Facultatea de Electronică, Telecomunicaţii și Tehnologia Informaţiei

Universitatea Politehnica București

**Reţele de Calculatoare**

Nivelul de Transport, comunicarea în rețea TCP/IP

*Profesor coordonator: Studenți:*

prof. univ.dr. ing. Ștefan Stăncescu Sterie Daniel

Sandu Laurențiu

2015-2016

***Cuprins:***

1. Noțiuni introductive TCP/IP *(Sterie Daniel)* 3

2. Nivele ale modelului TCP/IP 8

3. Diferențe majore dintre modelul TCP/IP si modelul OSI 12

4. Nivelul de Transport - prezentare si principii de funcționare 13

*(Sandu Laurentiu)*

5. Protocoale principale ale Nivelului de Transport 15

                5.1 Protocolul TCP 16

                5.2 Protocolul UDL 18

                5.3 Protocolul RSVP 19

                5.4 Protocolul DCCP 19

6.Concluzii *(Sterie Daniel, Sandu Laurentiu)* 20

7 Bibliografie  21

***1. TCP/IP - noțiuni introductive***

TCP/IP - provine de la "Internet Protocol Suite", traducerea fiind puțin diferita (Suita de Protocoale ale Internetului) - este un model de referința pe baza caruia se construiesc arhitecturile de rețea, cuprinzand protocoale ce ajuta la comunicarea dintre terminalele acesteia. Insa, pentru a putea ințelege cu adevarat scopul unui model TCP/IP si influența acestuia asupra telecomunicației, este necesar a ințelege temenii "protocol" si "rețea" .

Un termen cu o foarte mare importanța in cadrul acestei lucrari il rețea "comunicare". Prin comunicare se ințelege schimbul de informație intre un emitațor si un receptor uman, prin intermediul unui canal de transmisiune. Acest proces poate fi transpus in rețelistica prin comunicarea a doua terminale prin intermediul unei retele, reprezentand canalul de comunicație, realizand un transport de date (informații).

Cum oamenii folosesc cuvintele pentru a transmite o informație, asa "transporta" si terminalele pachete de date ce au un anumit scop prestabilit de catre utilizator.

Rețeaua o putem definii ca fiind o metoda de comunicare intre terminale si/sau servere, ce implica atat o parte fizica numita "topologie", alcatuita din fire (fibra optica) si componente caracteristice (routere, hub-uri etc.), cat si o parte abstracta, formata din reguli de transport, comunicare, inter-schimbare de resurse, ce constituie practic mecanismul central care asigura buna funcționare mult dorita de catre proiectant. In acest caz putem afirma faptul ca doua sau mai multe terminale (termen utilizat in cadrul acestei lucrari cu scopul de a face referire la orice aparat care se poate conecta la o rețea, cum ar fi un computer, telefon mobil, laptop, servere etc), interconectate printr-o metoda prestabilita ce prezinta reguli de comun acord pe baza carora se efectueaza un schimb de date, se afla "intr-o rețea" (noțiune ce urmeaza a fi explicata in capitolele viitoare ale lucrarii de fata). Asemanator acestui concept de canal de comunicație, s-au dezvoltat diverse mecanisme elaborate, precum accesul la World Wide Web, acționarea unui printer prin transmisiunea wireless, utilizarea aplicațiilor de tip e-mail sau mesagerie instantanee fapt ce a condus umanitatea cu un pas mai aproape de modernizare si progres.\*

Avand cunostințe referitoare la termenul de rețea, putem explica noțiunea de "protocol". Protocolul reprezinta un set de regului ce servesc un scop bine definit in transportul datelor, verificarea acestora la receptie , mențtinerea calitatii prin redugecea zgomotului la reproduscerea semnalului. Altfel spus, protocolul poate fi o implementare de timp hardware cat si software, reprezentand o notiune abstracta.

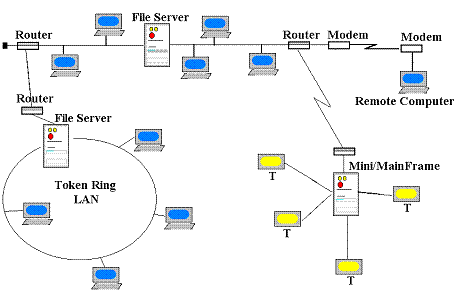


Fig 1. Structura unei retele

Cunoscand la nivel conceptual termenii de "rețea" si "protocol", putem incerca o definitie mai academica asupra modelului TCP/IP, si anume ca, reprezinta un sablon alcatuit din patru nivele, fiecare avand un set de reguli (protocoale) cu scop bine stabilit, pe baza caruia se poate intocmi un canal de transmisiune a datelor intre terminale (rețea), atat la nivel fizic prin componente caracteristice domeniului, cat si la nivel abstract, prin modelarea unor proceduri complexe ce au la baza protocoalele mentionate. Trebuie precizat faptul ca, modelul conceptual TCP/IP nu prezinta o stransa legatura cu topologia retelei, insa se impune o oarecare analiza asupra acesteia daca dorim a asigura un canal de transmisiune "error proof", cum ar fi (tradus ca "fara erori), ramanand in sarcina proiectantului de a gasi o punta de mijloc, optima, intre componentele utilizate si procedurile formate. [2] Componentele fizice care asigura buna functionare a retelei pot fi inctocmite dupa mai multe standarde, precum "Cisco Systems Incorporated – CISCO" sau "Institute of Electrical and Electronics Engineering – IEEE". \*

In ce le ce urmeaza vom detalia mecnismul aflat in spatele functionari acestul model TCP/IP si de asemenea prezentand necesitatea acestuia de a fi impartit in 4 nivele bine define.

Modelul TCP/IP si-a capatat denumirea datorita celor mai importante doua protocoale ale sale, si anume, "Transmission Control Protocol" (abreviat "TCP") ce se afla pe Nivelul de Transport, asigurand o comunicare bazata pe octeti, fara erori, intre terminalele retelei, si "Internet Protocol" (abreviat "IP") aflat pe Nivelul de Internet, cu scop in rutarea cailor dintre participantii retelei. De asemenea, trebuie mentionat faptul ca aceste doua protocoale au fost printre primele standardizate de catre asociatia IETF, mecanismul din spatele acestora servind si in dezvoltarea altor modele de rețea. \*

Precum sistemele de calcul, modelul TCP/IP trebuia sa dispuna de o arhitectura ce urma a fi utilizata drept referinta in conceperea altor modele de rețea. Astfel, analizand documentatia sub numele de "RFC 1122" publicata de catre IETF, putem descoperi faptul ca TCP/IP este alcatuit din patru nivele principale, fiecare cu protocoalele sale aferente, totodata fiind construit pe baza a doua principii foarte importante.

Astfel, primul se numeste "end-to-end", elaborat in 1981 de catre Saltzer Reed, care desi a fost schimbat de nenumarate ori de la aparitia sa, mecanismul din spate a ramas acelasi, exprimand ideea ca functiile specifice aplicatiilor ar trebui sa existe in terminalele de la capatele retelelor, decat in nodurile intermediare, astfel reducand costurile productiei, asigurand si un transport al informatiei mai apropiat de notiunile "complet" si "corect". Retelele zilelor noastre inca pastreaza acest principiu, insa modificat datorita necesitatii utilizarii unui firewall, translatarea adreselor din rețea si aparitia continutului web stocat sub forma de cache.

Al doilea principiu "robustness principle" (cunoscut si sub numele de "Legea lui Postel" datorita pionierului internetului, Jon Postel), impune ideea de a fi conservativ in ceea ce trimite, insa liberal in ceea ce se primeste. Cu alte cuvinte, implementarea TCP impune terminalelor de a fi foarte stricte si concise in codul elaborat spre transmitere, insa mecanismul de la receptia acestora sa permita citirea unui sir de octeti atata timp cat intelesul mesajului este clar. [3]

Procesul transmiterii datelor se realizeaza prin intermediul procesului de incapsulare datorita celor 4 nivele.

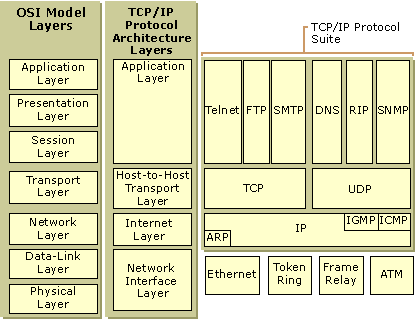


Fig 2. Nivelele modului TCP/IP

Asemeni unui continut intr-o cutiuța, ce se afla la randul ei intr-o cutie mai mare si asa mai departe (ne putem referi la papusile rusesti "matrioska" pentru o mai buna reprezentare), incapsularea presupune utilizarea notiunii de abstractizare a protocoalelor pentru a realiza o transmisiune de date intre doua terminale. Astfel, plecand de la doua procese aflate pe Nivelul de Aplicație (cel mai intal nivel) ce intercomunica, vrand sa transmita un pachet de date, putem observa ca, pe masura ce parcurgem modelul TCP/IP in adancime (spre ultimul nivel, cel de legatura), respectivul sir de octeti din pachet este "incapsulat", adaugandu-i-se alte string-uri, ce il ajuta in a fi trimis si receptionat cu succes. Pentru o mai buna analogie, ne putem gandi la pachetul de date drept o scrisoare, care poate ajunge la destinatar, dar trebuie sa fie pusa intr-un plic, apoi sa i se lipeasca un trimbru, iar ulterior, sa fie pusa intr-o cutie postala.

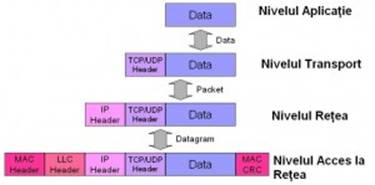


Fig 3. Incapsularea-datele pe nivele

Metoda de incapsulare a informatiei ajutat atat in buna realizare a transmisiunii (error-proof, de exemplu), fiecare protocol avand cate un rol prestabilit, insa totodata este foarte eficient deoarece imparte o sarcina in bucati de dimensiune redusa, ceea ce este o consecinta in viteza (un protocol ce asigura transmiterea fara erori si unul ce face rutarea sunt mai eficiente decat un singur protocol caruia i se impun ambele task-uri).

Pe masura ce nivele sunt parcurse, de la cel superior de Aplicație spre cel inferior de Legatura, putem afirma faptul ca exista o tranzitie de la un nivel de tip logic, mai apropiat de caracteristicile unei aplicatii, spre un nivel de tip fizic, ce are legatura mai mult cu transmiterea datelor (trecem de la o comunicare logica prin biti de 0 si 1 la una de tip fizic, ce ilustreaza transmiterea efectiva a impulsurilor electrice prin componentele aferente retelei). Putem observa acest lucru mai bine, folosindu-ne de diagrama de mai jos.

***2. Nivele ale modelului TCP/IP***

Dupa cum a tot mentionat mai sus, acest model este impartit in 4 nivele, fiecare avand protocolul sau, scopul fiind diferit de la un nivel la altul. Comform documentului "RFC 1122"[4], ne detaliaza pe scurt structura TCP/IP astfel:

*1) Nivelul de Aplicație* ("Application Layer"): este nivelul unde aplicațiile creaza date ce pot fi folosite de catre utilizatori cu scopul transmiterii acesteia spre alte aplicatii din cadrul aceleiasi retele sau aceluiasi terminal. Partenerii intre care se realizeaza transferul sunt caracterizati de catre arhitectura aplicației in cauza, putand fi de tipul "client-server model" (principiu bazat pe un program de distribuire si partitionare a resurselor unui distribuitor, pe baza cererilor emise de catre terminale) sau "peer-to-peer" (mecanism ce ilustreaza partajarea sarcinilor si resurselor intre nodurile retelei, fara a fi necesar un sistem central de administrare a acestora). Astfel, putem deduce faptul ca, acest nivel utilizeaza protocoale ce servesc in asigurarea serviciilor fata de utilizatori si realizarea schimbarii de date intr-o rețea descrisa fizic de nivelele inferioare. Drept exemple de protcoale caracteristice acestui layer avem "Hypertext Transfer Protocol", HTTP, cu scop in distribuirea informatiei de tip "hypermedia" (ceva mai mult decat siruri de biti, discutand despre poze, fisiere audio etc.) cu ajutorul "hypertext-ului" (text structurat ce ne arata legaturile logice dintre nodurile unei rețea), "Dynamic Host Configuration Protocol" ,DHCP, ce are ca rol distribuirea dinamica a parametrilor unei rete, precum adresele IP, eliminand astfel necesitatea unui sistem de tipul "administrator" sau configurarea manuala a acestuia, "File Transfer Protocol", FTP, ce asigura transmiterea datelor intre participantii comunicarii printr-o rețea ce foloseste protocolul"TCP" ("Transmission Control Protocol", va fi discutat mai in ansamblu in capitolele viitoare), sau "Simple Mail Protocol", SMP, care face posibila utilizarea bine cunoscutei aplicatii de "posta electronica". Trebuie specificat faptul ca protocoalele acestui nivel trateaza layer-ele inferioare drept "black boxes" ce asigura un canal de transmisiune stabil, chiar daca aplicatiile in cauza sunt constiente de anumiti parametrii, precum numerotarea porturilor spre exemplu. Pachetele de date sunt incapsulate,protocoalele nivelelor in ferioare neprezentand niciun interes fata de acesta.

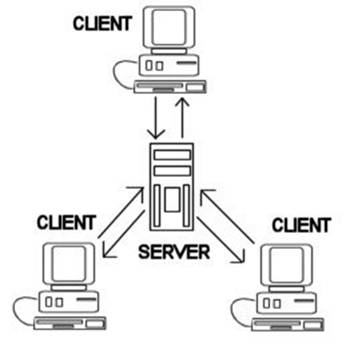


Fig 4. Modelul "client-to-client"

*2) Nivelul de Transport* ("Transportation Layer"): nivelul ce asigura transportul pachetelor de informatie intre terminalele gazda ale retelei. Acesta asigura o conectivitate "process-to-process",insemnand totodata ca realizeaza servicii "end-to-end" independente de structura datelor sau a scopului transmisiunii acestora, insa dispunand de caracteristici precum segmentarea mesajului, controlul transmisiunii sirurilor de biti (eliminand crearea unei congestii de date pe magistrala de transmisie), adresarea catre porturile cerute si controlul erorii. Atunci cand vorbim despre transport de tip "end-to-end",acesta poate fi categorisitdrept"connection-oriented",implementatin protocolul "Transmission Control Protocol", ce face referire la o sesiune de comunicare permanenta sau semi-permanenta stabilita inaintea transmiterii de date (sirul de biti este primit asa cum a fost trimis), sau drept "connectionless", find o metoda care impune transportul fiecarei bucati din pachetul de date in functie de continutul sau (implementata cu protocolul "User Datagram Protocol"). Mecanismul "connection-oriented transport" ajuta la asigurarea unui transport eficient de date, insa viteza redusa prezinta un deficit major compensatde catre metoda "connectionless", ce are aplicabilitate in domenii precum Voice Over IP Communications sau streaming-ul de fisiere video/audio. Astfel, in proiectarea unei retele, scopul defineste mecanismul utilizat din cadrul nivelului de transport.Cand ne referim la transport, dorim ca rețeaua noastra sa asigure un randament cat mai ridicat al datelor transmise cu succes, astfel protocoalele aferente Nivelului de Transport trebuie sa fie de incredere (spre exemplu, protocolul TCP asigura transmisiunea informatiei in ordine, prezinta rata de eroare minimala, datele duplicate sunt ignorate, etc.).\*

*3) Nivelul de Internet* ("Internet Layer"): acest nivel are sarcina de a realiza un schimb de datagrame in afara retelei locale, dispunand de o interfata uniforma ce ascunde topologia fizica. Practic, defineste modul de adresare si rutarea traseelor, parametrii folositide catre nivelul superior de transport (termenul de "rutare" sau "routing" in engleza reprezinta transmiterea pachetului de date de-a lungul mai multor retele, pana la destinatie, de obicei urmarindu-se traseul cel mai optim). Cel mai proeminent protocol este "Internet Protocol" (abreviat "IP"), ce serveste atat ca identificator al emitatorului, prin adresa IP, cat si alegerea unui traseu cat mai scurt dintre rețeaua de care apartine terminalul expeditor, catre rețeaua terminalului destinatar. "Internet Protocol-ul" indeplineste sarcini pentru diferite protocoale din nivele superioare, identificandu-le prin atribuirea unui numar unic fiecaruia (spre exemplu, "Internet Control Message Protocol" are numarul 1, pe cand "Internet Group Management Protocol" este notat ca 2). Dupa cum am mentionat anterior, una din functiile acestui protocol este atribuirea unei adrese pe 32 de biti fiecarui terminal pentru a putea fi identificat, sub forma a patru diviziuni de cate maxim trei cifre (ex "172.16.254.1"), astfel putand exista aproximativ patru miliarde de adrese diferite. \*

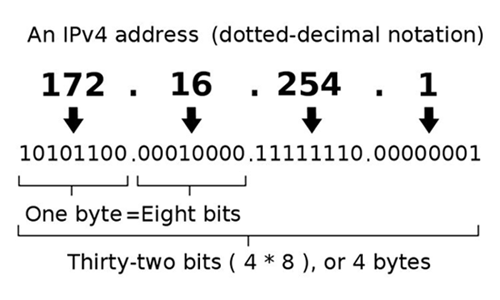


fig 5. Exemplu adresa IP

*4) Nivelul de Legatura* ("Link Layer"): este cel mai inferior nivel al modelului TCP/IP, ce asigura o buna conexiune la nivel fizic intre terminale, totodata avand protocoale ce descriu mai mult conectivitatea la un nivelmai rudimentar (descompunerea semnalului, transmisiunea acestuia prin fire etc.). Trebuie sa avem in minte faptul ca modelul din lucrarea de fata este independent de implementarea hardware, asadar o configuratie pe Nivelulde Legatura poate fi aplicat oricaror componente ce alcatuiesc o rețea. Procesul transmiterii unui pachet de date, examinat la acest layer, poate fi controlat software sau hardware prin "firmwares" (programe scrise special pentru un anumit tip de procesor de semnal) sau "chipsets" (SoC – "System on Chip"), atasand un header special inainte de a trimite in mediul fizic (de pilda, protocolul "MAC – Media Access Control" ce arata accesul la canalul de comunicare si modul de adresare al datelor acceptate, astfel impunand un mediu comun de transfer intre terminale). De asemenea, un atribut important il constituie si protocoalele de tip "tunneling", ce ofera o compatibilitate intre doua retele sau terminale ce nu utilizeaza acelasi tip de protocol (spre exemplu, comunicarea dintre o rețea cu IPv6 si una cu IPv4). \*

***3. Diferente majore intre modelul TCP si modelul OSI***

In acest domeniu de activitate cele mai importante arhitecturi de proiectare a unei retele sunt TPC/IP si OSI, desi exista mult mai multe modele.

Modelul OSI reprezinta un sablon conceptual de interconectare a terminalelor in rețea, ce tinde sa asigure toate beneficiile mentionate anterior la TCP/IP (comunicare prin transport de date, canal de transmisiune cu o rata a erorii scazuta etc.). Denumirea provine de la acronimul "Open Systems Interconnection" ce a fost un proiect initiat de catre "Internation Organization for Standardization" in scopul obtinerii unui model teoretic pe baza caruia se pot proiecta retelele de calculatoare.

In ultimul timp, desi ambele modele prezinta acelasi mecanism de functionare pe baza protocoalelor clasificate in nivele bine definite, OSI este preferat in dezvoltarea standardelor de comunicatie, cercetare in domeniul hardware al retelisticii si analiza conceptuala pentru o imbunatatire constanta, pe cand TCP/IP a demonstrat o aplicabilitate mult mai mare in lumea reala datorita protocoalelor sale caracteristice. \*

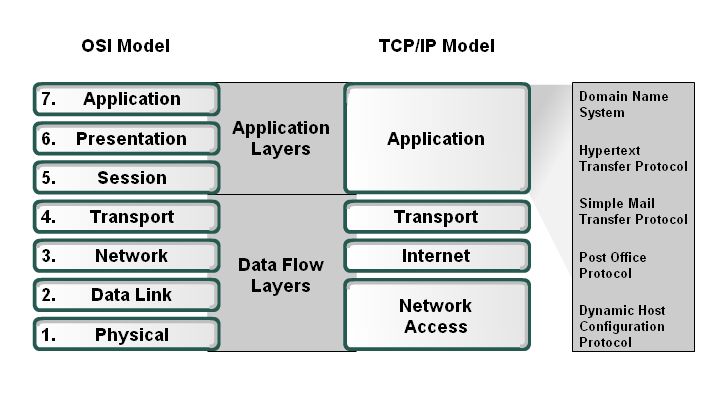


Fig 6. Diferenta de nivele dintre modelul OSI si TCP/IP

In categoria deosebiri fac parte dupa cum se observa si in fig 6, numarul de nivele, utilizarea protocolului UDP in modelul TCP , dar si Nivelul de Transport prezent in modelul OSI.O alta deosebire signifianta consta in faptul ca primele trei nivele ale modelului OSI au fost compresate intr-unul singur la TCP/IP, indeplinind functiile necesare stabilirii unei conexiuni in rețea.

Oricat de diferite ar parea cele doua modele la prima vedere, trebuie mentionat faptul ca ambele functioneaza prin mecanismul de "packet switching" ce are la baza protocoale asemanatoare ca scop. Asadar, in sarcina proiectarii unei retele, un inginer ar trebui sa aiba cu sine modelul ISO drept referinta teoretica deoarece dispune de o plaja mai ampla a posibilitatilor, insa ar trebui sa foloseasca modelul TCP/IP in cazul practic, datorita flexibilitatii si fiabilitatii sale.

***4. Nivelul de Transport/Transport Layer***

Cum in prima parte a acestei lucrari am prezentat modelul PCT/PI si diferite concepte utilizate in retelistica, putem prezenta si intelege mult mai bine Nivelul Transport.

Transport Layer-ul functioneaza utilizand protocoale, cel mai important dintre acestea fiind TCP-ul care actioneaza sub principiul "conection-oriented transmissions".

Cercetatorii ce au avut ca scop proiectarea acestui nivel, pe langa modelul de referinta OSI, au avut in vedere si serviciile de care va dispune, vrand sa imbunatateasca transportul pachetelor de date intre terminale (teoretic, comunicarea este doar o "mutare" de biti intre experditor si destinatar, insa practic, este un proces mult mai complex, fiind necesare masuri de precautie in vederea aparitiei problemelor precum aglomerarea canalului de transmisie, spre exemplu). Aceste servicii sunt conectate direct cu Nivelul de Tranrport cu ajutorul unei aplicatii ce dispune de o interfata programabile, astfel permitand specialistului sa editeze diversi parametrii:

*I) Serviciul de incredere* ("reliability"): un serviciu ce doreste asigurarea transportului total al datelor (informatia se poate pierde pe drum datorita zgomotului de pe canal sau a altor factori externi), astfel ca se poate efectua o detectie a erorii la receptionarea unui pachet, pentru a vedea daca acesta este corupt sau nu, ulterior trimitand un semnal de "acknowledgment" (ACK) daca s-a primit cu succes, sau "not acknoelwdment" (NACK) daca s-au intampinat probleme.\*

*II) Serviciul de comunicare orientata dupa conexiune*("connection-oriented communication"): exprima ideea discutata si in capitolele anterioare, despre faptul ca transmiterea pachetelor de date este mult mai sigura daca structura acestora nu este modificata iar calea este deja cunoscuta, in contradictie cu modelul de transmisie al datagramelor (un transfer de baza de tipul packet-switching, unde timpul pana la destinatie, ordinea bitilor si livrarea pachetului nu sunt garantate de catre rețea) impus de catre protocolul "User Datagram Protocol" \*

*III) Serviciul de asigurare a livrarii pachetelor in ordine* ("same order delivery"): este proiectat cu scopul de a micsora rata probabilitatii bucatirii pachetelor de date pe traseul retelei, iar in cazul folosii protocolului UDP, de a evita problema de "head-of-line blocking" (atunci cand unui port i se cuvin doua bucati din pachete de date diferite, transmise catre acesta in acelasi timp, ceea ce provoaca un efect de "bottleneck" asupra retelei, micsorand perfomantele acesteia). \*

*IV) Serviciul de rata transferului datelor* ("flow control"): un principiu bazat pe ideea ca un expeditor poate trimite pachetele de date cu o viteza mult mai mare decat le poate receptiona cu bine si procesa expeditorul.[9]

*V) Serviciul de evitare a congestionarii retelei* ("congestion avoidance"): se bazeaza pe controlul traficului astfel incat, rețeaua sa nu se aglomereze. Evitarea efectului de "bottleneck" se poate face prin controlul ratei transferului de date sau prin diverse tehnici, precum "slow-start" ce compenseaza cererile repetate de tip "automatic repeat request" (ARQ).\*

*VI) Serviciul de multiplexare* ("multiplexing"): un mecanism ce se refera la faptul ca un terminal poate expedia si trimite date in acelasi timp, utilizand deodata toate porturile sale aferente (fiind astfel o operatie de "multiplexare").\*

Cu ajutorul acestor servicii, acest nivel, isi indeplineste cu succes sarcina de a livra pachetele de date catre destinatari, ceea ce implica o multiplexare a informatiilor provenite de la diferite procese ale aplicației in cauza, adaugandu-i numerele porturilor unde ar trebui sa ajunga, alaturi de alte informatii relevante, in header-ul specific. Un lucru important de mentionat este faptul ca, anumite protocoale (TCP, spre exemplu, dar nu si UDP) dispun si de un mecanism abstract, ce poarta denumirea de "circuite virtuale" ("virtual circuit"), care reprezinta un proces de transmisiune a datelor intr-o asemenea maniera incat creaza iluzia existentei unui strat fizic de legatura intre sursa si destinatie. Astfel, sirul de biti este transportat, in timp ce ideea de "pachet" este oarecum ascunsa fata de procesul aferent aplicației in cauza (stream-urile de biti se numeste "segmente" in cazul de fata).

"User Datagram Protocol", functioneaza pe baza unui principiu simplu si eficient, asigurand transportul pachetelor numite "datagrame" in cazul nostru (adica livrarea la timp, ordinea bitilor si realizarea transportului complet nu sunt garantate), cu o viteza mult mai mare, astfel servind in aplicatii precum "streaming-ul" unui video pe internet sau ascultarea unui post radio prin browser-ul de navigare online.

***5. Protocoalele principale ale Nivelului de Transport***

Cele mai importante protocoale folosite de catre niveleul de trasport sunt TCP si UDP. In continuare vom detalia aceste protocoale, dar si altele, in functie de importanta lor.

*5.1 Protocolul TCP*

"Transmission Control Protocol", reprezinta baza Nivelului de Transport din punct de vedere al protocoalelor, acesta apartinand modelului PCT/PI, oferind servicii de livrare in ordine si verificarea bitilor transmisi intre terminalele din cadrul retelei.

Fiind necesara o metoda de comunicare intre terminalele din cadrul unei retele, a lluat nastere acest protocol la inceputul anului 1974.

Dupa cum am mentionat si in capitolele anterioare, protocolul TCP functioneaza pe baza idee de "connection-oriented transmission" (transmisiunea datelor in ordinea lor originala, asigurand livrarea si stabilind initial in canal de conexiune intre terminalele ce comunica), fiind nivelul dintre aplicație si "Internet Protocol" (IP). Atunci cand o aplicație doreste a efectua o transmisiune de pachete (compuse din octeti de date), si nu se recurge la mecanismul de bucatire a acestuia in dimensiuni si format precum a adreselor IP (metoda rar utilizata), se genereaza o cerere catre protocolul TCP, acesta intrand in functiune dupa cum se dicteaza. De obicei, pachetele ce urmeaza a fi trimise sunt alcatuite dintr-un "body" (continand informatia propriu-zisa pe care protocolul IP o transmite), caruia i se adauga niste "headers" ce ajuta la descrierea detaliilor precum sursa, destinatia si controlul informatiei. De asemenea, tinand cont de faptul ca un pachet de date poate fi pierdut ("lost"), duplicat ("duplicated") sau livrat in alta ordine ("delivered out-of-order"), protocolul TCP tinde sa anticipeze aceste probleme prin emiterea cererilor de retransmitere a datelor, re-aranjarea bitilor in cadrul pachetelor sau optimizarea traseului dintre sursa si destinatie, incercand decongestionarea retelei. \*

Drept mecanism de functionare, TCP accepta datele provenite dintr-un nivel superior, le imparte in pachete carora le adauga un "header" caracteristic, astfel creand un "segment TCP", ce urmeaza fi si el incapsulat la randul sau intr-o datagrama impusa de "Internet Protocol", ulterior facandu-se transferul cu acesta intre terminale.\*

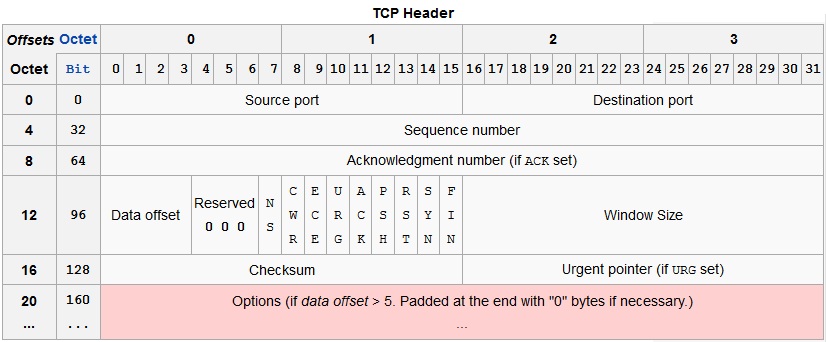


Fig 7. Protocolul TCP-header

Precum putem vedea in imaginea de mai sus, "header-ul" specific protocolului TCP contine informatii cruciale in transmiterea informatiei, precum portul sursa de unde se transmit datele ("Source Port"- adresa pe 16 biti), portul destinatie ("Destination Port" – adresa pe 16 biti), numarul de secvente ("Sequence Number" – dimensiune de 32 biti, cu rol dublu: daca fanionul "SYN" este egal cu 1, atunci acesta este numarul initial de secventa, sau in caz contrat, daca "SYN"=0, atunci acesta este numarul de secvente acumulate; "numarul secventei" adunat cu "1" este practic numarul recunoscut in cadrul procedurii de "ACK" atunci cand se face verificarea daca terminalul destinatie a primit pachetul sau nu), numarul de recunoastere ("Acknowledgement Number" – pe 32 de biti, despre care s-a discutat in paranteza anterioara), decalajul de date ("Data Offset" – pe 4 biti, specificand dimensiunea header-ului TCP in cuvinte de 32 biti, dimensiunea minima a acestuia putand fi de 5 cuvinte, respectiv maxima de 15), biti rezervati ("Reserved" – 3 biti, in cazul unei utilizari viitoare, de obicei setati pe zero), fanioane ("Flags" – 9 biti, existand 9 fanioane ce servesc scopuri diverse, precum "NS", "CWR" cu rol in decongestionarea canalului de transmieie, "ECE", "URG", "ACK", "PSH", "RST" pentru resetare, "SYN" despre care am vorbit anterior avand roluri in sincronizarea datelor, "FIN" ce indicata faptul ca nu mai sunt date de trimis de la sursa), dimensiunea ferestei de la destinatie ("Window Size" – 16 biti), verificarea erorilor ("Checksum" – 16 biti), pointer urgent ("Urgent Pointer" – 16 biti, daca fanionul "URG" este setat pe "1", atunci acest camp devine un decalaj pentru numarul de secventa ce indica ultimul octect urgent de date), si alte campuri pentru optiuni ("Options" – variaza intre 0 si 320 de biti). \*

*5.2 Protocolul UDP*

"User Datagram Protocol", este un protocol de baza al Nivelului de Transport, asemenea cu protocolul TCP.

UDP foloseste principiul "connectionless" (datele sunt aranjate intr-o alta ordine decat cea initiala, rutarea facandu-se cat mai rapid, in defavoarea optimizarii acesteia), ce nu prezinta mecanisme de tipul "hand-shaking", astfel compensand cu o viteza mult mai mare in comparatie cu TCP-ul. De asemenea, acest protocol nu garanteaza receptionarea pachetelor de fiecare data cand are loc o transmisiune, nu se ocupa de cele de tip "duplicated", insa utilizeaza serviciul de verificare a erorilor pentru integritatea informatiei si numerotarea porturilor destinatie.Putem spune ca UDP-ul ajuta in transmiterea pachetelor de date numite "datagrame" catre alte terminale, folosind "Internet Protocol" precum TCP, fara a fi necesara o conexiune stabilita apriorii, servind in aplicatii unde verificarea amanuntita pentru erori nu este necesara. \*

Protocolul UDP prezinta un set de caracteristici pentru care este preferat in proiectarea anumitor aplicatii, precum faptul ca este "transaction-oriented" (ofera raspunsuri rapide protocoalelor de tip "Domain Name System"), functioneaza prin transfer de datagrame (ajuta in modelarea protocoalelor de tipul "Remote Procedure Call" sau in metoda de "IP tunneling" mentionata in capitolele precedente), viteza de transmisie (potrivit pentru un numar foarte mare de clienti), functionarea avantajata de caile unidirectionale si lipsa de efectuare a cererii de retransmitere. Asadar, desi nu garanteaza transmiterea datelor de fiecare data, acesta compenseaza prin atributele enuntate mai sus, fiind potrivit pentru alte tipuri de aplicație. "User Datagram Protocol" functioneaza pe acelasi principiu de atribuire a pachetului cu informatie un "antent" specific, insa putin diferit fata de cel al lui TCP, acesta ilustrand doar numarul portului sursa ("Source Port Number"), numarul portului destinatie ("Desination Port Number"), lungimea "header-ului" si a datelor transmite ("Lenght") si algoritm de verificare a erorilor ("Checksum| – camp optional in cazul transmisiei prin IPv4 si oblicatoriu pentru IPv6).

*5.3Protocolul RSVP*

Rolul acestui protocol este de a se ocupa de alocarea resurselor in rețea, prescuratrea provenind de la "Resource Reservation Protocol. La baza acestuia se afla protocoalele IPv4 si IPv6 din Nivelul de Internet, asigurand rezervarea resurselor intr-un mod ce evita congestionarea retelei si timpii de asteptare indelungati, asemeni lui "Internet Control Message Protocol" (ICMP), fiind descris in lucrarea RFC 2205. Protocolul RSVP poate fi utilizat atat de terminalele expeditoare, cat si de cele destinatare, definind astfel modalitatea in care aplicatiile rezerva sau elibereaza datele o data ce tranzactia s-a incheiat cu succes. In zilele noastre, RSVP-ul este rar intalnit in forma sa initiala, insa extensiile sale, precum RSVP-TE devin din ce in ce mai intalnite in comunicatiile de zi cu zi. [13]

Drept caracteristici principale, putem observa faptul ca RSVP-ul se distinge prin cererile de alocare a resurselor intr-o singura directie (de la expeditor catre destinatar), prin faptul ca nu este un protocol de rutare la baza (insa noile sale extensii tind a indeplini si aceasta sarcina), suporta ideea schimbarilor dinamice ce au loc intr-o rețea printr-un mecanism de "soft state" ce impune o reimprospatare a resurselor blocate de fiecare nod si dispune de cateva moduri de rezervare a datelor, ceea ce permite o flexibilitate enorma in proiectarea retelei. De asemenea, acesta dispune si de atribute precum raportarea de erori sau informatii despre sirul de date generat de catre RSVP. \*

RSVP-ul functioneaza dupa doua principii bine stabilite: "flowspect" si "filterspect". Primul principiul se ocupa cu rezervarea resurselor pentru un stream de date, ce este identificat pe baza adresei destinatie si calitatea serviciului ceruta de catre aplicație (denumita despre "flowspec", "specificarea stream-ului", neputand intelege cererea eliberata de catre QoS). Un "flowspec" se poate defini fie drept clasa unui serviciu, rezervarea resurselor (determinata de catre QoS) sau specificatiile traficului de date. Cel de-al doilea principiu are ca rol definirea pachetelor ce sunt afectate de catre "flowspec-ul" mentionat anterior, putand fi de tip fix (rezerva resursele pentru stream-ul respectiv), partajat (pune la dispozitie resurse comune pentru mai multe stream-uri) sau de tip "wildcard" (rezerva un tip de resursa doar pentru anumite stream-uri). O cerere emisa catre RSVP, continand un "flowspec" si un "filtersper"se numeste "flowdescriptor", ceeea ce ajuta la descrierea parametrilor principali ai stream-ului de informatii.

*5.4 Protocoalele DCCP*

Datagram Congestion Control Protocol reprezinta functia ce asigura transferul optim al datelor in rețea, decongestionarea traficului de informatii.

DCCP-ul prezinta o idee centrala de a avea acces la mecanismele de decongestionare a canalului de transmisiune fara a fi necesara implementarea acestora in Nivelul de Aplicație. Putem spune faptul ca, pe langa caracteristicile sale de baza, le imprumuta si pe cele ale protocolului TCP (fara cea de a transmite bitii in ordinea sa originala), insa nu suporta concentrarea asupra mai multor stream-uri de date precum "Stream Control Transmission Protocol" (SCTP). Asadar, putem afirma faptul ca protocolul DCCP este util in implementari precum streaming-ul video, suportul jocurilor online, VoIP sau alte aplicatii ce prezinta un numar mare de terminale client. \*

Acest protocol lucreaza atat cu chei, ca si cu pachete de date, oferind utilizatorului posibilitatea de a vedea daca informatia a fost receptionata de destinatar.

***6. Concluzii***

Conform celor prezentate putem ajunge la urmatoarea conlcuzie, TCP/IP, poate fi interpretat ca o trusa de unelte ce ajuta la proiectarea diverselor retele.

Despre Nivelul de Transport putem spune ca este o componenta importanta pt o buna functionare a unei conexiuni, asigurand transmitearea datelor, o ordine in trimiterea si receptionarea acestora prin atasarea "header-elor".

In opinia noastra, pentru un proiectant, modelul OSI poate reprezenta o parte teoretica, iar modelul TCP/IP fiind aplicarea partii teoretice, in felul acesta obtinandu-se o performanta ridicata in transmiterea datelor din rețea, deci a retelei propriu zise.

***7.Bibliografie***

* Computer network - Andrew Tanenbaum
* RFC 1123, "Requirements for Internet Hosts – Application andSupport*"*, R. Braden (ed.), October 1989"RSVP Project".
* USC Information Science Institute. Retrieved 2011-02-16RFC 4340 -
* Datagram Congestion Control Protocol"Stream Control Transmission
* Protocol". IETF. October 2000. RFC 2960
* fig 1: http://www.rasfoiesc.com/educatie/informatica/retele-calculatoare/RETELE-DE-CALCULATOARE-SUPORTU56.php
* fig 2: http://cristis.ro/computere/p200200/internet.html
* fig 3: http://www.competentedigitale.ro/internet/internet\_TCP\_IP.html
* fig 4: http://www.scrigroup.com/afaceri/economie/Aplicatii-clientserver-de-sala53513.php
* fig 5: http://cs.stanford.edu/people/eroberts/cs181/projects/2010-11/FreedomOfInformationChina/wp-content/uploads/2011/05/ip\_address\_structure1.gif
* fig 6: https://rotorwanker.wordpress.com/2010/10/25/ccna-semestrul-1-modulul-3/
* fig 7: http://en.wikipedia.org/wiki/Transmission\_Control\_Protocol