicate ale pachetelor VoIP;
- Compromisul dintre compresie şi calitatea vocii: operatorii pot folosi un codec pentru a comprima pachetele VoIP şi a reduce lățimea de bandă folosită. Cu toate acestea, compresia scade per ansamblu calitatea convorbirii, forțând operatorii să își echilibreze lățimea de bandă utilizată pentru a lua în calcul indicatorii de calitate;
- Utilizarea maximă planificată (projected peak use): operatorii alocă lățime de bandă bazată pe estimarea de apeluri în timpul orelor de vârf. Subestimarea lățimii de bandă necesară poate degrada calitatea vocii. Operatorii trebuie, de asemenea, să aloce lățimea de bandă corespunzătoare semnalizării, pentru a se asigura că apelurile de voce se termină cu succes, şi pentru a evita întreruperea serviciilor.

Formula de calcul a lățimii de bandă este:

Biți pe secundă = **Eşantioane pe secundă** x **Dimensiunea unui pachet** x x **Numărul de convorbiri** x **8 biți pe secundă**

unde:

Eşantioane pe secundă = 1.000 ms / Rata de creare a pachetelor

De exemplu, pentru a calcula lățimea de bandă necesară susținerii a 2.000 de convorbiri full-duplex encodate cu codec-ul G.711 care au o rată de creare a pachetelor de 20 ms şi o dimensiune a unui pachet de 200 octeți (40 octeți are antetul IP + 160 octeți au informațiile utile), se fac următoarele calcule:

Eşantioane pe secundă = 1000 ms / 20 ms = 50

Lățime de bandă (biți/s) = 50 eşantioane pe secundă x **200 octeți** x
x **2.000 apeluri** x **8 biți/s = 160 Mbiți/s**

Acest rezultat este unul brut şi include doar traficul efectiv de voce, nu ia în considerare antetele adăugate de mediile de transport (routere) şi protocoalele de la nivelul legătură de date. Lățimea de bandă necesară pentru semnalizare depinde de viteza cu care apelurile sunt generate şi de protocolul de semnalizare folosit. Dacă sunt inițiate într-un timp relativ scurt, lățimea de bandă necesară pentru semnalizare va fi destul de mare. În general, lățimea de bandă necesară pentru semnalizare într-o rețea IP este de aproximativ 3% din totalul traficului brut din rețea. Folosind exemplul anterior, pentru semnalizarea a 2.000 de apeluri inițiate într-o secundă, este nevoie de aproximativ 4,8 Mbps (3% din 160 Mbps). Astfel, pentru a suporta traficul brut şi traficul de semnalizare generat de 2.000 de apeluri encodate cu G.711 este nevoie de o lățime de bandă de aproximativ 164,8 Mbps. Această lățime de bandă este calculată pentru acest caz specific. Orice schimbare a codec-ului, a compresiei sau a vitezei de creare a pachetelor duce la schimbarea necesarului de bandă.

4.5. Compensarea pierderilor de pachete

În cazul aglomerării rețelei, router-ele şi switch-urile pot depăşi limita zonei tampon (buffer) şi pot fi forțate să ignore pachete. Spre deosebire de aplicațiile unde pierderea de pachete nu este critică (real-time), pentru VoIP este o problemă serioasă. UDP nu permite retrimiterea pachetelor, şi chiar dacă ar retransmite pachetele, acestea nu ar ajunge la destinație în timp util pentru reconstrucția vocii, rezultând un delay semnificativ. Într-o sesiune RTP, în cele mai multe cazuri, gateway-ul primeşte datele restransmise mult prea târziu ca să mai poată reconstitui vocea. Terminalele, la rândul lor, trebuie să se confrunte cu eşantioane de voce lipsă, deci ultimul cuvânt asupra calității vocii îl au de spus terminalele.

În cel mai simplu caz, dacă un eşantion lipseşte, terminalul lasă pur şi simplu un spațiu în fluxul de voce. Dacă sunt pierdute prea multe pachete, vocea va fi sacadată, cu silabe sau cuvinte întregi lipsă. O posibilă strategie de recuperare a porțiunii de voce

lipsă este de a reda din nou ultimul eşantion de voce recepţionat de la sursă. Această metodă funcţionează bine doar dacă lipsesc doar câteva eşantioane de voce. Pentru a acoperi erorile, de obicei se foloseşte interpolarea: pornind de la eşantioanele anterioare de voce decodorul va prezice care sunt eşantioanele lipsă. Această tehnică este cunoscută sub denumirea de „mascarea pierderilor de pachete" (Packet Loss Concealment – PLC).

De exemplu, în specificaţiile ITU-T pentru codec-ul G.771, anexa I descrie un algoritm PLC pentru PCM. Ultimele eşantioane de voce sunt reţinute în permanenţă într-un buffer circular de 48,75 ms. În momentul în care este detectată o lipsă de pachete, conţinutul buffer-ului este folosit pentru a estima conţinutul porţiunii de timp libere. Cu ajutorul PLC, în G.711 ieşirea audio este întârziată cu încă 3,75 ms pentru a obţine o tranziţie lentă între semnalul real şi cel sintetizat. Codec-urile bazate pe CELP[1] ca de exemplu G.723.1, G.728, G.729 au şi ele algoritmi PLC implementaţi. În general, dacă segmentele de erori nu sunt prea mari, şi semnalul nu se schimbă foarte rapid, eşantioanele lipsă nici nu se simt după reconstrucţia finală a vocii.

În anexa I, a specificaţiilor protocolului G.113, se prezintă estimări ale impactului pierderilor de pachete asupra calităţii vocii. Această influenţă este calculată în Ie, factor care desemnează slăbiciunea echipamentului (*impairment factor*). Acest factor este deviat subiectiv din MOS. Ie este un număr întreg, valoarea 0 este asociată unui echipament perfect (vezi tabelul 4.4).

Tabelul 4.4

Impactul pierderilor de pachete asupra factorului Ie

Codec	Ie (0% pierderi)	Ie (2% pierderi aleatoare de cadre)	Ie (5% pierderi aleatoare de cadre)
G.711 fără PLC	0	35	55
G.726 cu PLC	0	7	15
G.729A	11	19	26*)
G.723.1 (6,3 kbps)	15	24	32*)

*) Valorile sunt pentru de 4% pierderi aleatoare de cadre. Valorile pentru 5% nu au fost specificate în anexă.

[1] **CELP** – *Code Excited Linear Prediction* este un algoritm de codare a vocii ce aplică o predicţie liniară în reconstrucţia vocii.

Pentru G.711, Ie este de 35, atunci când rata de pierdere a pachetelor este de 2%. Cu toate acestea, cu ajutorul PLC, Ie este redus la 7. De reținut că, în cazul unor codec-uri ce folosesc o rată de transfer mai scăzută, ca de exemplu G.729A şi G.723.1, Ie al echipamentelor este de 11 şi respectiv 15, chiar şi atunci când nu există pierderi de pachete. O pierdere de 2% ar creşte Ie la 19 şi respectiv la 24.

Cu toate că pierderile de pachete sunt nedorite, ele pot fi tolerate atâta timp cât sunt împărțite la un număr mare de utilizatori. Calitatea vocii, în general, nu este afectată în proporție de mai mult de 5% din numărul total de apeluri.

4.6. **Resource Reservation Protocol (RSVP)**

Protocolul RSVP este parte a arhitecturii IETF Integrated Services (IntServ), descrisă în RFC 1633-1944, care permite diferitelor echipamente să comunice cererile QoS. IntServ produce un protocol pentru semnalizarea cererilor QoS ale aplicațiilor şi încorporează specificații pentru descrierea cererilor de servicii şi alte funcționalități ale echipamentelor şi ale altor elemente de rețea care suportă QoS. IntServ este îndreptat spre suportul traficului de date video sau voce în timp real.

RSVP a fost creat pentru a semnala cererile de QoS peste conexiunile Internetului. Este un protocol bazat pe protocolul IP, hop-by-hop care informează echipamentele de pe o anumită cale a unui flux de date despre cererile pentru calitatea serviciului.

Obligația de bază a protocolului este să rezerve resurse astfel încât aplicația să poată primi cantitatea de lărgime de bandă dorită. În această încercare de rezervare de resurse, politica de management a lărgimii de bandă poate fi aplicată prin „policy objects", mai departe crescând eficiența alocării de bandă. În fond, RSVP este o metodă de a emula o rețea cu comutație de circuite peste o rețea cu comutație de pachete. Din cauza funcționării în IP, rețelele bazate pe acest protocol plasează responsabilitatea pentru menținerea stării conexiunii doar la echipamentele terminații ale rețelei, deci RSVP cere ca fiecare ruter intermediar să mențină informațiile despre starea conexiunii.

RSVP poate fi folosit pentru trafic unicast şi multicast. În cazul traficului multicast, un emiţător (sender) înaintează o copie a fiecărui pachet către mai mulţi receptori. O raţiune pentru crearea acestei legături bazate pe receptor – rezervări este multicast-ul. Acest tip de trafic adesea presupune cereri de rezervare de la diferiţi receptori cu caracteristici foarte diferite – şi, astfel, un asemenea de protocol este necesar pentru a suporta aceste cereri.

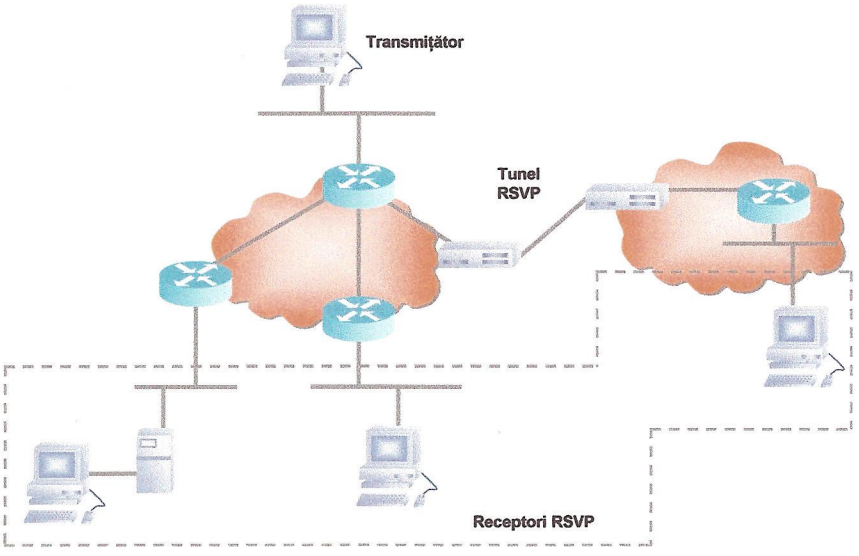

Fig. 4.4 – **În RSVP, informaţiile despre host-uri sunt expediate spre receptori prin intermediul fluxului de date**

Din cauza complexităţii fluxurilor multicast, rezervările trebuie să fie unificate. În traficul multicast, pachetele ce trebuiesc livrate spre diferite next-hop-uri sunt replicate. În RSVP, cererile de rezervare trebuie unificate la fiecare punct de replicare. Într-adevăr, cererile multiple de rezervare sunt combinate în aceste puncte de unificare şi apoi înaintate ca o singură cerere de rezervare. Aceasta are legătură cu conceptul de rezervări distincte şi rezervări multiple, ambele suportate de RSVP. În cazul rezervărilor distincte, o singură rezervare este făcută o dată pentru fiecare emiţător upstream. În cazul rezervărilor multiple, un număr de emiţători într-un flux multicast folosesc o aceeaşi rezervare. (Receptorul sau destinaţia e referit ca downstream, iar transmiţătorul e referit ca upstream.)

Există subcategorii de moduri de rezervare. În modul Wild-Card Filter (WF), o singură cerere de rezervare e generată şi împărţită între toate fluxurile de la toţi emiţătorii. În modul Fixed Filter (FF), fiecare emiţător generează câte o cerere distinctă

de rezervare. În modul Share Explicit (SE), mai mulți emițători selectați împart o aceeași cerere de rezervare. Aceste moduri de rezervare ajută la optimizarea rezervărilor de la diferiți emițători în același timp.

RSVP este executat printr-o serie de mesaje (figura 4.5). Un pachet standard RSVP are header-ul format din:

- câmpuri de 4 biți:

  Vers – arată numărul versiunii protocolului;

  Flags – câmp nespecificat.

- câmpuri de 8 biți:

  Reserved Field;

  Send TTL – indică time to live al mesajului trimis;

  Message Type – indică funcția mesajului.

- câmpuri de 16 biți:

  Checksum – suma de control standard TCP/UDP;

  Length Field – lungimea headerului și cea a obiectelor care urmează.

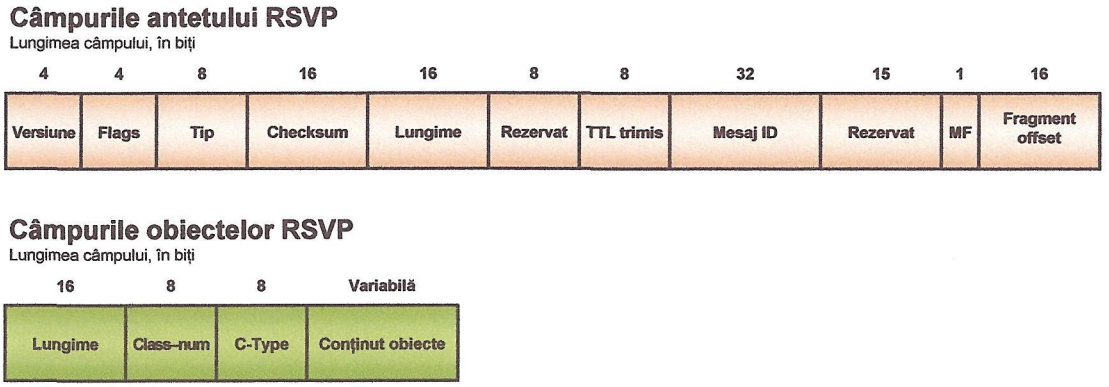

Fig. 4.5 – **Formatul pachetului RSVP, alcătuit din antetul mesajului și câmpuri de obiecte**

Antetul e urmat de un set de obiecte care includ informațiile necesare pentru descrierea și caracterizarea acelui mesaj particular. Aceste obiecte sunt împărțite în clase și fiecare clasă de obiecte poate conține mai multe tipuri ce caracterizează formatul datelor în detaliu mai mare.

Există 2 tipuri de RSVP: Nativ şi UDP–încapsulat. În cazul Nativ, header-ul şi încărcătura sunt încapsulate, cu numărul de protocol 46, în datagrame IP. RSVP UDP–încapsulat poate permite terminalelor să comunice cu primul şi ultimul hop, chiar dacă aceste terminale nu suportă RSVP.

În RSVP, un mesaj PATH este transmis de la emiţător la un receptor (sau un receptor multiplu). Mesajele PATH reţin informaţii de cale în fiecare nod străbătut de-a lungul căii – minim, un mesaj PATH conţine adresa IP a fiecărui hop de dinainte din calea străbătută. Aceste adrese IP stabilesc calea pentru transmiterea mesajelor „cerere de rezervare subsecventă" RESV.

Mesajul PATH conţine un obiect Sessiune care include adresa de destinaţie şi informaţiile de port şi un obiect „Previous Hop" (PHOP) care defineşte ruterul precedent în direcţia fluxului. În plus, mesajul PATH conţine obiectele Sender Template şi un Sender Traffic Specification (Tspec). Sender Template conţine specificaţii de filtru (Filter Specs) şi identifică formatul traficului pe care emiţătorul îl va trimite. Sender Tspec caracterizează fluxul de trafic pe care emiţătorul intenţionează să-l genereze, în sensul atributelor ca lărgime de bandă, jitter sau întârziere. Mesajul PATH mai poate conţine un obiect Adspec care descrie tipul serviciului, caracteristici ale performanţelor specifice serviciului şi suma resurselor disponibile pentru o rezervare particulară. Adspec e trimis către controllerul local de trafic din fiecare nod sau ruter de-a lungul căii pentru updatare.

Mesajul PATH mai poate conţine un obiect Policy Data care include informaţii despre sursa sau hop-ul precedent al unui flux de date. Procesul de control al politicii foloseşte date de politică pentru a stabili autorizări şi feed-back-uri pentru fluxul de date. În afară de primirea mesajelor PATH, ruterele ce suporta RSVP au o stare PATH care, la minimum reţin adresa IP a hop-ului precedent (PHOP). Aceasta informaţie este utilizată pentru a răspunde în direcţie inversă cu mesaje RESV. Starea PATH informează componentele reţelei de-a lungul căii despre celelalte noduri RSVP.

Receptorul face cererea de rezervare prin transmiterea înapoi a unui mesaj RSVP în direcţie inversă (spre sursa datei). Mesajul RESV include o specificaţie de rezervare Rspec, care specifică ce tip de serviciu este cerut (Guaranteed sau Controlled Load). Guaranteed Service oferă o conexiune ca un circuit virtual, pe când Controlled Load depăşeşte un serviciu „best-efort", dar nu e la nivelul Guaranteed Service.

Mesajul RESV include informații despre modurile de rezervare ca şi un FlowDescriptor care conține un obiect FlowSpec şi un obiect FilterSpec. Obiectul FlowSpec stabileşte parametrii QoS pentru procesul de programare (aranjare) a pachetelor. FilterSpec îndeplineşte aceeaşi funcție în procesul de clasificare a pachetelor.

În afara primirii mesajului RESV, ruterele şi nodurile de rețea RSVP-enabled au un proces de control al admisiei care indică care dintre noduri pot suporta cererile impuse de QoS. Dacă acest proces se încheie cu succes, mesajul RESV este trimis către următorul ruter. Ruterele ce cunosc RSVP de-a lungul căii organizează şi prioritizează pachetele în funcție de cereri. Aceste sisteme trimit pachetele de date ce vin către un clasificator de pachete, apoi sunt stocate într-o coadă de aşteptare a unui organizator de pachete. Un filtru de pachete asignează (mapează) pachetelor o anumită clasă de servicii, definind clase de rute şi QoS pentru aceste pachete. Organizatorul de pachete forțează alocarea de resurse şi selectează pachetele pentru transmisie.

După ce ultimul ruter de-a lungul căii garantează cererea, o confirmare de rezervare (ResvConf) notifică recepției că cererea a fost plină de succes. RSVP este un protocol soft-state aşa încât rezervările trebuiesc periodic reconfirmate. Această caracteristică ajută la schimbările în rutere şi schimbările în alcătuirea grupurilor de multicast.

Sesiunile se termină prin transmiterea unor mesaje ce anulează căile sau stările de rezervare în nodurile spre recepție. Mesajul PathTear sunt create de transmițători (sau de opțiunea de timed out al RSVP) în fiecare nod pe cale şi trimise către toți receptorii. Trimis de nodul care l-a creat, mesajul anulează starea căii în fiecare nod de-a lungul căii. Mesajele Path Tear anulează stările de rezervare în aceste noduri sau echipamente.

În cazul unei erori de control al emisiei, un mesaj de eroare este trimis către cel ce a trimis cererea. Dacă unul dintre nodurile rețelei nu poate executa starea PATH, el trimite către emițător un mesaj PathError (PathErr). Dacă o stare de rezervare nu poate fi invocată, atunci componenta trimite către emițător un mesaj Reservation Error (ResvErr).

RSVP poate fi combinat cu alte protocoale QoS şi tehnologii, ca de exemplu Differentiated Services (DiffServ) sau MPLS (Multiprotocol Label Switching). DiffServ marchează şi prioritarizează traficul, iar RSVP asigură resursele necesare pentru a

transmite acel trafic. Este, de asemenea, compatibil cu MPLS, adică MPLS este capabil de a asigna etichete în concordanță cu specificațiile RSVP Flowspec. Ca puncte negative, complexitatea şi sofisticarea RSVP scade performanța pe ruterele backbone-ului. Este simplu de folosit pe unele aplicații de complexitate mai scăzută, însă RSVP este adesea înlocuit de alte protocoale pe backbone-ul rețelei, ca de exemplu DiffServ.

4.7. Arhitectura serviciilor diferențiate (DiffServ)

DiffServ a fost creat pentru a oferi o modalitate mai simplă de a stabili clase de servicii diferențiate pentru traficul Internet.

În modelul IntServ, resursele sunt alocate pentru fluxuri individuale, ceea ce duce către limitări de scalabilitate. Traficul este divizat într-un număr mic de clase de trimitere mai departe (forwarding), iar resursele sunt alocate pe clase. Nivelurile de performanță dorite sunt atinse printr-o combinație dintre provisoning, prioritizare şi control al admisiei. Aceasta este în contrast cu alte tehnici, ca de exemplu rezervarea de resurse „cap la cap" (end-to-end), tehnică ce stă la baza RSVP.

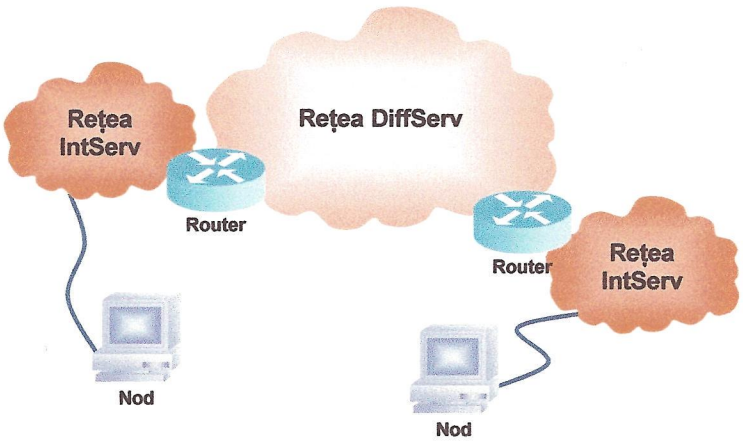

Fig. 4.6 – IntServ şi DiffServ

DiffServ a fost proiectat pentru a oferi servicii pentru mai multe clase agregate conținând mai multe fluxuri. Simplitatea este dată de faptul că aceste fluxuri sunt

grupate într-un număr relativ mic de agregate care primesc prin rețea un număr limitat de tratamente diferențiate (definite prin politici).

Unul dintre scopurile DiffServ a fost să elimine nevoia de stări de rezervare de resurse pentru fiecare flux, ca şi a semnalizările din fiecare router aflat de-a lungul căii. Mai toate clasificările şi politicile sunt făcute la marginea rețelei. În partea sa centrală, router-ele inspectează doar un câmp din header-ul pachetului IP – câmpul DiffServ – pentru a determina unde să trimită pachetul în continuare, ca diferență față de păstrarea informației pentru fiecare flux în parte (câmpul DiffServ se mai numeşte câmp DS sau octet DS).

Scopul suprem al arhitecturii DiffServ (figura 4.7) este de a simplifica înaintarea prin partea centrală a rețelei şi de a plasa spre marginea rețelei sarcinile de procesare ce apar odată cu clasificarea traficului. Această arhitectură este mai adecvată, decât multe alte variante posibile, pentru facilitarea nivelelor de scalabilitate cerute de rețelele din ziua de astăzi.

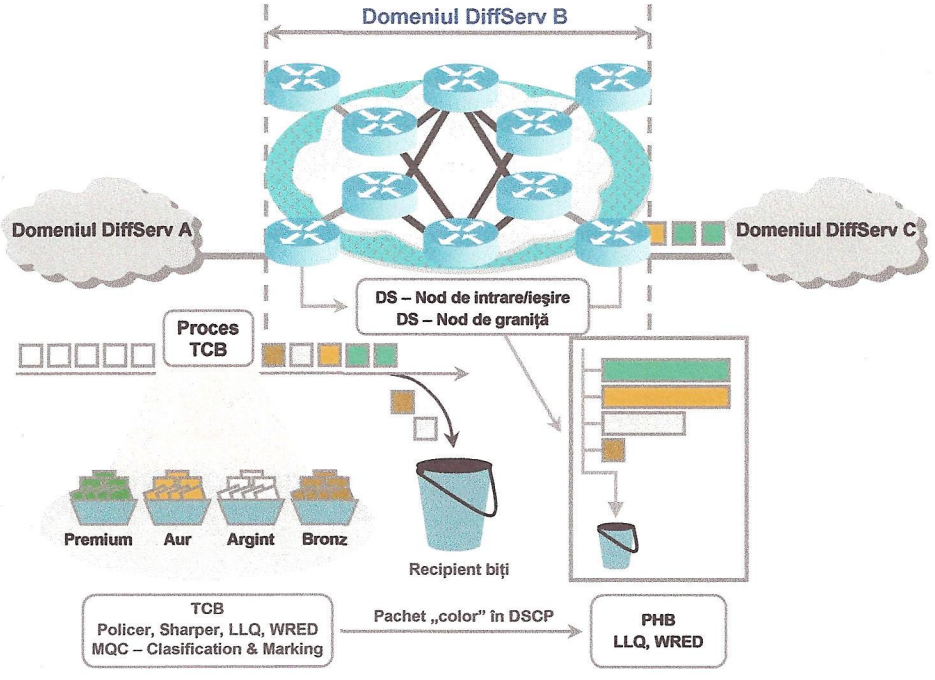

Fig. 4.7 – **Arhitectura serviciilor diferențiate (DiffServ)**

Atunci când traficul apare pe o interfață de intrare într-o rețea cu arhitectură DiffServ, acesta este clasificat şi supus unui proces de admisie preconfigurat, apoi este

făcut să îndeplinească cererile de politică în concordanţă cu o clasificare specifică. Apoi, fluxul de date este asignat unui comportament agregat. Aceasta se face prin marcarea câmpurilor IDS a headerelor pachetelor IP cu DSCP[1] corespunzător. Valoarea DSCP iniţiază un comportament per-hop (PHB – Per-Hop Behaviour) în componentele reţelei şi clasifică nivelul serviciului pentru pachetul respectiv. Termenul PHB specifică tratamentul specific de înaintare a unui pachet care apare într-un nod particular.

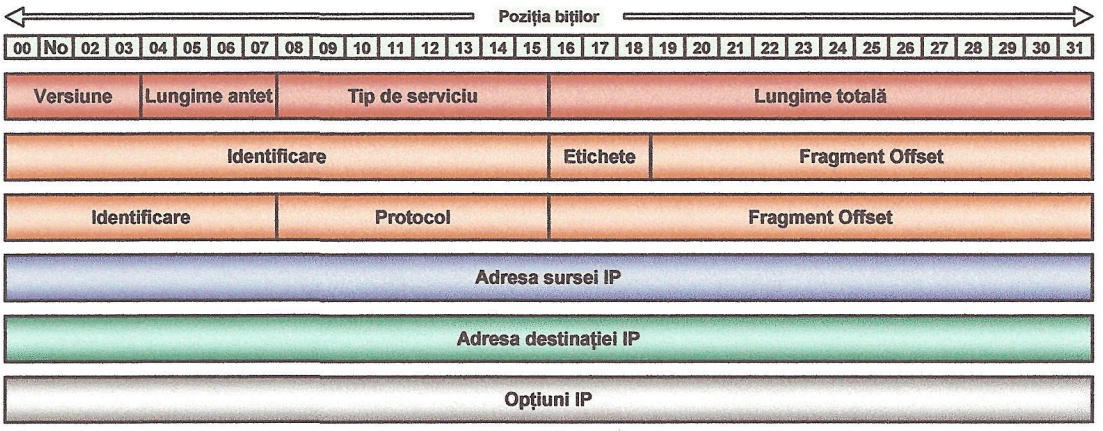

Fig. 4.8 – **Structura antetului unui pachet IP**

În DiffServ, câmpul Type of Service (ToS) din headerul IPv4 este înlocuit de câmpul DS care conţine o valoare pe care routerele DiffServ o foloseşte pentru a determina un PHB specific pentru fiecare nod aflat de-a lungul căii (fifurile 4.9 şi 4.10).

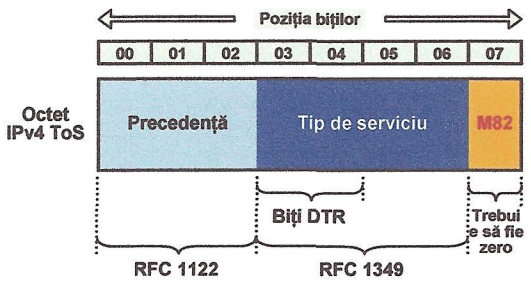

Fig. 4.9 – **Octet cu tipul serviciului în IPv4**

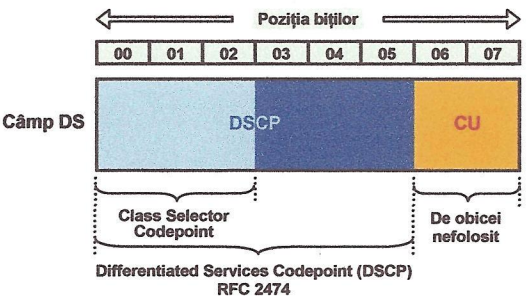

Fig. 4.10 – **Octet cu informaţiile DiffServ**

[1] DSCP – *Differentiated Services Code Point* – câmp în antetul pachetelor IP introdus cu scopul de a le clasifica

Primii 6 biți ai câmpului DS compun DSCP. Acesta este în concordanță cu PHB, care este recepționat de la fiecare pachet ce conține acest câmp la fiecare nod. Valoarea din câmpul DSCP se numește code points. Ultimii 2 biți din cadrul câmpului DS se numesc CU (Curently Unused) și suportă versiunile anterioare de echipamente ce nu cunosc DiffServ și care utilizează octetul ToS pentru a determina tratamentul de înaintare (forwarding). Câmpul DS poate avea până la 64 de valori.

Octetul DS este de asemenea ingredientul fundamental pentru asigurarea SLA (Service Level Agreement) între componenții unei rețele sau între rețelele propriu-zise. Mai exact, TCA (Traffic Conditioning Agreements) sunt definite pentru atingerea acestui scop. TCA poate include parametrii privind profilul traficului (întârzieri, lărgimi de banda, priorități etc.) și instrucțiuni privind tratamentul aplicat altor pachete.

Cele două procese fundamentale în implementarea DiffServ sunt clasificarea și condiționarea (figura 4.11). În DiffServ, traficul este condiționat și clasificat la frontiera rețelei, pe baza parametrilor TCA care sunt stabiliți între provider-ul de servicii și clientul rețelei.

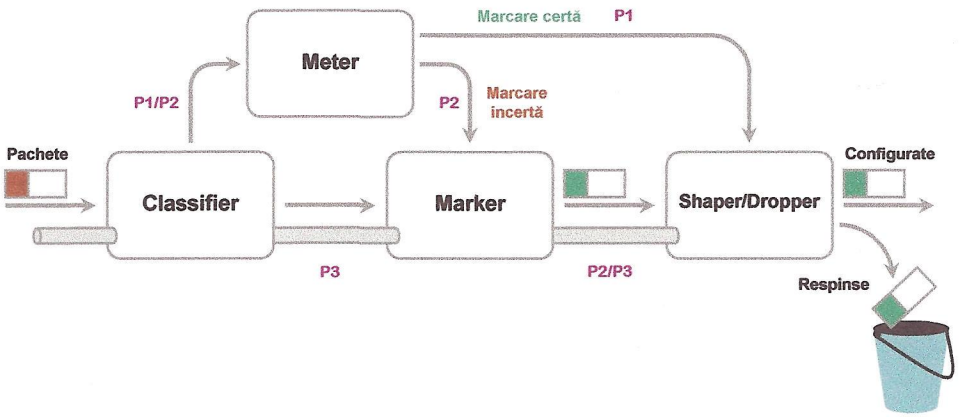

Fig. 4.11 – **Blocul de condiționare a traficului în serviciile diferențiate**

În DiffServ, un clasificator selectează pachetele în funcție de informația din headerul pachetului, corelat cu politicile stabilite pentru controlul admisiei. Există două tipuri de clasificatoare DiffServ: Behavior Aggregate (BA) și Multi Field (MF). Clasificatorul BA clasifică pachetele în funcție de valoarea DSCP din headerul pachetului. Clasificatorul MF clasifică pachetele după unul sau mai multe câmpuri din header, ceea ce permite o alocare a resurselor mult mai complexă decât cea oferită de

BA. Acesta permite o marcare a pachetelor bazată pe adresele sursă sau destinație, porturile sursă sau destinație, ID-ul protocolului, printre alte variabile.

Condiționarea implică metering, marking, shaping şi dropping.

Un meter monitorizează traficul după modul în care acesta este clasificat. El determină care tip de trafic are anumite caracteristici, şi, de asemenea, poate ține statistici despre trafic, statistici folosite în procesele de contabilizare şi facturare a serviciului oferit.

Markerul setează câmpul DS cu o valoare specifică. Pachetele pot fi marcate pentru un anumit flux pentru a ajuta la asigurarea secvenței corecte a PHB pentru acel flux. Markerul poate fi folosit pentru remarcări (de exemplu pentru schimbarea unei valori a octetului DS) sau pentru a şterge marcarea pentru pachetele ce nu mai corespund profilului. Pachetele ce ajung în diferite domenii trebuie remarcate repetat pentru a asigura proprietatea de a fi în concordanță cu profilul traficului din domeniul în care intră. Un domeniu este un set de noduri de rețea care operează după un set de politici şi definiții PHB comune.

Funcția shaper-ului este de a reține pachetele în cozi, pentru a asigura ca traficul să fie în profilul declarat, de multe ori reținând multe pachete până a le elibera din nou în rețea.

Atunci când un flux depăşeşte specificațiile de trafic, pachetele în exces pot fi pur şi simplu aruncate din acel flux. Alternativ, pachetele pot fi întârziate sau li se poate reduce nivelul de prioritate. Aceste măsuri sunt luate atunci când, de exemplu, un flux depăşeşte rata de transmisie negociată.

Un aspect cheie al implementării DiffServ este de a determina care dintre pachete vor primi prioritate la înaintare. Există două tipuri primare de înaintări: Expedited Forwarding (EF) şi Assured Forwarding (AF). EF oferă minim de întârzieri, jitter, pierderi de pachete şi bandă asigurată. În EF, rata de sosire a pachetelor la acel nod trebuie să fie mai mică decât rata de ieşire. Pachetele care nu corespund profilului de trafic sunt aruncate sau oferite în afara secvenței. EF este destinat aplicațiilor cu sensibilitate la întârzieri, ca traficul audio sau video. În AF, există 4 clase, fiecare conținând 3 proceduri de aruncare a pachetelor. Prioritățile de aruncare sunt date pentru a determina care pachete să fie aruncate în timpul perioadelor de congestie în rețea. Pachetele ce nu mai satisfac profilul sunt aruncate în conformitate cu procedurile. Pachetele cu prioritate mai mare sunt aruncate primele.

În DiffServ, SLA între clienţi şi service provider stabilesc criteriile de politică şi definesc profilurile de trafic. Traficul este clasificat şi condiţionat la interfaţa de intrare în reţea, însă dacă traficul trebuie să traverseze mai multe domenii, unele dintre aceste procese pot fi efectuate şi la interfeţele de ieşire din reţea sau în nodurile interioare ale reţelei. Internetul şi unele reţele de companii conţin mai multe domenii. Asigurarea lărgimii de bandă de-a lungul mai multor domenii este principala problemă dacă se doreşte un anumit nivel QoS cap–la–cap. Profilul traficului care traversează frontiera dintre două domenii se specifică în SLA care există între cele două domenii. Dar odată cu creşterea traficului între două domenii, creşte şi nevoia de a ajusta profilul traficului, ducând la nevoia unei alocări de resurse mult mai flexibilă.

Aici apare Bandwith Broker (BB). BB începe cu o setare cap–la–cap a legăturii cu alţi BB de-a lungul căii dorite, făcând negocieri de resurse prin mai multe domenii. BB performează control al admisiei, control de politici, agregări şi urmăriri de rezervări. BB poate facilita cererile de rezervare dintre reţeaua unei întreprinderi şi utilizatorii ei.

DiffServ are potenţialul de a suporta trafic multicast, dar unele estimări de trafic trebuiesc realizate înainte de folosirea lui pe o scară mare. Faptul că membrii grupurilor de multicast se pot schimba foarte repede face dificilă cunoaşterea cantităţii de trafic implicată într-o sesiune multicast – o nevoie pentru asigurarea calităţii unei transmisii multicast. În plus, un arbore de distribuţie multicast poate avea un punct de intrare şi mai multe puncte de ieşire, ceea ce complică estimarea de trafic.

În continuare, se depun eforturi pentru determinarea unei coabitări între DiffServ şi RSVP. Ideea este de a se utiliza RSVP la marginea reţelei şi DiffServ în interiorul ei. Cele două tehnologii au câteva proprietăţi complementare care ar face din aceasta combinaţie o soluţie de aplicat. RSVP excelează la managementul pe flux unic, dar nu e scalabil. De asemenea, deoarece e mult mai complex decât DiffServ, se recomandă a nu se utiliza pe backbone de reţea.

Pe de altă parte, DiffServ are capabilităţi de management de resurse limitat, dar este mult mai scalabil decât RSVP. De aceea, folosirea RSVP ca metodă de acces şi a DiffServ ca metodă de forwarding ar putea fi o soluţie foarte eficientă în încercarea de a susţine nivelurile de QoS peste o reţea.

4.8. **Fiabilitatea**

Cu toate că întreruperile la nivel de rețea sunt rare, este esențial să fie luate în calcul. Operatorii au nevoie de strategii pentru a întâmpina situațiile când rețeaua de date întâmpină dificultăți. Pentru a acoperi erori de acest fel la nivel de rețea, operatorii trebuie să aibă legături redundante între diverse puncte ale ei. De asemenea, mai poate ajuta şi o structură de gateway-uri redundante.

Rețele IP folosesc protocoale de rutare pentru a schimba informația de rutare. Una dintre funcționalitățile protocoalelor de rutare este de a monitoriza legăturile de interconectare. În cazul în care apar erori la nivelul rețelei, protocoalele de rutare retrimit pachetele pe altă rută dacă aceasta există. Timpul necesar pentru detectarea erorilor şi pentru recalcularea rutelor poate varia în funcție de mediile de interconectare dintre echipamente.

Dacă în structura rețelei există gateway-uri, acestea pot detecta statusul următorului punct (hop) din rețea, acesta funcționând ca şi mecanism de eliminare a erorilor.

4.9. **Securitatea**

Rețelele VoIP sunt vulnerabile la multe dintre riscurile la care sunt expuse şi rețelele de date, incluzând atacuri de tip Denial of Service (DoS), furtul serviciilor şi fraudă. Multe dintre firewall-urile obişnuite nu pot combate atacurile VoIP pentru că VoIP este implementat şi la nivelul de semnalizare, şi la nivelul de transmisie.

VoIP prezintă un set de vulnerabilități ale sale proprii, multe legate de căile de semnalizare folosite şi de mediile de transmisie. SBC-urile (Session Border Controller) oferă un set de unelte pentru păstrarea securității în rețea. De exemplu, SBC filtrează sesiunile VoIP în funcție de diferite criterii: doar traficul dorit traversează SBC-ul către destinații predefinite. De asemenea, filtrează traficul pentru a proteja echipamentele VoIP împotriva atacurilor de tip DoS.

Firewall-urile pot avea şi funcţionalitate de protecţie a securităţii. Dar pentru a deschide şi închide porturile pentru traficul VoIP doar în timpul unei convorbiri, ele trebuie să recunoască protocoalele de semnalizare folosite în VoIP. Altfel, apelurile de voce nu pot trece prin firewall doar dacă un set de porturi sunt deschise – ceea ce expune reţeaua la atacuri neautorizate.

Operatorii trebuie să îşi instaleze firewall-uri pentru a proteja serverele şi terminalele VoIP. Aceste politici de limitare trebuie să se bazeze pe autorizarea traficului realizat de terminalele din reţea. Firewall-urile pot fi folosite şi pentru a separa traficul VoIP de restul traficului din reţea, şi pentru a stabili nişte priorităţi în transmiterea pachetelor.

Capitolul 5

APLICAȚIA VoIP

5.1. Descriere generală

Aplicația VoIP realizată este un client simplu de voce ce respectă standardul H.323 prin intermediul librăriilor PWLib şi OpenH323. Aplicația implementează câteva funcții de bază ale VoIP: apelarea unui IP, aşteptarea apelurilor, trimiterea de mesaje text.

Pentru a stabili o legătură de voce între două locații (calculatoare) aplicația trebuie să existe pe ambele stații iar între cele două să existe o legătură permanentă, printr-o rețea LAN sau prin Internet astfel încât între cele două locații să se poată crea o legătură bazată pe adresa IP a acestora. Aplicația este realizată în Visual Studio VC++ 6.0.

Fereastra principală arată ca în figura 5.1. Implicit, aplicația aşteaptă să primească apeluri. Pentru a iniția o convorbire spre o anumită destinație, se introduce adresa IP în câmpul „apelează" şi se apasă butonul „apel". În permanență în căsuța „mesaje" se poate observa statusul convorbirii. La inițierea apelului, aplicația trimite un mesaj de tip H.225 către IP-ul apelat, cerând începerea unei conexiuni.

Persoana apelată, care trebuie să aibă o copie a programului în execuție, va fi sesizată cu un mesaj că este apelată de cineva, activându-i-se prin fereastra de comunicare butoanele de acceptare sau respingere a apelului. Dacă se respinge apelul persoana care l-a inițiat va primi mesajul corespunzător iar dacă se acceptă se trece la negocierea legăturii între cele 2 aplicații client.

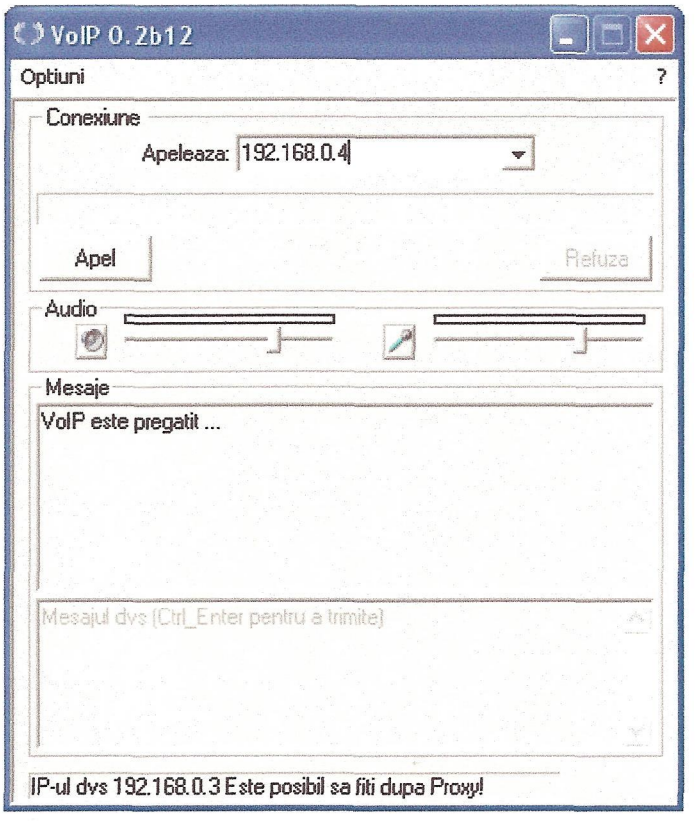

Fig. 5.1 – **Fereastra principală a aplicației VoIP**

5.2. Biblioteci/librării folosite

Programul VoIP prezentat este bazat pe librăriile PWLib şi OpenH323.

PWLib este o librărie destul de extinsă care a fost creată pentru a permite aplicațiilor scrise cu ajutorul ei să ruleze şi sub Microsoft Windows, dar şi sub UNIX.

Librăriile au crescut până au ajuns să se extindă la mai mult decât portabilitatea mediului grafic Windows. Există clase pentru operațiuni de intrare–ieşire, portabilitatea aplicațiilor multi-threading, posibilitatea de creare a daemon-ilor în UNIX şi a serviciilor în Windows NT şi, de-a lungul timpului, au fost adăugate diverse protocoale.

Ținta proiectului OpenH323 este aceea de a dezvolta o bibliotecă de programe open-source care să implementeze setul de protocoale VoIP H.323. Codul este scris în C++ şi, prin efortul multor oameni din jurul lumii, suportă în acest moment în totalitate protocolul H.323. Aceste biblioteci sunt folosite chiar şi de aplicații comerciale.

5.3. Funcționalități

Pe lângă funcția de bază, aceea de stabili apeluri de voce, aplicația are un panou de opțiuni, prezentat în continuare.

Tab-ul de opțiuni generale (figura 5.2), oferă posibilitatea utilizatorului să își seteze numele de la care provine apelul, să seteze aplicația să răspundă automat la orice apel primit, precum și setarea unui fișier wav redat la primirea unui apel.

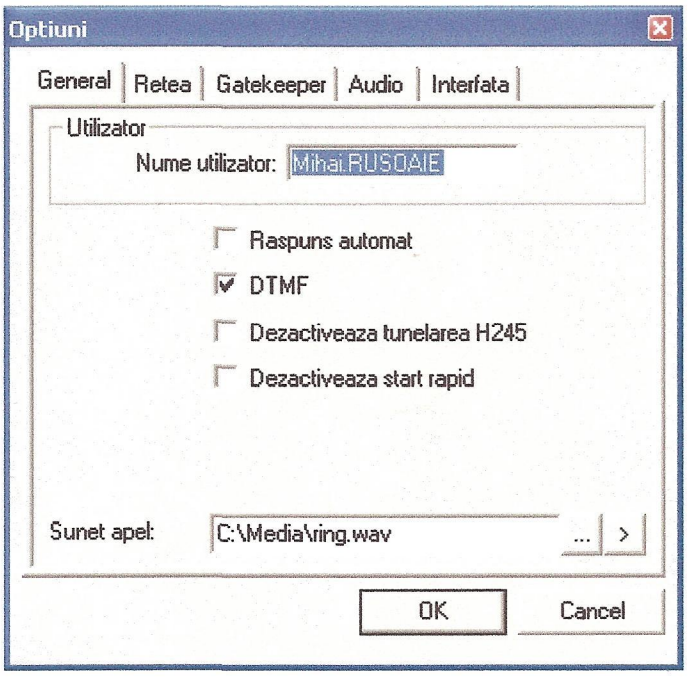

Fig. 5.2 – **Fereastra cu opțiuni generale**

Tab-ul cu opțiuni de rețea (figura 5.3) permite setarea lățimii de bandă disponibilă, pentru a se putea stabili o conexiune de voce în condiții optime de calitate a sunetului. În cazul în care clientul se află după NAT, aici se poate specifica adresa acestuia.

Pentru a specifica *gatekeeper*-ul, se folosește fereastra de opțiuni, prezentată în figura 5.4, de unde se poate seta adresa IP a gatekeeper-ului, precum și modalitățile de acces la el.

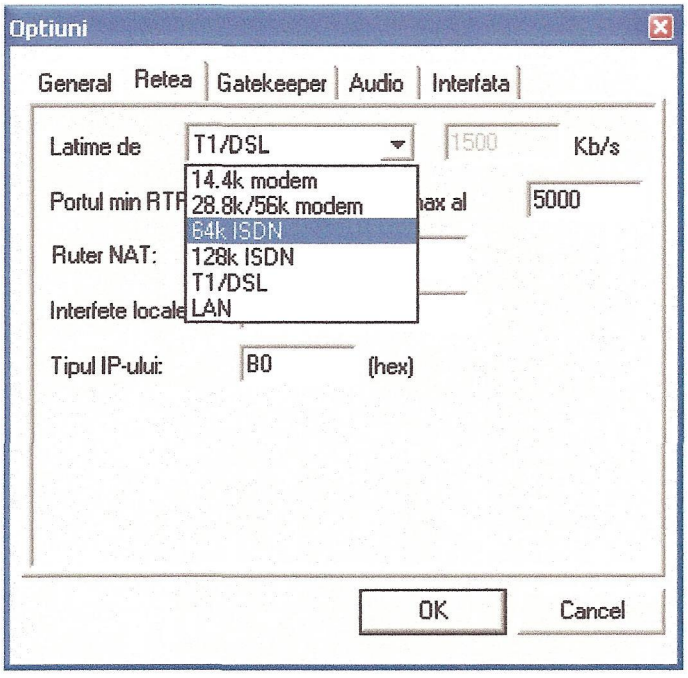

Fig. 5.3 – **Fereastra cu opțiuni de rețea**

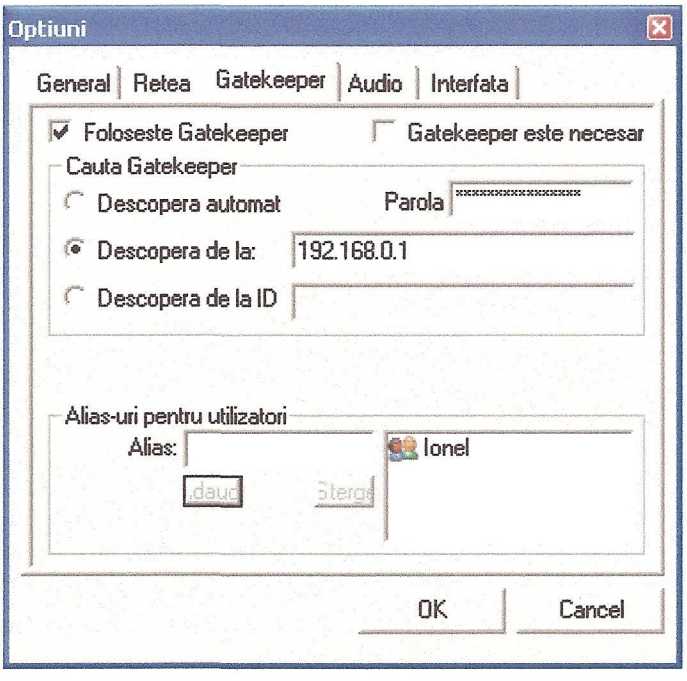

Fig. 5.4 – **Fereastra cu opțiuni privind gatekeeper-ul**

Opțiunile audio (figura 5.5) permit schimbarea interfeței de sunet folosită de aplicație, precum şi selecția codec-urilor preferate.

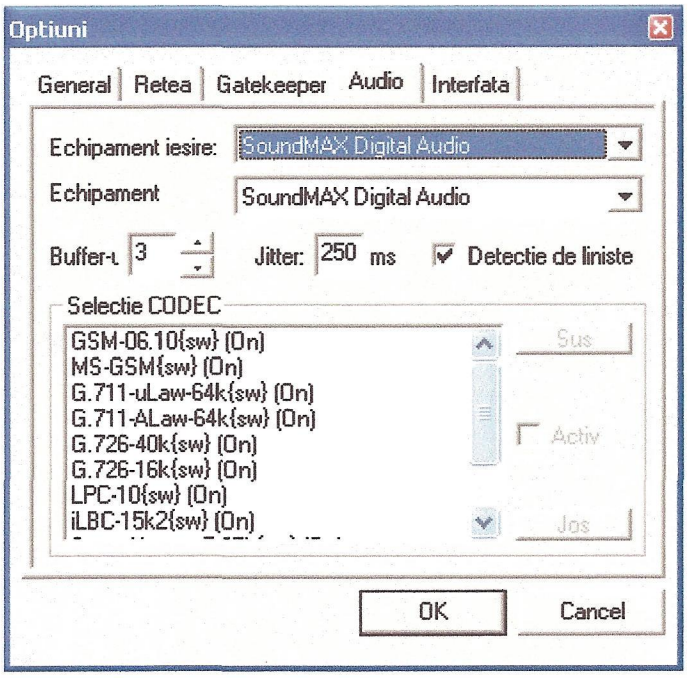

Fig. 5.5 – **Fereastra cu opțiuni audio**

Din *opțiunile legate de interfață* (figura 5.6), se poate selecta vizibilitatea anumitor controale de pe panoul principal.

Fig. 5.6 – **Fereastra cu opțiuni legate de interfață**

5.4. **Aspecte de programare**

Aplicația este constituită dintr-o clasă principală CmyPhoneConnection: CMyPhoneConnection ce implementează funcțiile de apelare ale aplicației, precum şi alte clase ce realizează funcționalitățile interfeței.

La recepționarea unui apel, interfața apelează metoda OnCall:

```
void CMyPhoneDlg::OnCall()
{
	if(!m_endpoint.HasConnection(m_token))
	{
		UpdateData();
		m_call.EnableWindow(FALSE);
		m_refuse.EnableWindow();
		GetDlgItem(IDC_REFUSE)->SetWindowText((LPCTSTR)LoadStringLang(IDS_HANGUPBSTR));
		//
		CString curDest = m_destination;
		curDest.TrimLeft();
		int iPos=-1;
		if((iPos=curDest.Find(" "))>0)
			curDest = curDest.Left(iPos);
		m_endpoint.MakeCall((const char *)curDest, m_token);
		// Maybe this should be moved to OnConnectionEsteblished
		PhoneBookAddCall(1, curDest, curDest);
	}
	else
	{
		ringSoundTimer.Stop();
		m_caller.SetWindowText("");
		//		m_answer.EnableWindow(FALSE);
		m_refuse.EnableWindow(FALSE);
		//		m_hangup.EnableWindow(FALSE);
		m_call.EnableWindow();
		GetDlgItem(IDC_CALL)->SetWindowText((LPCTSTR)LoadStringLang(IDS_CALLBSTR));

		H323Connection * connection = m_endpoint.FindConnectionWithLock(m_token);
		if (connection == NULL)
			m_call.EnableWindow();
		else
		{
			connection->AnsweringCall(H323Connection::AnswerCallNow);
			connection->Unlock();
		}
	}
}
BOOL CMyPhoneDlg::OnAnswerCall(const H323Connection & connection)
```

```
{
        PString caller = FindContactName(connection)/*connection.GetRemotePartyName()*/;
        m_token = connection.GetCallToken();
    m_caller.SetWindowText(CString((const char*)caller) + LoadStringLang(IDS_CALLINGSTR));
    //m_answer.EnableWindow();
    m_refuse.EnableWindow();
    m_call.EnableWindow(TRUE);
        GetDlgItem(IDC_CALL)->SetWindowText((LPCTSTR)LoadStringLang(IDS_ANSWERBSTR));
        GetDlgItem(IDC_REFUSE)->SetWindowText((LPCTSTR)LoadStringLang(IDS_REFUSEBSTR));
    OutputStatusStr((LPCTSTR)LoadStringLang(IDS_CALLING1STR), S_SYSTEM, (const char *)caller);
        PTime now;
        PString nowStr = now.AsString("w h:m a");

        // Check for auto answer option
        if (m_endpoint.m_fAutoAnswer)
        {
                ringSoundTimer.Stop();
                return TRUE;
        }

        if (!ringSoundFile)
        {
                PSound::PlayFile(ringSoundFile, FALSE);
                ringSoundTimer.RunContinuous(5000);
        }

        //      if (!noAnswerForwardParty)
        //              noAnswerTimer = noAnswerTime;
        return FALSE;
}
```

De exemplu, pentru a optimiza apelul în funcție de lățimea de bandă disponibilă, se poate schimba această setare din fereastra de opțiuni rețea:

```
void CNetworkPage::OnSelchangeBandwidthCombo()
{
        int nSel = m_BandwidthTypeCtrl.GetCurSel();
        m_BandwidthEdt.EnableWindow(nSel==6);
        if (nSel<6 && nSel>=0)
        {
                static double const bandwidths[6] = {14400, 28800, 64000, 128000, 1500000,
10000000};
                double bandwidth = bandwidths[nSel]/1000.0;
                CString value;
                value.Format(_T("8.2f"),bandwidth);
                m_BandwidthEdt.SetWindowText(value);
        }
}
```

Pentru a susține fluxul audio, se folosesc codec-uri specifice, instanțiate cu ajutorul funcției:

```
BOOL CMyPhoneEndPoint::OpenAudioChannel(H323Connection & connection, BOOL isEncoding,
unsigned bufferSize, H323AudioCodec & codec)
{
    return H323EndPoint::OpenAudioChannel(connection, isEncoding, bufferSize, codec);
}
```

5.5. Analiza calității apelurilor de voce

Analiza s-a realizat cu ajutorul programului Hammer Call Analyzer (HCA) – produs de Empirix – software specializat în analiza sistemelor VoIP. Acesta capturează pachete IP dintr-o rețea şi detaliază un apel la nivel de protocoale şi mesaje ce se schimbă între părți. În figura 5.7 se poate vedea detaliat interfața programului în stadiul de captură. Aceasta se împarte în 4 mari componente: lista pachetelor capturate, schema cu protocoalele VoIP detectate, detaliile pachetelor IP şi pachetul capturat în format binar.

Pe lângă măsurătorile clasice, prezentate în tabelul 5.1, Hammer Call Analyzer extrage şi caracteristicile SQS (Stream Quality Signature), din graficul care arată frecvența şi distribuția variației timpului la care ajung pachetele (figura 5.10). Distribuția este prezentată pe nouă sloturi de timp statice aranjate pe axa X în ordine crescătoare. Numărul de pachete ce există în fiecare slot este arătat pe axa Y folosind o scală logaritmică. Pentru fiecare caz, se afişează procentajul din toate pachetele ce sunt conținute în acel slot de timp.

Scorul SQS este calculat după următoarea formulă:

$$\textbf{Scorul SQS pentru pachetul N+1} = |(P2_{at} - P1_{at}) - ((P2_{ts} - P1_{ts})/8)|$$

unde:

$P1_{at}$ = timpul de recepție a pachetului n, în milisecunde
$P2_{at}$ = timpul de recepție a pachetului n+1, în milisecunde
$P1_{ts}$ = timpul conținut în pachetul n, în unități a câte 125 de milisecunde
$P2_{ts}$ = timpul conținut în pachetul n+1, în unități a câte 125 de milisecunde

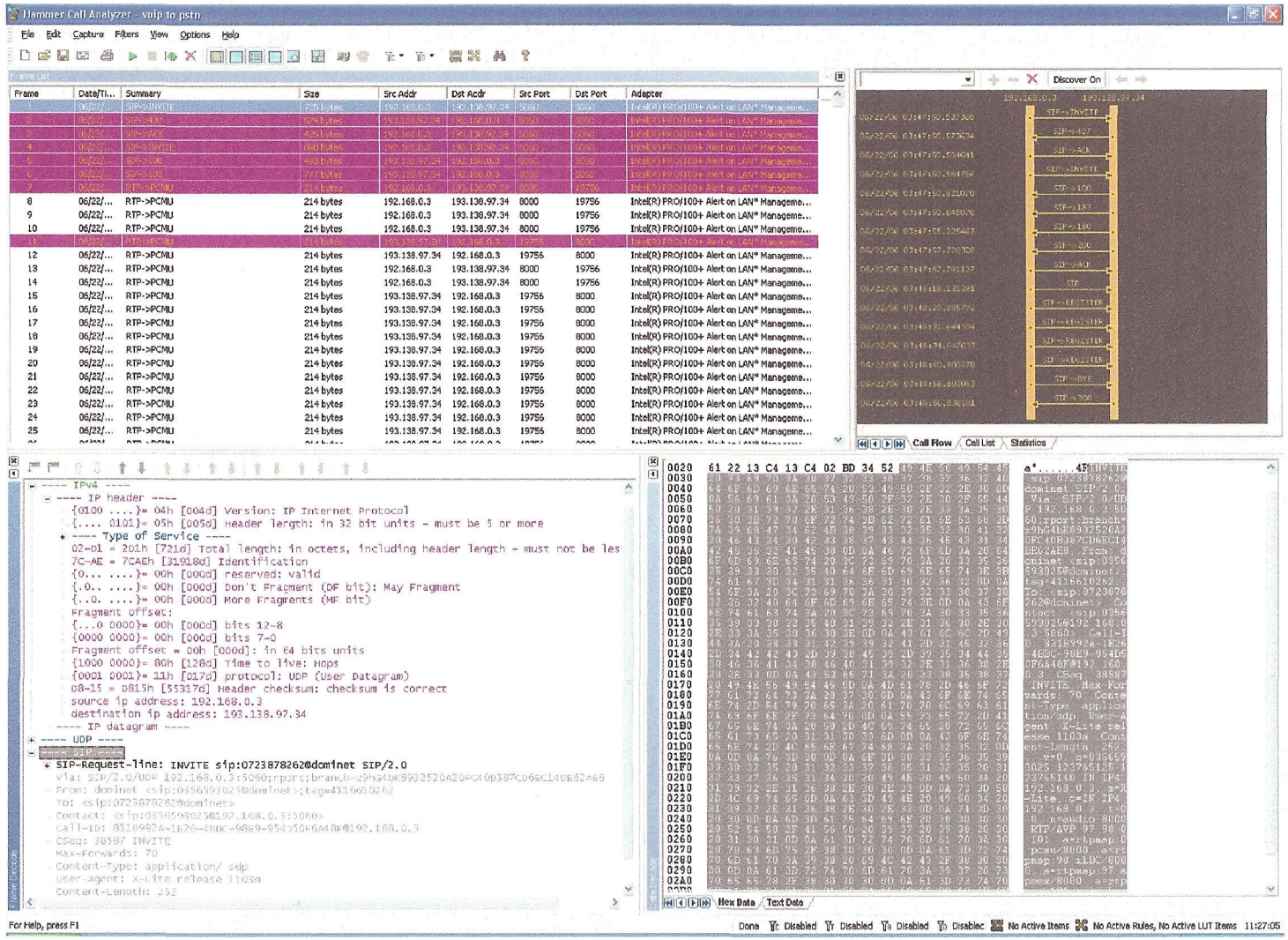

Fig. 5.7 – **Interfața programului Hammer Call Analyzer (HCA)**

Tabelul 5.1

Măsurătorile standard obținute cu ajutorul HCA

Metrică	Descriere
Tipul încărcăturii	Tipul încărcăturii fluxului RTP așa cum este identificat de către antetul RTP.
Pachete recepționate	Numărul de pachete RTP recepționate
Pachete pierdute	Numărul de pachete RTP pierdute, așa cum se identifică spațiile din secvența de numere RTP.
Pachete nesincronizate	Numărul de pachete detectate ca nefiind recepționate în ordinea firească indicată de RTP.
Pachete duble	Numărul de pachete duble recepționate
Jitter	Media jitter-ului pentru fluxul RTP
Jitter maxim	Jitter-ul maxim pentru fluxul RTP.
Numărul de resetări ale buffer-ului jitter	De câte ori buffer-ul jitter-ului s-a resetat.
Numărul de resetări ale amprentei de timp (timestamp) a buffer-ului jitter-ului	De câte ori s-a resetat amprenta de timp (timestamp) a buffer-ului jitter-ului
Factorul R	Standard de măsurare a calității vocii. Se măsoară de la 0 (cel mai slab) la 100 (cel mai bun).Un factor R de 80 se traduce într-un MOS de 4, iar unul de 60 într-un MOS de 3
Mean Opinion Score (MOS)	Factor de măsurare a calității vocii.
Media RTT	Media timpului de răspuns (round trip time).
RTT maxim	Maximul timpului de răspuns.
Întârzierile buffer-ului jitter (ms)	Setările curente sau implicite ale buffer-ului.
Pachete eronate din buffer	Numărul de pachete ignorate datorită timpului de recepție mult mai mare decât permite buffer-ul.
Pachetele jitter-ului nesincronizate	Numărul de pachete ce au ajuns la buffer nesincronizate
SQS Bin 1 (0.5)	Numărul de pachete ce au obținut un scor SQS mai mic sau egal cu 0,5 ms.
SQS Bin 2 (5)	Numărul de pachete ce au obținut un scor SQS mai mare decât 0,5 ms, dar mai mic sau egal cu 5 ms.
SQS Bin 3 (10)	Numărul de pachete ce au obținut un scor SQS mai mare decât 5 ms, dar mai mic sau egal cu 10 ms.
SQS Bin 4 (15)	Numărul de pachete ce au obținut un scor SQS mai mare decât 10 ms, dar mai mic sau egal cu 15 ms.
SQS Bin 5 (20)	Numărul de pachete ce au obținut un scor SQS mai mare decât 15 ms, dar mai mic sau egal cu 20 ms.
SQS Bin 6 (25)	Numărul de pachete ce au obținut un scor SQS mai mare decât 20 ms, dar mai mic sau egal cu 25 ms.
SQS Bin 7 (30)	Numărul de pachete ce au obținut un scor SQS mai mare decât 25 ms, dar mai mic sau egal cu 30 ms.
SQS Bin 8 (35)	Numărul de pachete ce au obținut un scor SQS mai mare decât 30 ms, dar mai mic sau egal cu 35 ms.
SQS Bin 9 (>35)	Numărul de pachete ce au obținut un scor SQS mai mare decât 35 ms

Pentru a realiza o comparație între informațiile capturate, considerăm următoarele trei cazuri:

1. rețea ce se bazează pe transferul datelor printr-o rețea publică de date (Internet). Se folosește un gateway aflat la distanță mare de cele două terminale care comunică;
2. rețea ce comunică datele printr-o rețea publică de date, dar la nivel metropolitan (local) – MAN[1];
3. rețea ce comunică datele doar la nivel local – LAN[2].

Pentru **cazul 1** considerăm structura rețelei din figura 5.8.

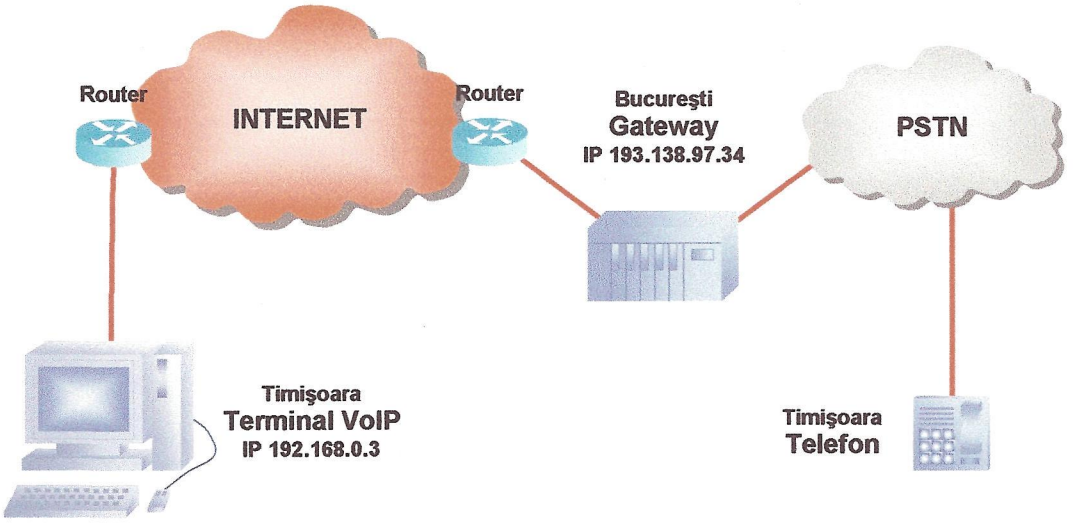

Fig. 5.8 – **Structura rețelei în cazul 1**

În acest caz, datorită folosirii unei rețele publice de date, precum și din cauza gâtuirii traficului prin router-ul aflat la terminalul VoIP, convorbirea a obținut un MOS de doar 2,48 (tabelul 5.2). Jitterul se poate observa cu roșu în figura 5.10

[1] **MAN** – *Metropolitan Area Network* – rețea de comunicații de date la nivel metropolitan, de obicei se referă la rețeaua de date din interiorul unui oraș sau campus
[2] **LAN** – *Local Area Network*

Tabelul 5.2

Detaliile convorbirii în cazul 1

Metrică	Valoare
Tipul încărcăturii	PCMU
Pachete recepționate	3385
Pachete pierdute	894
Pachete nesincronizate	0
Pachete duble	0
Jitter	4
Jitter maxim	27
Factorul R	58
Mean Opinion Score (MOS)	2,48438
Media RTT	nedisponibil
RTT maxim	nedisponibil
Întârzierile buffer-ului jitter (ms)	20
Pachete eronate din buffer	0
Pachetele jitter-ului nesincronizate	*0*
SQS Bin 1 (0.5)	1664
SQS Bin 2 (5)	1701
SQS Bin 3 (10)	14
SQS Bin 4 (15)	3
SQS Bin 5 (20)	0
SQS Bin 6 (25)	0
SQS Bin 7 (30)	0
SQS Bin 8 (35)	0
SQS Bin 9 (>35)	0

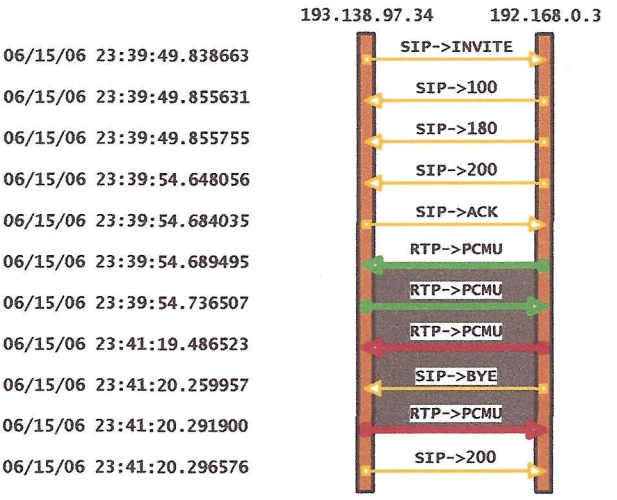

Fig. 5.9 – **Schema cu protocoalele VoIP, în cazul 1**

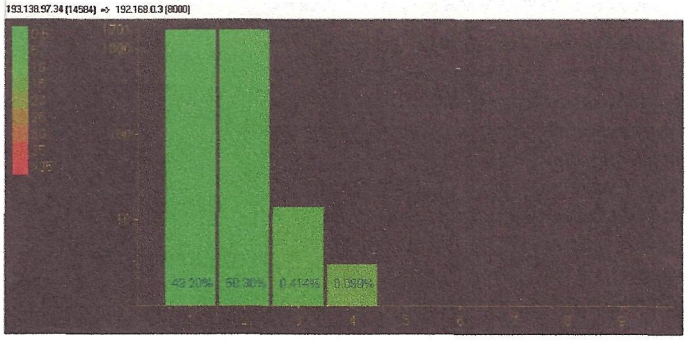

Figura 5.10 – **Cazul 1: scor SQS**

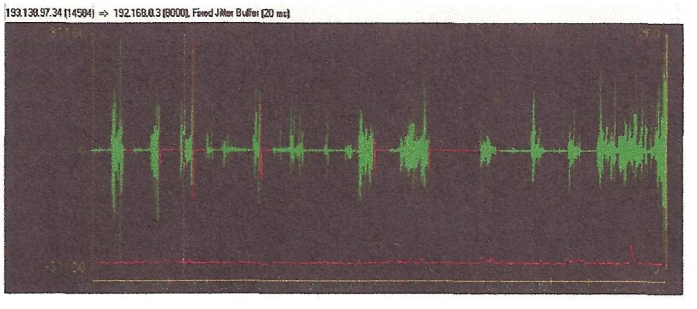

Figura 5.11 – **Cazul 1: eşantioane de voce (verde) şi jitter (roşu)**

În **cazul 2** avem o infrastructură fără gateway, apelul realizându-se de la IP la IP (figura 5.12). Din aceste considerente, legătura dintre terminale se va realiza prin intermediul rețelei MAN, ce are timpi de răspuns mult mai buni decât în primul caz (IP-ul considerat în acest exemplu se află la 3 hop-uri de sursă şi are timp de răspuns 3 ms). MOS obținut este de 3,6 (tabelul 5.3).

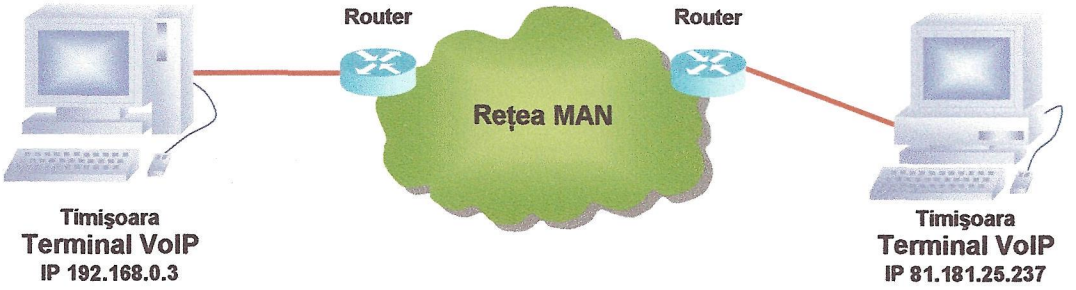

Fig. 5.12 – **Structura rețelei în cazul 2**

Tabelul 5.3

Detaliile convorbirii în cazul 2

Metrică	Valoare
Tipul încărcăturii	PCMU
Pachete recepționate	966
Pachete pierdute	0
Pachete nesincronizate	0
Pachete duble	0
Jitter	11
Jitter maxim	41
Factorul R	91
Mean Opinion Score (MOS)	3,60156
Media RTT	nedisponibil
RTT maxim	nedisponibil
Întârzierile buffer-ului jitter (ms)	20
Pachete eronate din buffer	1
Pachetele jitter-ului nesincronizate	0
SQS Bin 1 (0.5)	89
SQS Bin 2 (5)	849
SQS Bin 3 (10)	0
SQS Bin 4 (15)	16
SQS Bin 5 (20)	3
SQS Bin 6 (25)	3
SQS Bin 7 (30)	2
SQS Bin 8 (35)	0
SQS Bin 9 (>35)	1

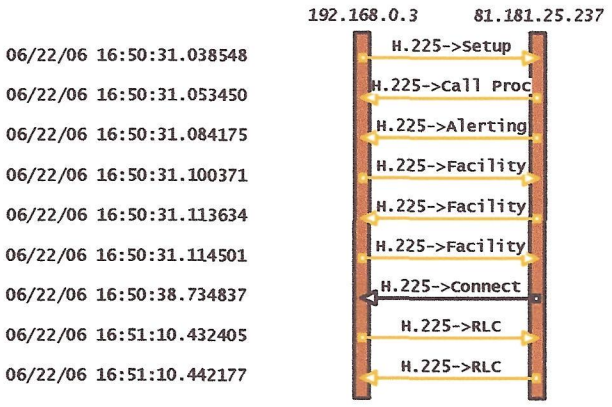

Fig. 5.13 – **Schema cu protocoalele VoIP, în cazul 2**

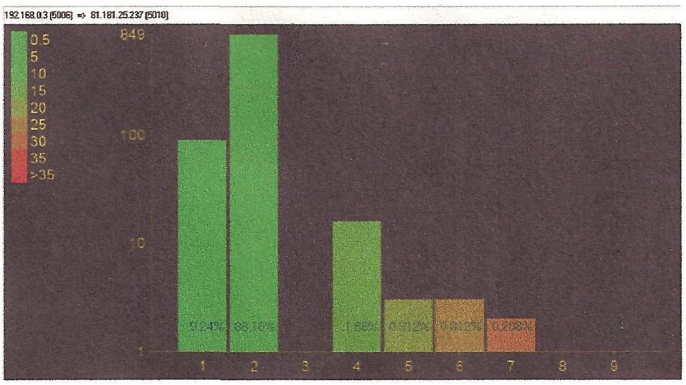

Figura 5.14 – **Cazul 2: scor SQS**

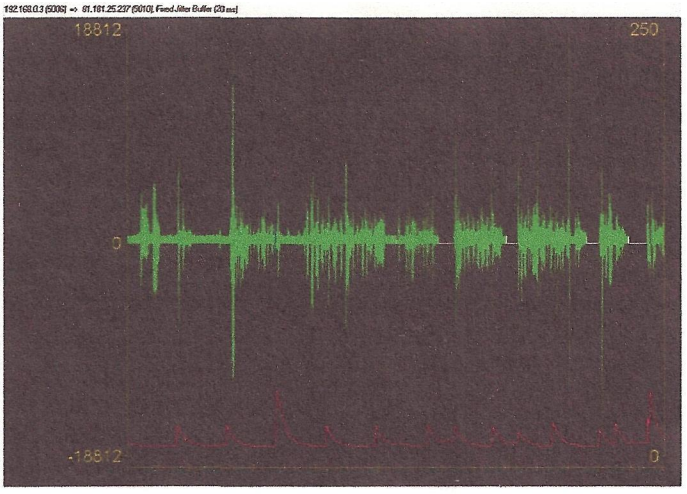

Figura 5.15 – **Cazul 2: eşantioane de voce (verde) şi jitter (roşu)**

Cazul 3, considerat este şi cel mai simplu ca şi arhitectură, convorbirea desfăşurându-se într-o reţea locală – LAN (figura 5.16), caz în care între cele două terminale există un singur router, iar timpii de răspuns sunt mai mici decât 1 ms. MOS obţinut este de 3,63 (tabelul 5.4).

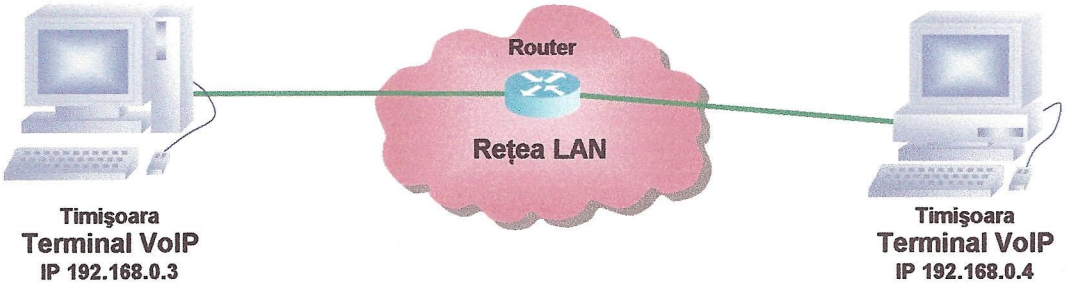

Fig. 5.16 – **Structura reţelei în cazul 3**

Tabelul 5.4

Detaliile convorbirii în cazul 3

Metrică	Valoare
Tipul încărcăturii	PCMU
Pachete recepționate	842
Pachete pierdute	0
Pachete nesincronizate	0
Pachete duble	0
Jitter	8
Jitter maxim	34
Factorul R	93
Mean Opinion Score (MOS)	3,63672
Media RTT	nedisponibil
RTT maxim	nedisponibil
Întârzierile buffer-ului jitter (ms)	20
Pachete eronate din buffer	0
Pachetele jitter-ului nesincronizate	0
SQS Bin 1 (0.5)	82
SQS Bin 2 (5)	738
SQS Bin 3 (10)	2
SQS Bin 4 (15)	10
SQS Bin 5 (20)	6
SQS Bin 6 (25)	0
SQS Bin 7 (30)	1
SQS Bin 8 (35)	0
SQS Bin 9 (>35)	0

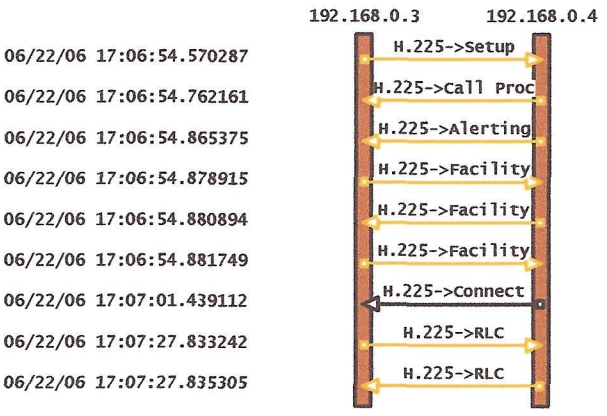

Fig. 5.17 – **Schema cu protocoalele VoIP, în cazul 3**

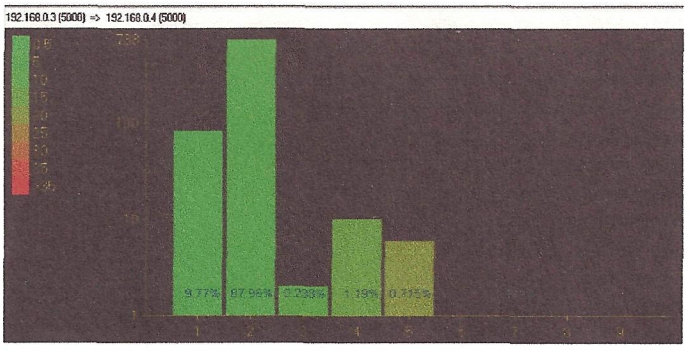

Figura 5.18 – **Cazul 3: scor SQS**

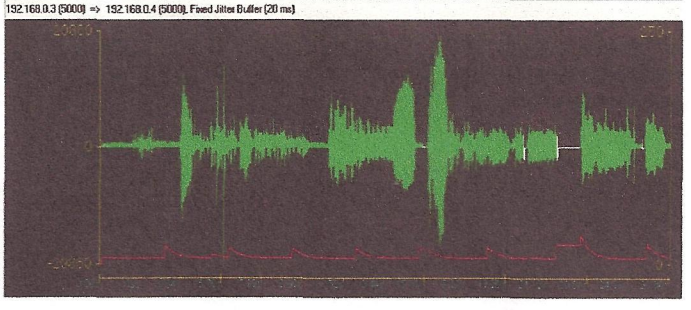

Figura 5.19 – **Cazul 3: eşantioane de voce (verde) şi jitter (roşu)**

CONCLUZII

VoIP este o aplicație ce rulează într-o rețea orientată pe cumutația de pachete şi are nevoi stringente de performanță. Această performanță a rețelei IP are un impact direct asupra calității vocii. Această lucrare identifică factorii perturbatori ai transmisiei ce trebuie măsurați. Acestea includ pierderile de pachete, întârzierile şi jitterul. Calitatea serviciilor este o componentă esențială într-o rețea IP. În momentul în care există o restricție de resurse, ca de exemplu congestionarea rețelei, este important ca aceasta să ofere servicii mai bune traficului în timp real (real-time) ca de exemplu traficul făcut de conexiunile VoIP, comparativ cu fluxul de date.

Când se doreşte o analiză asupra unei rețele VoIP există un set de teste din care se poate alege. Acestea includ analiza rețelei IP, testarea vocii la ambele capete ale comunicației, simulări de situații limită precum şi teste ale semnalizării.

În general, pentru ca furnizorii de servicii în telecomunicații să poată oferi clienților lor un serviciu bun din punct de vedere calitativ, se impune, de cele mai multe ori, să se folosească echipamente hardware dedicate traficului de voce, iar mediul fizic de transmisie să fie unul dedicat, sau, în cazul în care se foloseşte, din considerente de costuri, o rețea publică de date, trebuie asigurată o lățime de bandă minim garantată pentru un serviciu acceptabil.

BIBLIOGRAFIE

Brunner St.– *Understanding VoIP Networks*, Juniper, 2004

Gallon C.– *Quality of Service for Next Generation Voice Over IP Networks*, 2003

Liu Chunlei – *Multimedia Over IP: RSVP, RTP, RTCP, RTSP*, în: Handbook of Communication Technologies: The Next Decade, CRC Press, Boca Raton, Florida, pp 29–46, 1999.

Mathys H. – *Next Generation Voice Networks*, 2005

Padjen R. – *CISCO AVVID and IP Telephony – Design and Implementation,* Syngress Publishing, 2001

Schulzrinne, H. – *Comparison of H.323 and SIP*, 1999

Tanenbaum A. S. – *Computer Networks*, Forth Edition, Pretince Hall, 2003

Cisco IP Telephony, Student Guide, ver. 2.2, 2002

Voice over IP (VoIP) – Spirent Communications, 2001

Cisco – Understanding H.323 Gatekeepers, 2006

DiffServ—The Scalable End-to-End QoS Model 1992-2001 Cisco Systems

Cisco IP Telephony QoS Design Guide, 2001

SIP: Protocol Overview, Radvision, 2001

Internet Engineering Task Force - Real Time Streaming Protocol (RTSP), 1998

Resurse Internet:

www.cisco.com

www.protocols.com

www.wikipedia.org

developer.intel.com