

Lucrare de laborator №2.

Studiul procesului încapsulării și decapsulării în rețelele de comunicații, în baza simulatorului rețelistic Cisco Packet Tracer

Protocoalele sunt folosite pentru a face posibilă comunicarea într-o rețea de la un calculator la altul. Un protocol este un set de reguli și convenții cu ajutorul cărora se realizează comunicarea într-o rețea. Protocoalele determină formatul, timpul, secvențele și controlul erorilor în comunicarea de date. Pentru ca datele să călătorească de la sursă la destinație, fiecare nivel al modelului OSI pentru calculatorul sursă trebuie să comunice cu nivelul pereche al calculatorului destinație. Această formă de comunicație se numește peer-to-peer.

În timpul acestui proces protocoalele fiecărui nivel schimbă informații, numite PDU (protocol data unit). Fiecare nivel depinde de serviciile nivelului inferior. Pentru a oferi acest serviciu, nivelul inferior folosește încapsularea pentru a pune datele nivelului superior în câmpul său de date, apoi adaugă un header și un trailer pentru ca nivelul să-și îndeplinească funcțiile. Datelor ce coboară de-a lungul nivelelor modelului OSI le sunt adăugate mai multe headere și trailere.

În cadrul lucrării date, utilizând softul aplicativ Cisco Packet Tracer se va analiza procesul de încapsulare și decapsulare a datelor în fram-uri de comunicare.

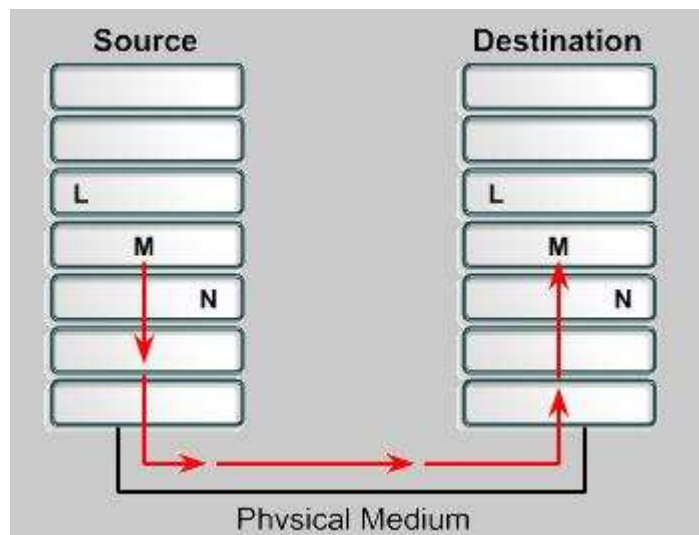
Scopul: de a se învăța de a analiza procesele în rețea cu Cisco Packet Tracer și GNS

Perioada executării lucrării de laborator: 90 minute.

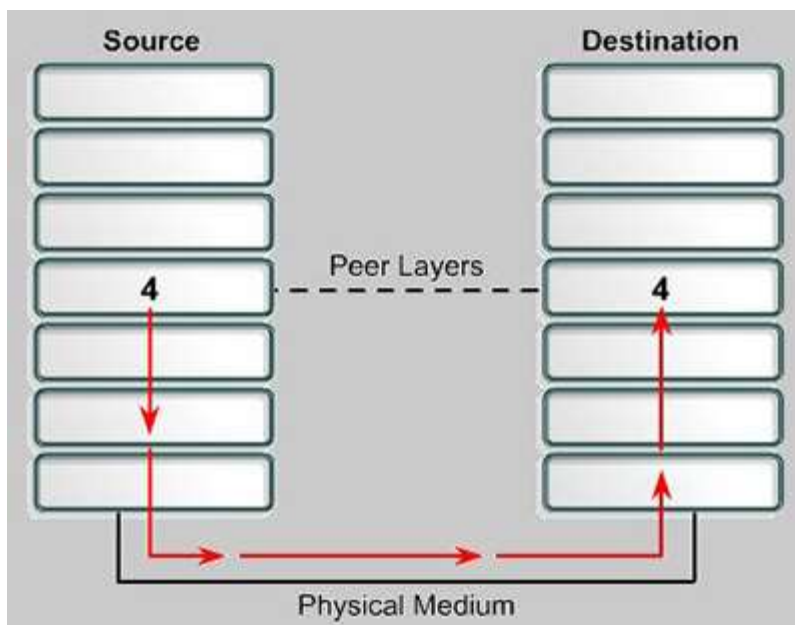
Material teoretic: Protocoale de rețea

Protocoalele controlează toate aspectele comunicării datelor, printre care și:

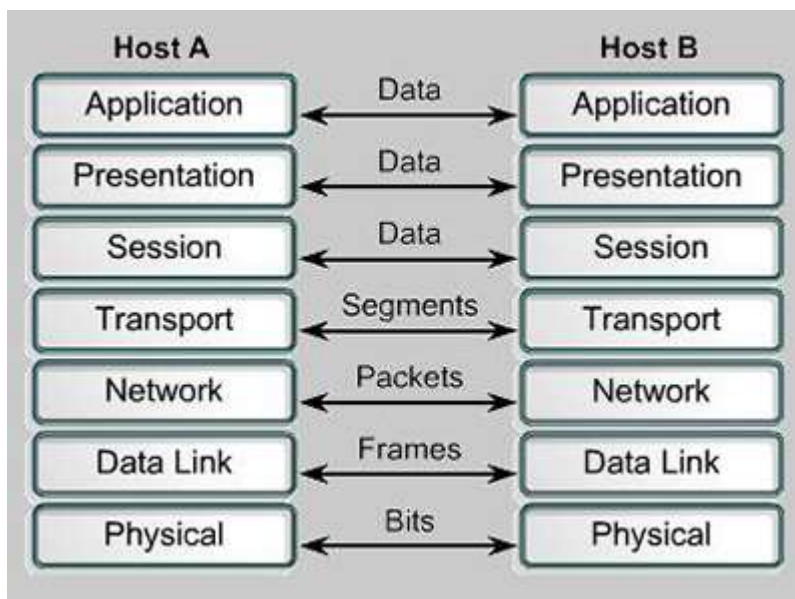
- cum este construită rețeaua fizică;
- cum sunt conectate calculatoarele la rețea;
- cum sunt formate datele pentru transmisie;
- cum sunt transmise datele;
- cum sunt conectate erorile.
-



Pentru ca datele să călătorească de la sursă la destinație, fiecare nivel al modelului OSI pentru calculatorul sursă trebuie să comunice cu nivelul pereche al calculatorului destinație. Această formă de comunicație se numește peer-to-peer.

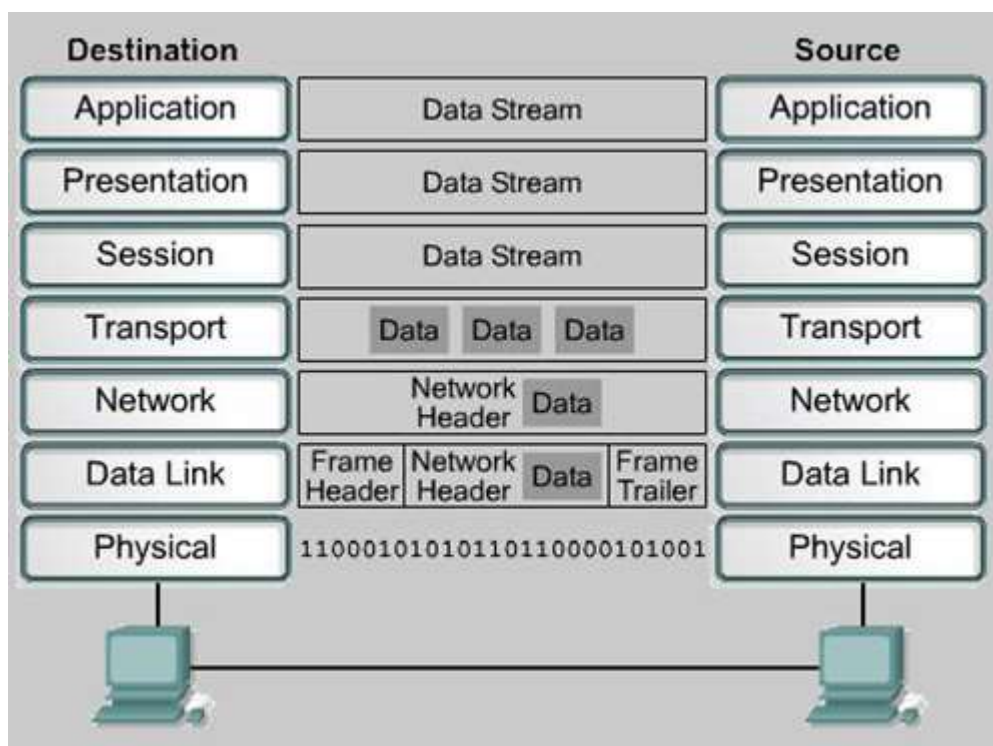


În timpul acestui proces protocoalele fiecărui nivel schimbă informații, numite PDU (protocol data unit). Fiecare nivel depinde de serviciile nivelului inferior. Pentru a oferi acest serviciu, nivelul inferior folosește încapsularea pentru a pune datele nivelului superior în câmpul său de date, apoi adaugă un header și un trailer pentru ca nivelul să-și îndeplinească funcțiile. Datelor ce coboară de-a lungul nivelelor modelului OSI le sunt adăugate mai multe headere și trailere.



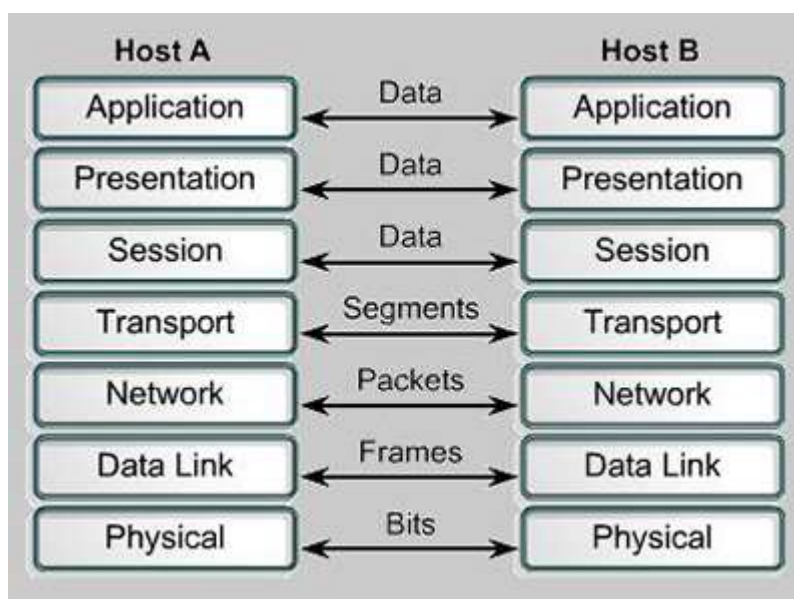
După ce nivelele 7,6 și 5 și-au adăugat propriile informații, nivelul 4 grupează datele sub formă de segmente. Nivelul rețea oferă servicii nivelului superior, transport, având sarcina să transfere datele în rețea. Pentru a-și îndeplini această sarcină el atașează un header ce cuprinde informații privind sursa și destinația adreselor logice, creând astfel pachete. Nivelul legătură de date oferă servicii nivelului

rețea, încapsulând informația într-un frame. Header-ul frame-ului conține informații privind adresa fizică, necesară îndeplinirii funcțiilor nivelului legătură de date. Nivelul fizic oferă servicii nivelului legătură de date, codând frame-urile în secvențe de biți 0 și 1, în vederea transmiterii lor.

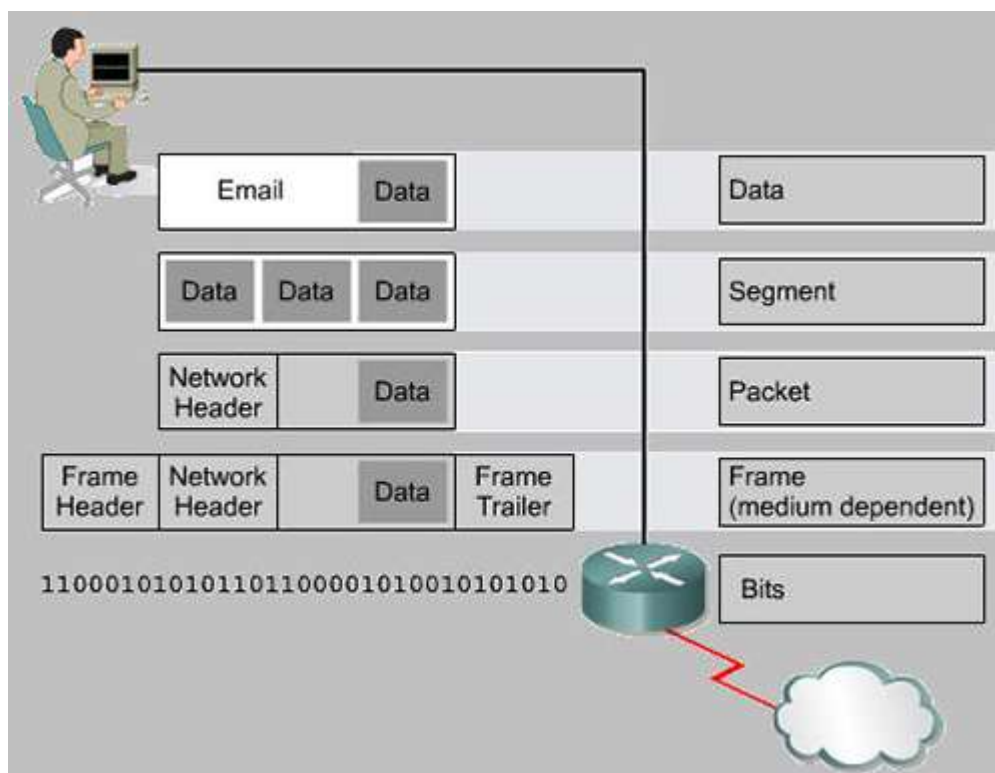


Procesul de încapsulare a datelor

Dacă se dorește ca un calculator A să transmită date calculatorului B, datele trebuie mai întâi împachetate printr-un proces numit încapsulare.



Încapsularea combină datele cu informațiile protocolului necesar înainte de transmiterea lor în rețea. În timp ce datele se deplasează în jos de-a lungul nivelelor modelului OSI, ele primesc header-e, trailer-e și alte informații.



Procesul de încapsulare a datelor pentru IPv4

La încapsularea datelor se urmează cinci pași de conversie:

1. Crearea datelor

Când un utilizator trimite mesaje e-mail, caracterele alfanumerice sunt convertite la date ce pot traversa rețeaua.

2. Împachetarea datelor pentru transport

Datele sunt împachetate pentru transportul în rețea. Folosind segmente, funcțiile de transport asigură comunicarea calculatoarelor prin sistemul e-mail.

3. Adăugarea adresei IP

Datele sunt puse într-un pachet ce conține un header cu adresele logice ale sursei și destinației. Aceste adrese ajută componentele de rețea să trimită pachete de-a lungul rețelei prin alegerea unei căi.

4. Adăugarea header-ului și trailer-ului la nivelul legătură de date

Fiecare componentă de rețea trebuie să pună pachetul într-un frame. Frame-ul asigură conexiunea cu următoarea componentă de rețea.

5. Conversia datelor în biți pentru transmiterea lor în rețea

Frame-urile trebuie convertite în biți de 1 și 0 pentru transmiterea acestora în mediu. Anumite funcții permit componentelor să distingă acești biți ce traversează mediul folosit în conectarea rețelelor într-o rețea.

Procesul de încapsulare a datelor pentru IPv6

În IPv6, antetul pachetului și procesul de forwardare a pachetelor de către rutere au fost simplificate. Deși antetele de IPv6 sunt de 2 ori mai mari decât cele de IPv4, procesarea pachetelor este mai eficientă:

- antetul de IPv6 a fost simplificat prin mutarea câmpurilor rar folosite în antete de extensie opționale.

- ruterele IPv6 nu fragmentează pachetele. Gazdele cu IPv6 trebuie ori să descopere dimensiunea maximă a pachetelor acceptate de toate ruterele de pe cale, să fragmenteze pachetele la capetele transmisiunii sau să folosească un MTU de 1280 de octeți, care este minimul acceptat de protocol;

- antetul IPv6 nu este protejat de o sumă de control, făcându-se presupunerea că integritatea datelor este asigurată atât la nivelul legătură de date, cât și la nivelul transport. În acest fel, ruterele IPv6 nu trebuie să recalculeze suma de control atunci când schimbă unul din elementele pachetului

- câmpul TTL din IPv4 a fost redenumit în Hop Limit, deoarece ruterele nu mai trebuie să calculeze timpul petrecut de pachet în cozile dispozitivelor de rețea.

Lucrarea practica 1-1.

Procesul de rutare. Configurare default gateway

Pentru ca un dispozitiv să poată comunica în afara rețelei locale, în altă rețea, trebuie să cunoască modul în care ajunge la acea rețea. Acest mod este specificat prin rute. O rută descrie unui dispozitiv ce cale să urmeze pentru a ajunge la rețeaua destinație.

Procesul de rutare constă în primirea unui pachet, investigarea tabeli de rutare, descoperirea rutei corespunzătoare și rutarea pachetului, adică transmiterea pachetului mai departe conform rutei.

Rutele se găsesc în tabela de rutare și constau din două elemente:

- partea de match: adresa de rețea destinație (adresă și mască de rețea)
- partea de acțiune: următorul dispozitiv de rutare (next hop) (sau interfața de ieșire)

În momentul în care un dispozitiv care rutează (un ruter) primește un pachet, extrage adresa IP destinație, localizează adresa de rețea destinație în tabela de rutare și apoi rutează (dirijează, retransmite) pachetul către următorul ruter (next hop). Procesul se reia pe următorul ruter până când pachetul ajunge la destinație.

Rutarea este un proces care are loc la nivelul 3 (Rețea) din stiva OSI, lucrând cu adresa IP.

În topologia <[lab02_default_gw.pkt](#)>¹ se află două rețele interconectate printr-un ruter.

Observați că există conectivitate între ruter și oricare stație; folosiți comanda {ping} de pe ruter sau mesajele din Packet Tracer.

În Cisco IOS primul pachet trimis poate să nu fie de fapt trimis din cauza absenței tabeli ARP. De aceea când trimiteți un pachet în Packet Tracer la sau de la ruter, prima oară nu va funcționa. Următoarele pachete, însă, vor funcționa.

Observați că **nu** există conectivitate între stații din rețele diferite, deși adresele IP și măștile de rețea sunt configurate pe ruter și pe toate stațiile. Nu există conectivitate pentru că o stație nu are configurată rută către rețeaua cealaltă. Pentru a asigura conectivitate între stații din rețele diferite, va trebui să adăugăm, pe fiecare stație, rută către cealaltă rețea având ca next hop ruterul. Vom realiza în continuare acest lucru – conectivitatea între toate stațiile.

Pentru a asigura conectivitate între toate stațiile, indiferent din ce rețea fac parte trebuie să configurăm o rută implicită (*default gateway*) pe fiecare stație. Pașii următori vor fi:

¹ <https://drive.google.com/file/d/1PrskyCrHyZN0kaLcdhpgfqUTGARUHBCp/view?usp=sharing>

1. Determinăm adresele IP ale interfețelor ruterului.
2. Pentru fiecare stație, adăugăm o rută implicită (de tip *default gateway*) având ca next hop adresa IP a interfeței ruterului din rețeaua din care face parte stația.
3. Folosim comanda {ping} de pe ruter sau mesajele din Packet Tracer pentru a vedea conectivitatea între oricare două stații, chiar aflate în rețele diferite.

Pentru început, vrem să aflăm adresele IP configurate pe fiecare interfață a ruterului. Dați click pe [Router0], și mergeți pe tab-ul [CLI], unde aveți acces la consola ruterului, și rulați comenzile:

Router>enable					
Router#show ip interface brief					
Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	172.16.100.1	YES	manual	up	up
FastEthernet1/0	192.168.100.1	YES	manual	up	up

Interfața [FastEthernet0/0] este conectată la rețeaua din care fac parte stațiile [PC0] și [PC1]. Adresa IP a acestei interfețe este [172.16.100.1]. Pentru a asigura conectivitate cu stațiile din cealaltă rețea, pe stațiile [PC0] și [PC1] vom configura o rută implicită (de tip *default gateway*) având ca next hop adresa [172.16.100.1].

Pentru a configura *default gateway*, faceți click în Packet Tracer pe stația dorită (PC0 sau PC1), mergeți pe tab-ul [Desktop] și apăsați pe iconița [IP Configuration]. În zona de text cu eticheta [Default gateway] completați adresa [172.16.100.1].

În mod similar, configurați pe stațiile PC2 și PC3 adresa [192.168.100.1] (adresa interfeței [FastEthernet1/0]) pe post de *default gateway*.

Observați conectivitatea între oricare două stații indiferent de rețeaua de care aparțin.

Lucrarea practică 1-2.

Antetul pachetelor în procesul de rutare

Vrem să urmărim antetul pachetelor în timpul procesului de rutare. Vom trimite un pachet între stații din rețele diferite și vom observa modificarea antetului.

Folosiți în continuare topologia <[lab02_default_gw.pkt](#)>. Intrați în modul [Simulation] și trimiteți un mesaj Packet Tracer de la stația [PC0] la stația [PC2]. Urmăriți antetul celor două pachete transmise:

- pachetul trimis de stația [PC0] și primit de ruter (îl vom denumi pkt1)
- pachetul trimis de ruter și primit de stația [PC2] (îl vom denumi pkt2)

Pentru a urmări antetul unui pachet folosiți lista de evenimente din dreapta spațiului de lucru (Event list); faceți click pe pătrățelul colorat al fiecărui eveniment din listă.

Observați următoarele efecte ale procesului de rutare.

- Pachetul [pkt1] are ca adresă MAC sursă adresa MAC a stației [PC0] iar ca adresă MAC destinație adresa MAC a interfeței [FastEthernet0/0] a ruterului.
- Pachetul [pkt2] are ca adresă MAC sursă adresa MAC a interfeței [FastEthernet1/0] a ruterului iar ca adresă MAC destinație adresa MAC a stației [PC2].
- Se decrementează câmpul [TTL] (*Time To Live*) din antetul IP în pachetul [pkt2] față de pachetul [pkt1].

- Adresa IP sursă respectiv adresa IP destinație rămân nemodificate între cele două pachete.

În concluzie, procesul de rutare folosește adresele MAC specifice fiecărei rețele locale în care este trimis pachetul, lasă nemodificate adresele IP (sursă și destinație) și decrementează câmpul [TTL] (un contor al numărului de hop-uri).

Decrementarea câmpului [TTL] are rolul de a atenua efectele unor posibile bucle de rutare. În momentul în care câmpul [TTL] ajunge la valoarea {0} pachetul este aruncat (*dropped*).

Lucrarea practica 1-3. Rute și pachete de reply

Vrem să urmărim de ce sunt necesare rute pe ambele stații care comunică din rețele diferite. Vom folosi în continuare topologia <[lab02_default_gw.pkt](#)>.

Intrați în modul [Simulation] și trimiteți un mesaj Packet Tracer de la stația [PC0] la stația [PC2]. Observați drumul parcurs de mesaj. Pachetul urmează drumul [PC0]→ ruter →[PC2]→ ruter →[PC0]. Este vorba de un pachet transmis și un pachet de răspuns (de tip *reply*).

Ștergeți de pe stația [PC2] configurația de [*Default Gateway*] și, în modul [Simulation], trimiteți încă o dată un mesaj Packet Tracer de la stația [PC0] la stația [PC2]. Observați drumul parcurs de mesaj. Observați că pachetul este aruncat (*dropped*) de stația PC2. De ce se întâmplă acest lucru și nu se trimite pachet de răspuns (de tip *reply*)?

Înainte de a finaliza exercițiul, refaceți configurația de [*Default Gateway*] pe stația [PC2].