



# Internet

## Modele arhitecturale de referință



# Cuprins

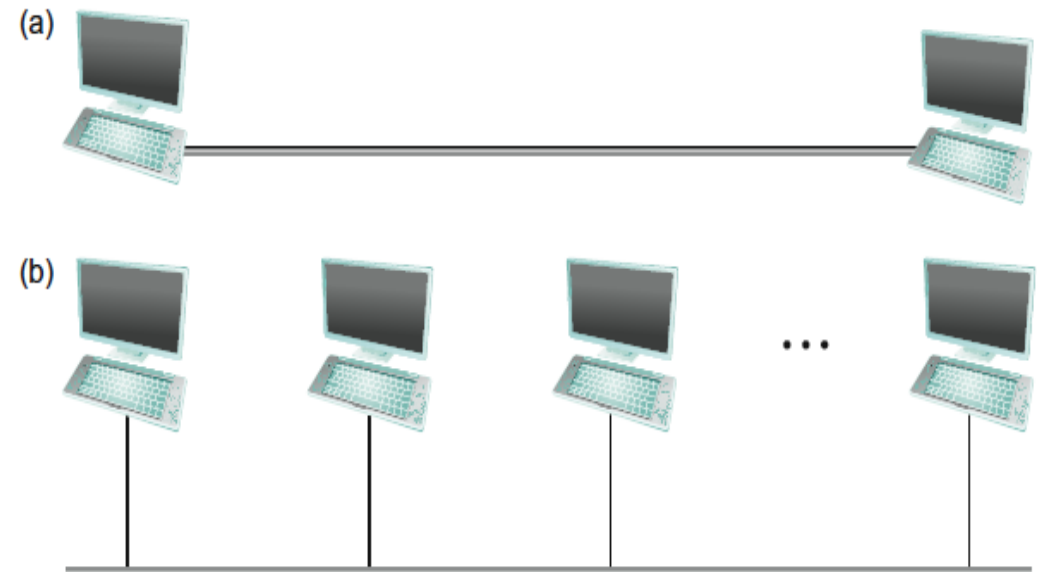
- Arhitectura rețelelor de calculatoare
- Modelul de interconectare a sistemelor deschise - ISO-OSI
- Rolul ierarhiei de protocoale
- Formatul datelor – antet si continut
- Servicii si primitive de serviciu
- Modelul TCP/IP

## Obiective:

- Notiuni fundamentale utilizate in restul cursului
- Imagine de ansamblu a protocoalelor ce vor fi studiate

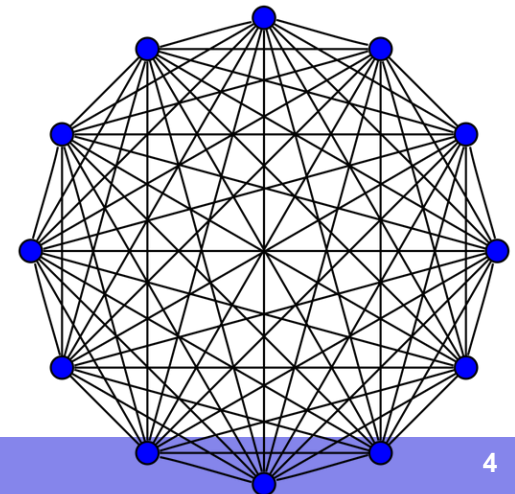
# Comunicarea in aplicatii Internet

- Internetul este cunoscut de utilizatori prin aplicatiile sale
  - e-Mail
  - Transferul de fisiere
  - Web
  - si multe altele
- Bazate pe **comunicarea** prin legaturi intre calculatoare
- **Legatura** poate fi directa, intre doua sau mai multe calculatoare (noduri)
  - **punct la punct** – prin fire
  - **acces multiplu** – de ex wireless (retele celulare sau Wi-Fi)



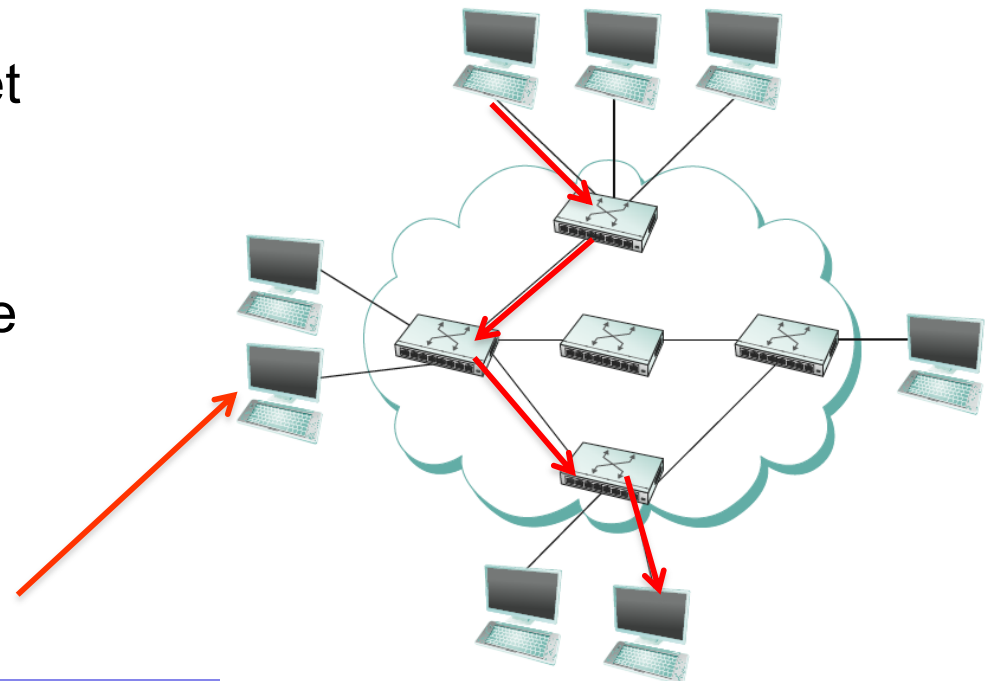
# Legatura de date

- Legaturile asigura functionalitati specifice **datelor** (diferite de cele pentru comunicari vocale)
  - **codificarea** bitilor pentru transmisie si intelegerea codurilor la receptie
  - **detectia** erorilor de transmisie si **corectarea** lor
    - delimitarea sirurilor de biti (**incadrarea**) care constituie mesaje complete (**cadre**) si atasarea unor informatii de **control**
  - controlul **fluxului** de date
  - **controlul accesului** la media (pentru acces multiplu)
- Legaturile directe intre noduri  
**nu ofera scalabilitate**



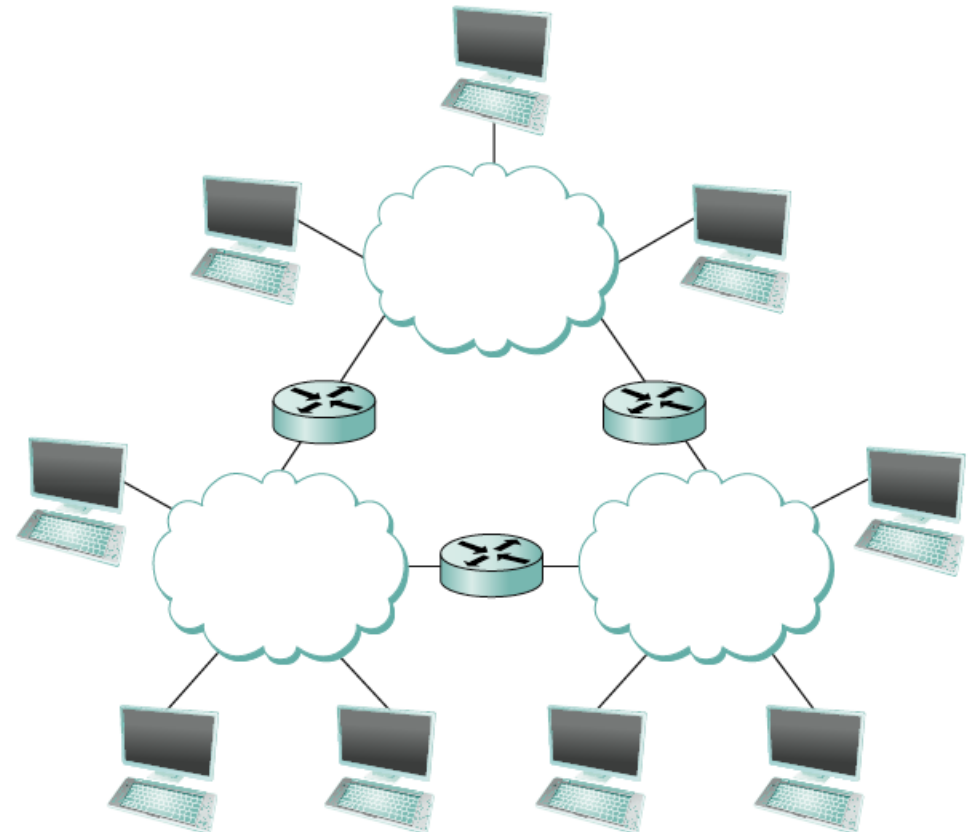
# Retele

- Conexiunea între două noduri se poate face și **indirect**, prin **noduri intermediare** (switch-uri)
  - atasate la mai multe legături
  - au funcții speciale
  - formează o rețea
- Funcționează după principiul sistemului **postal**; un nod
  - primește câte un bloc complet de date (**pachet**)
  - îl **memorează** temporar
  - îl **re-transmite** (**dirijează**) către destinație, printr-una din legăturile la nodurile vecine
- Nodurile din afara rețelei sunt numite **gazde** (ale aplicațiilor)
- Pentru ca rețeaua să funcționeze, fiecărui nod i se atasează o **adresă** de rețea pentru identificare
- Nodul **sursă** include în pachet **adresa** nodului **de destinație**



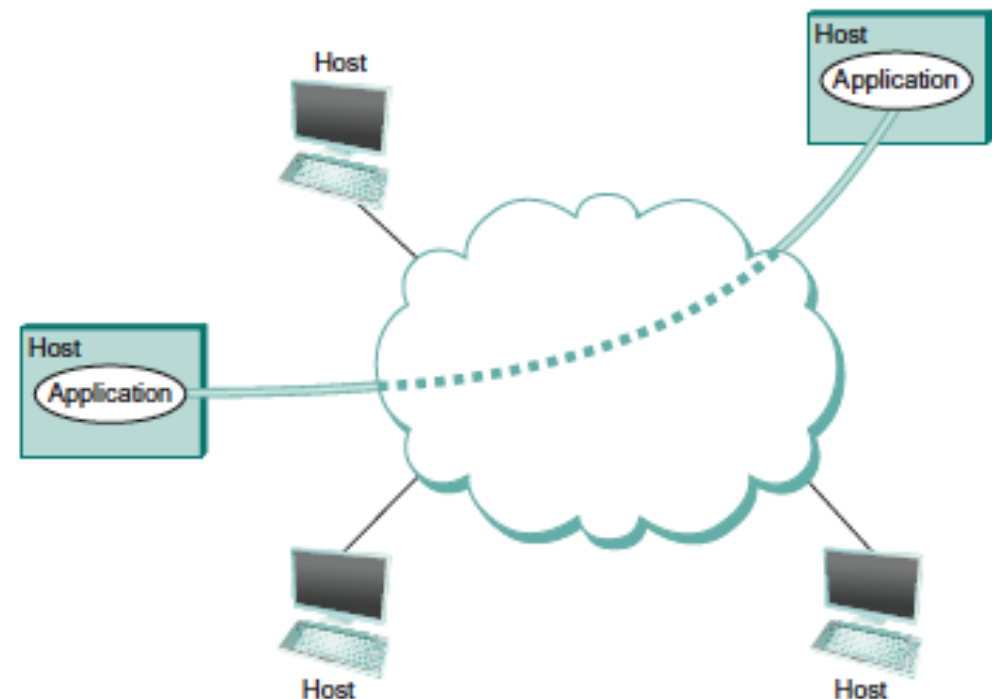
# Retele interconectate

- Nodurile intermediare pot fi organizate in mai multe rețele interconectate
- Un nod conectat la mai multe rețele este numit **ruter** (sau **gateway**)
  - are functii similare cu switch-ul
- Retelele se bazeaza pe buna functionare a **legaturilor** dintre noduri
- Retelele transmit pachete între oricare doua noduri neconectate direct
- Nu asigura **corectitudinea** transmisiei
  - e.g. pachete pierdute



# Comunicarea între aplicații

- Reteaua trebuie să asigure comunicarea **între aplicații** din calculatoare diferite, adică să ofere **canale "logice"** prin care **procesele** de aplicație să comunice între ele
- Aceasta presupune facilități suplimentare peste cele de rețea
- **identificarea** unică a fiecărui **capăt de** canal logic printr-o pereche (adresa de rețea, **port**)
- mecanisme de **trimitere / primire** de mesaje de către procese
- garantarea **recepției corecte** a mesajelor
- livrarea mesajelor în **ordine** în care au fost transmise
- pastrarea **confidențialității**
- etc.





# Cerintele aplicatiilor

- Modul in care este folosit un canal logic difera de la o aplicatie la alta
- In **Web**, comunicarea se face intre doua procese
  - la comanda unui utilizator, un proces **client** (browser) trimite o **cerere** catre un **server** Web – un mesaj care include identificatorul paginii dorite
  - procesul server trimite un raspuns catre client – un mesaj care contine pagina Web solicitata, pe care clientul o afisaza pe ecran
- In aplicatii de livrare de continut **audio/video**
  - transmiterea cererii este similara cu Web
  - livrarea este diferita
    - se trimite ca raspuns o serie de mesaje
    - continutul este livrat utilizatorului pe masura ce mesajele sunt primite
    - impune respectarea unor constrangeri de timp



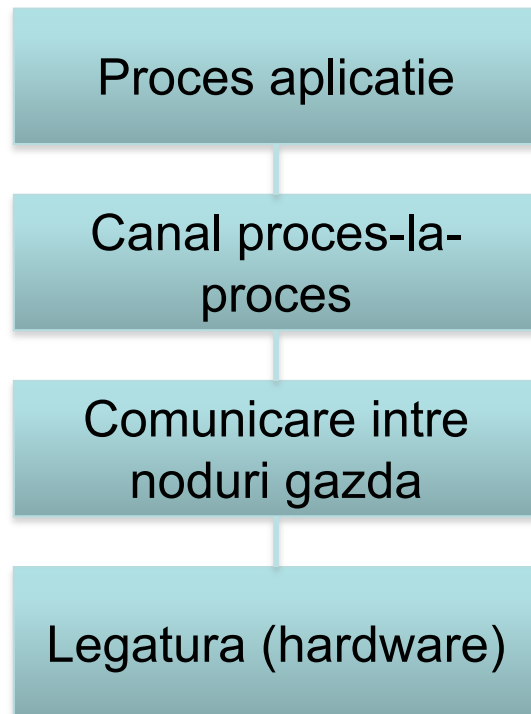


# Arhitectura Retelelor de Calculatoare

- **Arhitectura** descrie
  - modul in care **componentele** sunt organizate si
  - felul in care ele **interactioneaza**
- Retelele de calculatoare sunt sisteme complexe
- Abordarea lor ca **ansambluri de componente** logice simplifica intelegerea si realizarea
- Pentru **retele de calculatoare** avem componente pentru
  - comunicarea pe o legatura de date
  - comunicarea intre doua noduri gazda
  - comunicarea pe canale logice

# Arhitectura ierarhica

- Componentele care asigura comunicarea intre doua noduri gazda folosesc **serviciile** oferite de componentele pentru o legatura de date si adauga noi functionalitati (dirijarea, adresarea...)
- Arhitectura este o **ierarhie** de **nivele** functionale



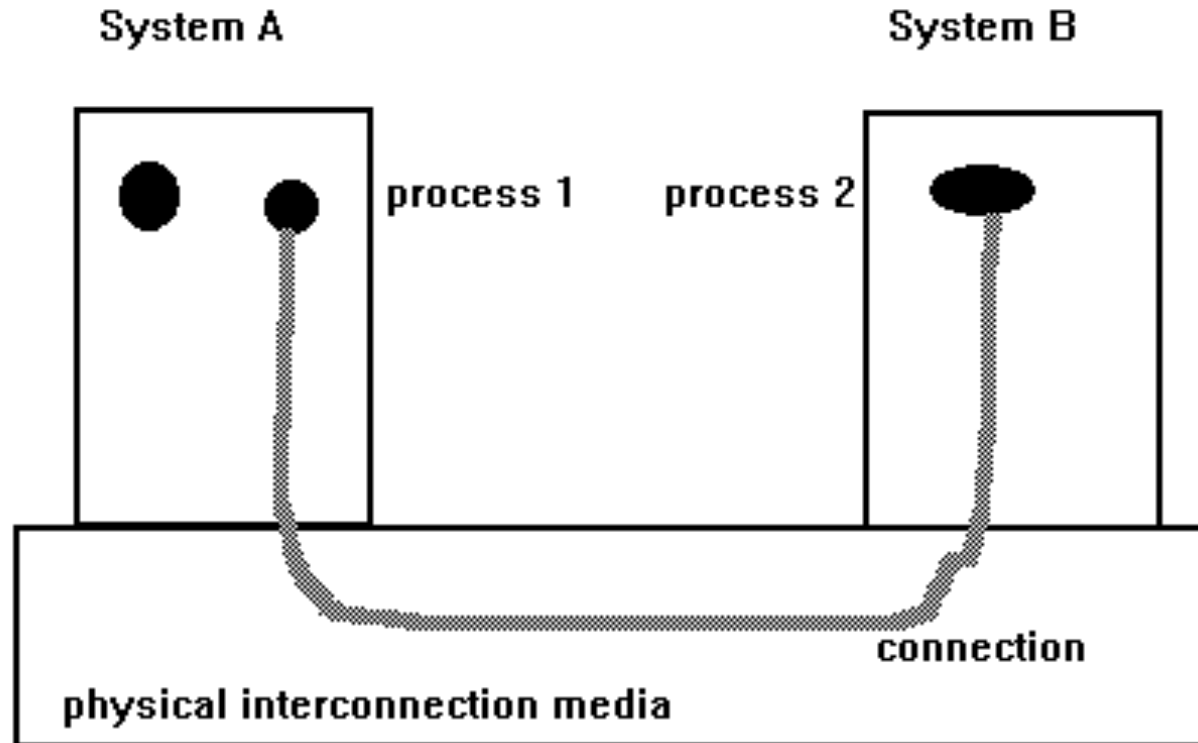
In figura se arata ierarhia componentelor dintr-un nod gazda

Fiecare **componenta** apartine unui **nivel** diferit

Fiecare nod gazda are o structura similara

# Componente de bază

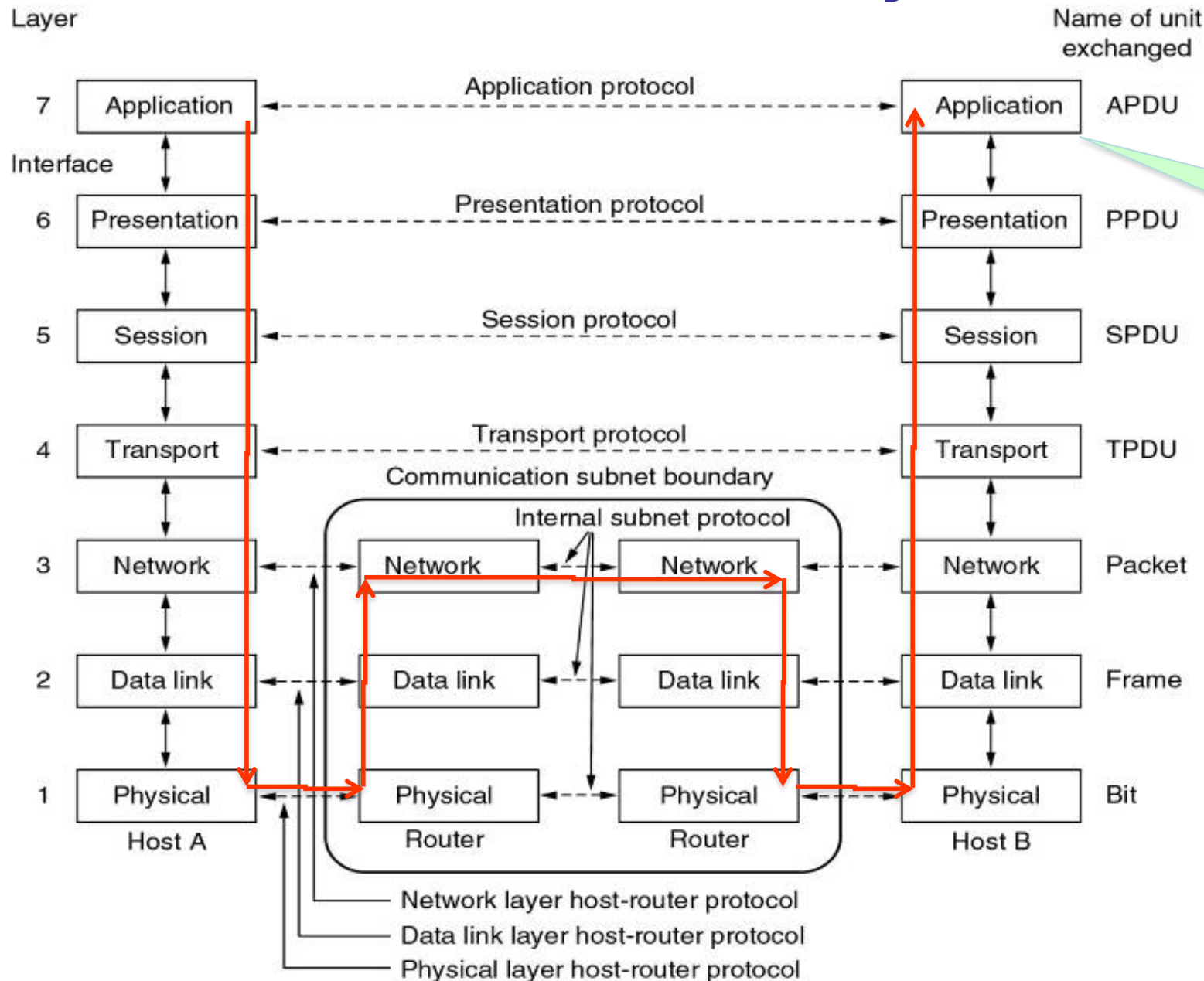
- Componente de baza ale unui model de comunicare:



**Protocol** – set de reguli respectate de parti pentru a comunica intre ele; se refera la

- **formatul** mesajelor communicate: continut + meta-descriere (antet)
- **operatiile** executate
  - trimite cerere conectare, primește raspuns, confirma raspuns etc.

# Modelul de Referință ISO OSI



entitate de protocol

- Fiecare entitate are legături
- cu alte entități din același nod (prin interfețe de serviciu) și
  - cu entități pereche din alte noduri (prin mesaje)



## Nivel fizic

- **Funcție** - transmitere a șirurilor de biți pe un canal de comunicație
- Principalele **probleme**
  - codificarea zerourilor și a unităților
  - stabilirea și desființarea conexiunilor fizice
  - modul de transmisie (semiduplex sau duplex) etc.
- **Exemplu**
  - 802.11 Wi-Fi

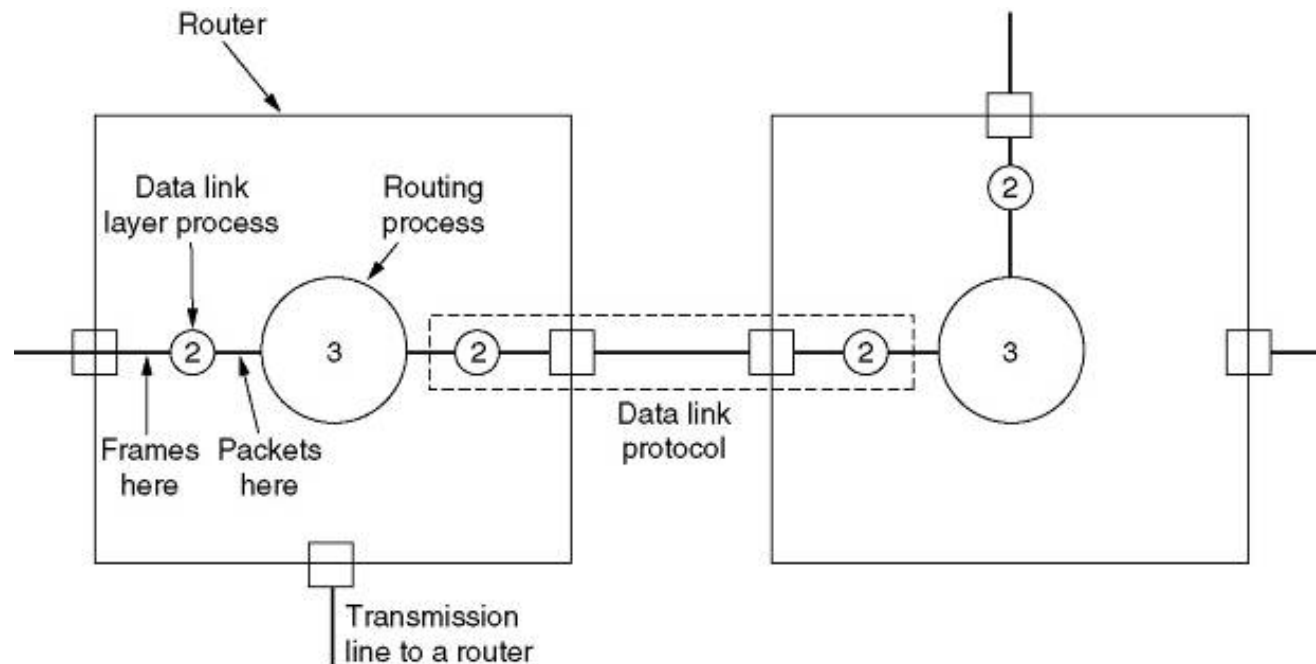
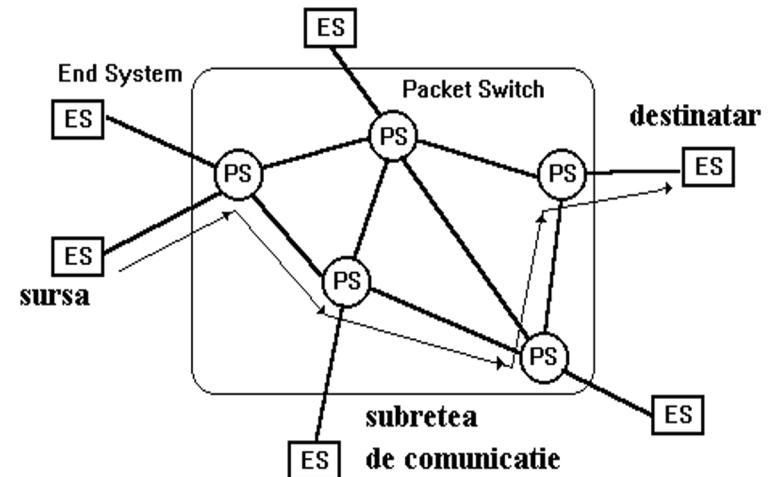


# Legătura de date

- **Funcție** - realizează **agregarea** unor siruri de biti in **cadre** si **comunicarea lor sigură** și **eficientă** între două noduri adiacente (conectate printr-un canal fizic de comunicație)
- **Probleme rezolvate**
  - Încadrare
  - Control erori
  - Control flux
  - Transmisie transparentă
  - Management legătură
- **Exemplu** de protocol: HDLC (High Level Data Link Control)  
**flag address command data FCS flag**
- **Implementare** prin
  - adaptoare de retea
  - drivere din sistemul de operare al calculatorului

# Nivel rețea

- **Funcție** - transmiterea pachetelor între oricare două noduri din rețea
  - prin dirijarea lor de la un nod la altul
- **Probleme rezolvate**
  - alegerea legăturii următoare (dirijarea)
  - adresarea
  - calculul tabelului de dirijare



# Nivel Transport

**Funcție** - asigura un transfer de date corect, eficient **între procese** din sistemul sursă și din sistemul destinar

## Oferă

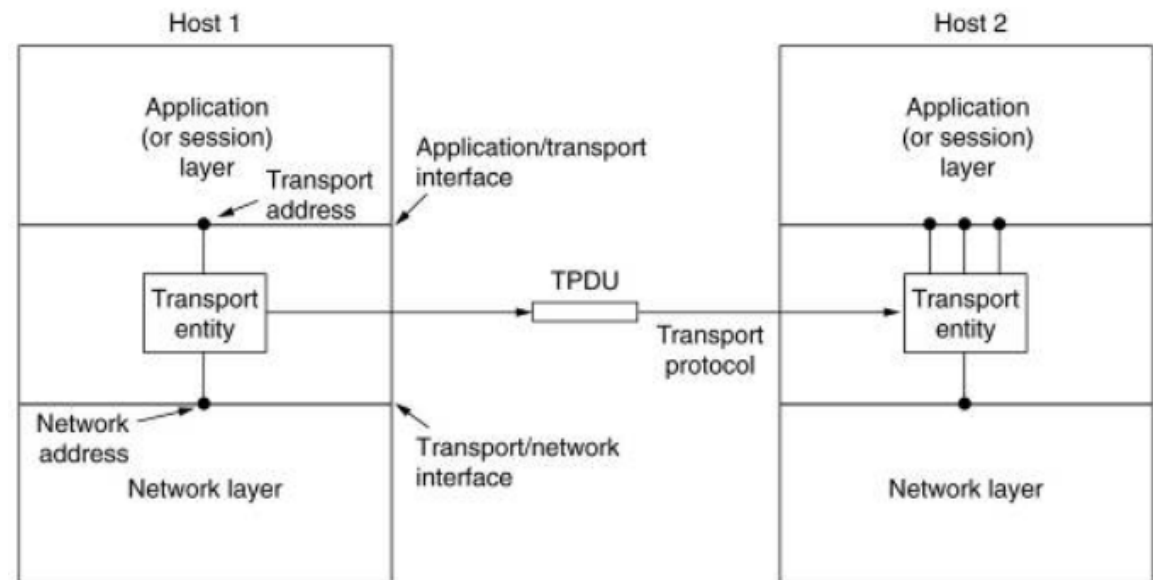
- un transfer sigur al datelor, chiar cu o rețea nesigură;
- o **interfață uniformă** pentru nivelul superior, independent de tipul rețelei utilizate.

## Separă două categorii de nivele

- furnizorul serviciilor de transport (nivele 1-4)
- utilizatorul serviciilor de transport (nivele 5-7)

## Probleme

- gestiunea conexiunilor
- transferul datelor
- controlul fluxului
- adresarea







# Nivele superioare

- **Nivel Sesiune**

- Controlul dialogului intre aplicatii
- Sincronizarea transferurilor
- Stabilirea unor puncte de verificare si reluare a transferurilor

- **Nivel Prezentare**

- Conversia formatului datelor intre
  - sintaxa folosita de aplicatii si
  - sintaxa de transfer



## Nivel Aplicație

- Servicii comune unor categorii de aplicații
  - Mesagerie
  - Transfer de Fișiere
  - Terminal Virtual
  - Serviciu de Directoare

# Ierarhii de protocoale

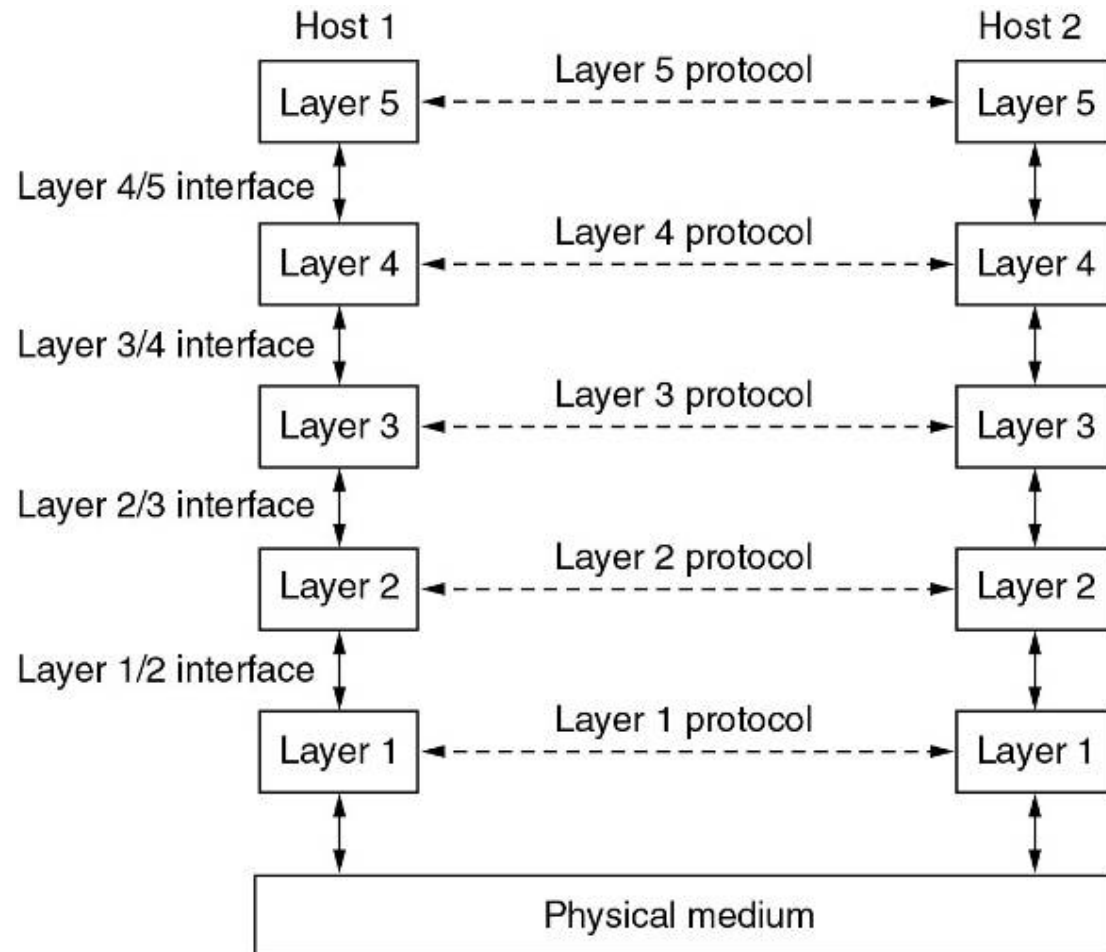
Complexitatea **conexiunii** → organizarea pe mai multe **nivele** cu functii distincte

- **arhitectura rețelei** = setul de nivele si protocoale
- **stiva de protocoale** = lista ordonata a protocoalelor folosite

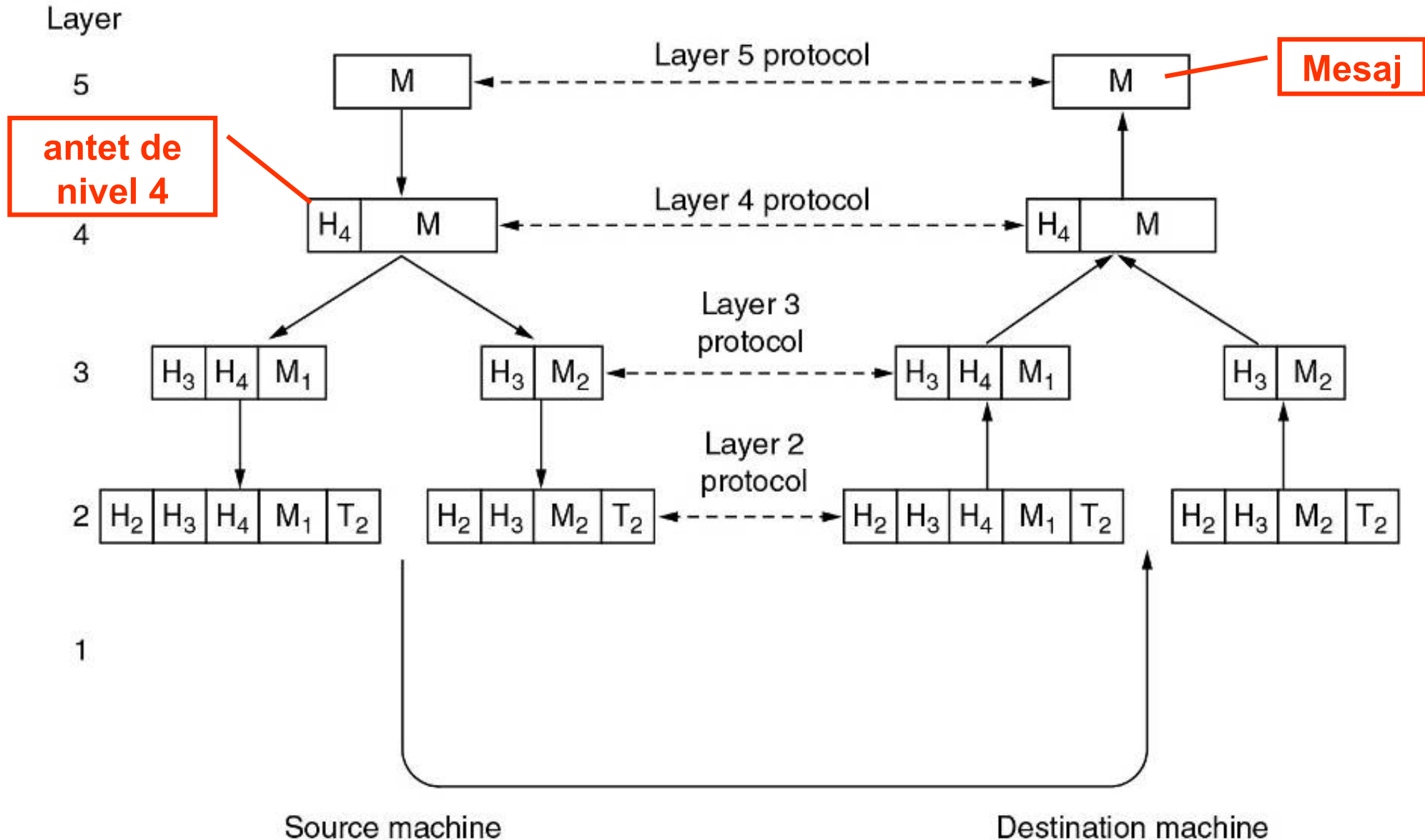
Interfata defineste **serviciul (operatiile primitive)** pe care un nivel il ofera nivelului de deasupra

Ex. de primitive:

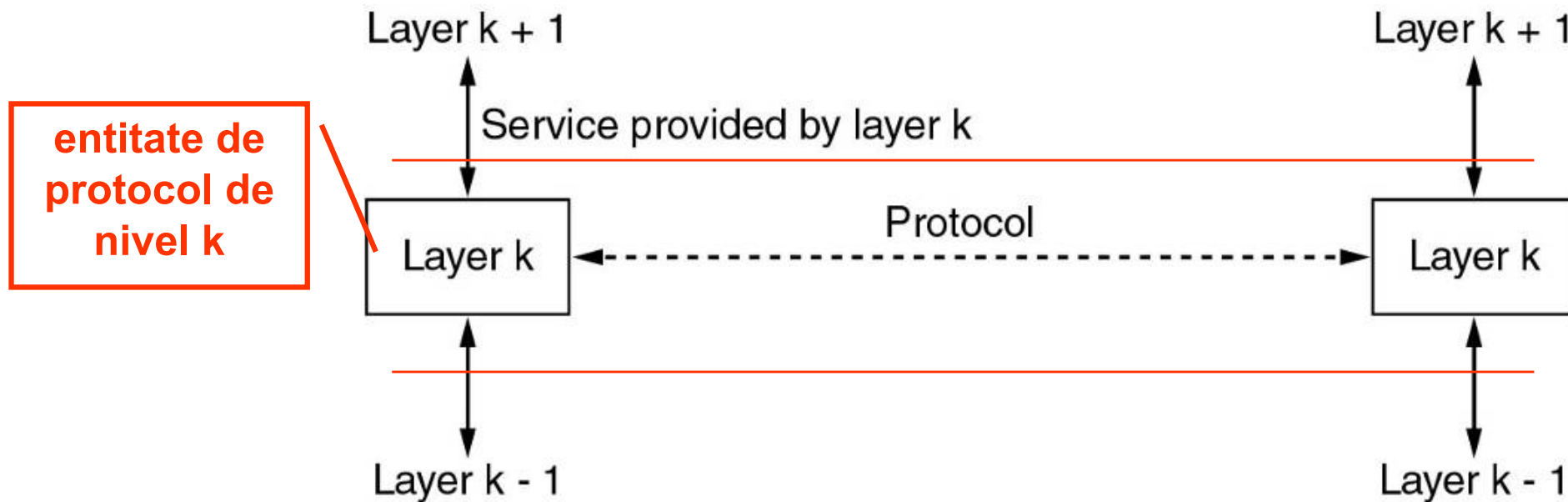
- connect
- accept
- send
- receive
- disconnect



# Flux de informație suportând o comunicare virtuală în nivelul 5



# Relația între servicii și protocoale



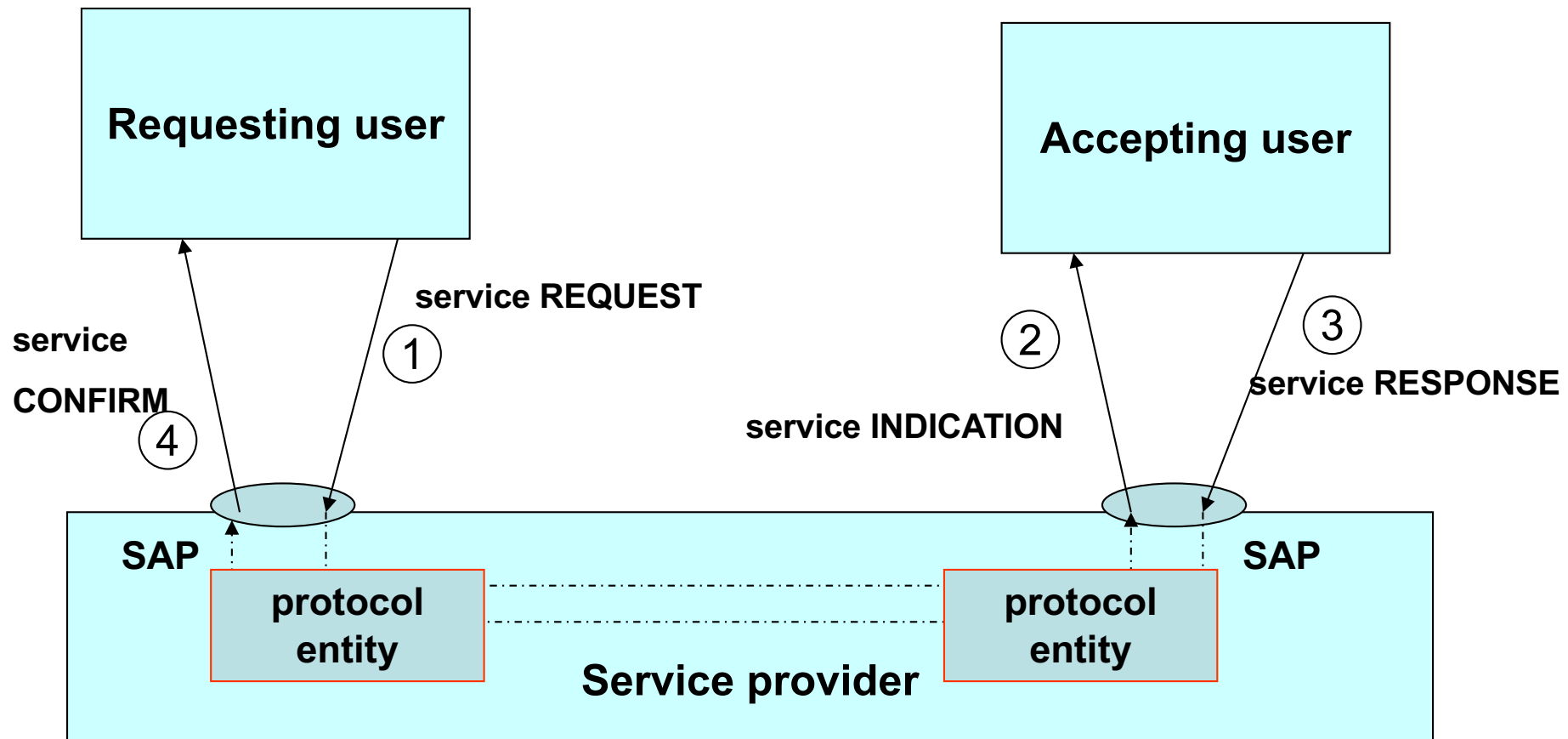
- entitățile de protocol din nivelul k
- colaborează folosind **protocolul** de nivel k
- folosind **serviciul** nivelului k-1
- pentru a furniza **serviciul** de nivel k
- **entitatilor** aflate pe nivelului k+1



# Primitive de serviciu

- Un serviciu este specificat de un set de **primitive** (operații accesibile utilizatorului serviciului)
- Patru clase de primitive
  - REQUEST cere un serviciu
  - INDICATION anunța primirea unei cereri
  - RESPONSE răspunde cererii
  - CONFIRM confirmă cererea

# Servicii confirmate

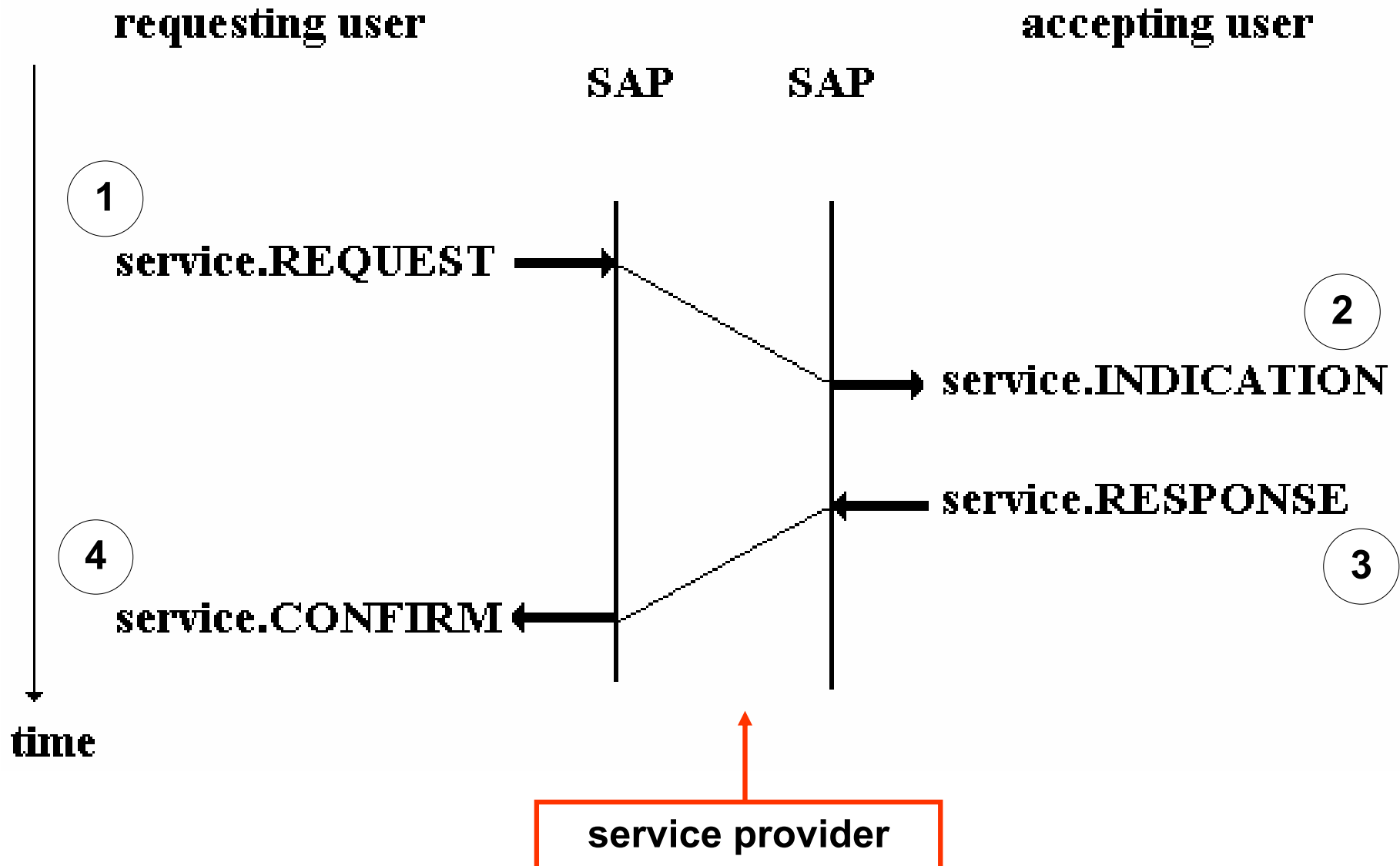


Entitatile de protocol pot inter-schimba mai multe mesaje pentru un singur serviciu confirmat

— ex.: mesaje de negociere a serviciului, de repetare la eroare etc.

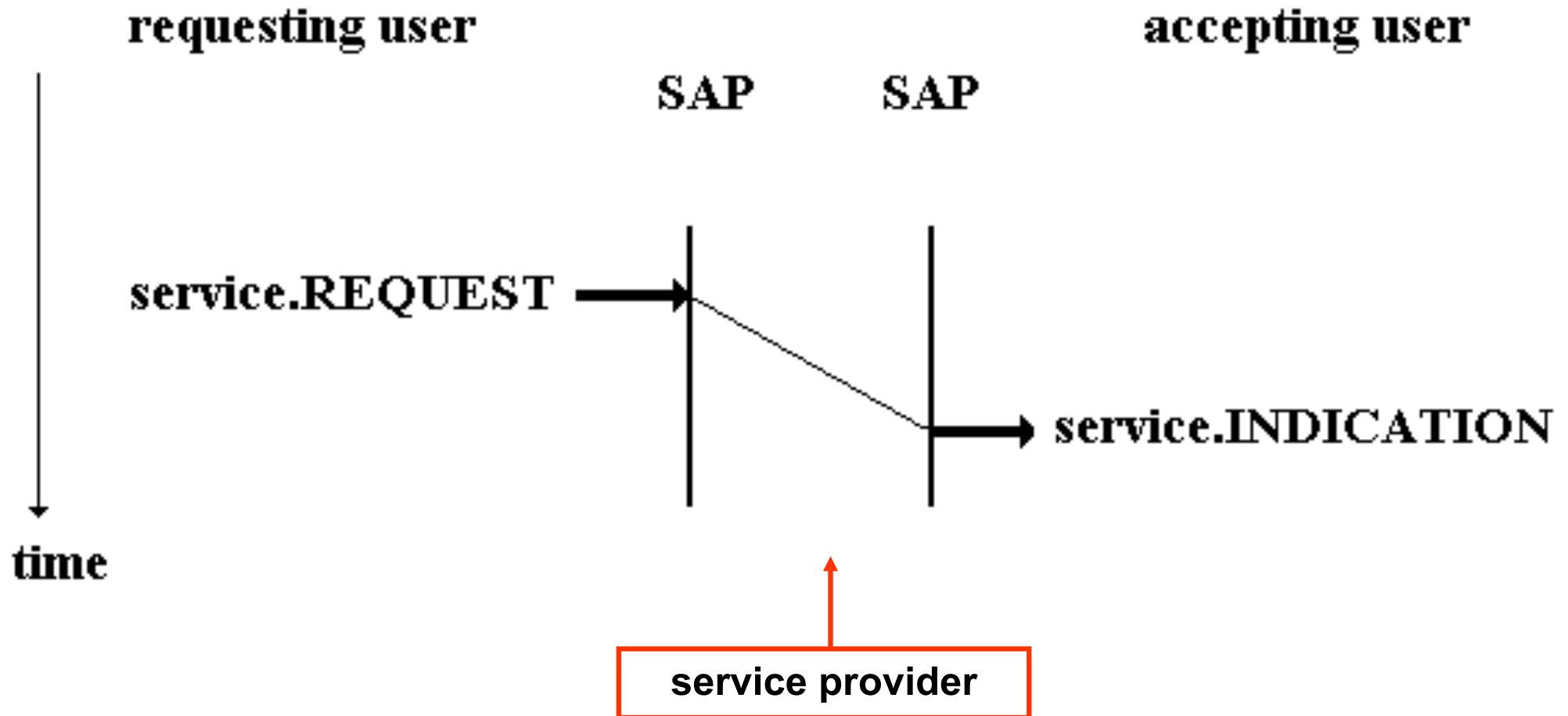
**SAP = Service Access Point**

# Servicii confirmate (o alta reprezentare)





# Servicii neconfirmate



The diagram illustrates a sequence of events over time. A vertical arrow on the left labeled 'time' points downwards. Two vertical lines represent the lifelines of two 'user' entities. Between these lifelines, a signal path is shown. It starts with a thick arrow labeled 'service.INDICATION' pointing left towards the first 'user' lifeline. This is followed by a step-like signal that rises and then falls, with a solid black dot at its peak. This signal path is labeled 'SAP' at both ends where it meets the lifelines. From the falling part of the signal, a thick arrow labeled 'service.INDICATION' points right towards the second 'user' lifeline. Below the diagram, a red box labeled 'service provider' has a red arrow pointing upwards towards the peak of the step-like signal.



## Mod orientat pe conexiune

utilizator solicitant

utilizator solicitat





# Ce conțin specificațiile ?

## Specificație Serviciu

- primitive (operații)
- parametri
- reguli asupra ordinii operațiilor (state machine)

## Specificație Protocol

- scop și funcții
- servicii oferite
- servicii utilizate din nivel inferior
- structura internă (entități și relații)
- tipuri și formate mesaje schimbate între entități
- reguli de reacție a fiecărei entități la comenzi, mesaje și evenimente interne

# Protocoloale OSI

**Physical layer:** V10, V11, V24, V35  
X.21, EIA RS-232-D  
MAC for LANs  
ISDN physical interface

**Data Link Layer:** HDLC LAP B for X.25  
LLC for LAN  
LAP D for ISDN

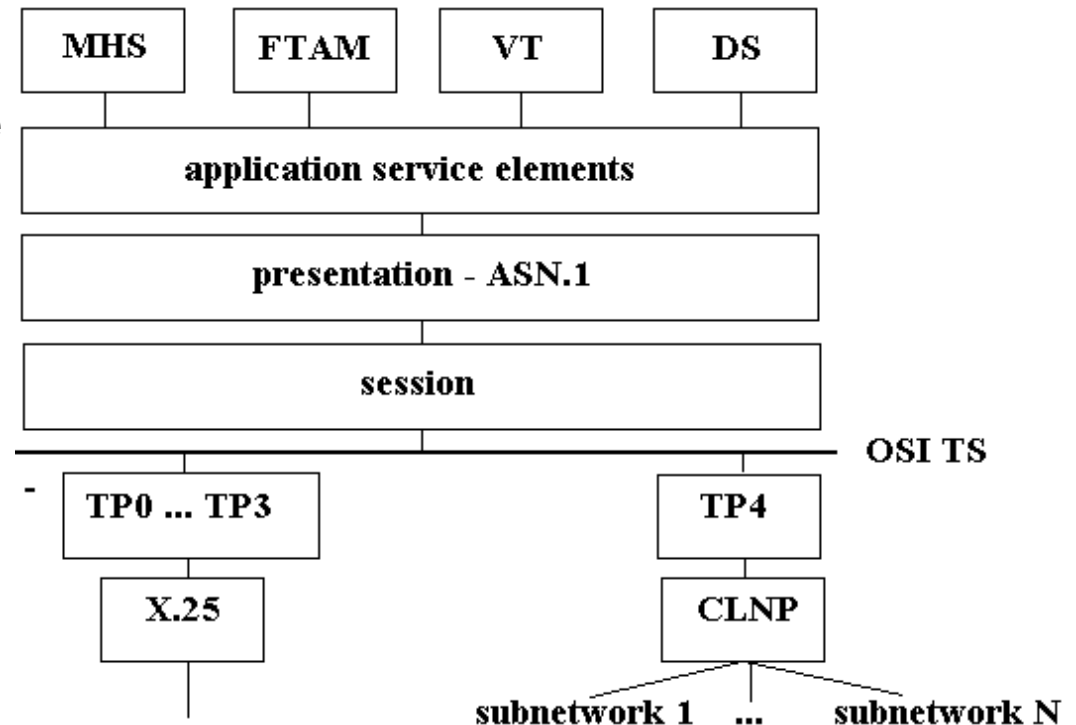
**Network Layer:** X.25, X.3, X.28, X.29  
CLNP

**Transport Layer:** TP0,..., TP4

**Session Layer:** session protocol

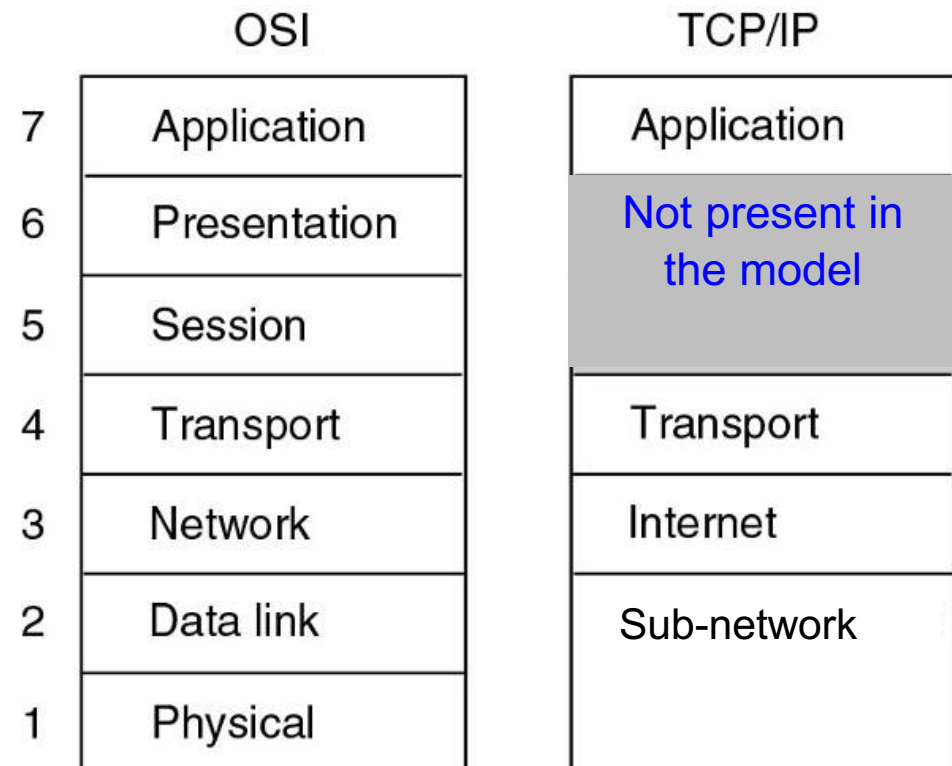
**Presentation Layer:** ASN.1 – Abstract Syntax Notation One

**Application Layer:** MHS - Message Handling System, X.400  
FTAM - File Transfer, Access, and Management  
VT - Virtual Terminal  
DS - Directory Services, X.500



# Modelul de referință TCP/IP

- Nivelul inferior este **sub-retea**
  - mare varietate de protocoale (ex. 802.11 Wi-Fi)
  - pot fi mai multe sub-nivele
  - modelul nu da detalii despre acest nivel
- Nivelul Internet
  - un singur protocol, IP
  - pentru rețele interconectate
- Nivelul Transport
  - TCP – canal sigur pentru siruri de octeti
  - UDP – canal nesigur pentru livrarea datagramelor (user datagram – sinonim pentru mesaje)



# Protocoloale în modelul TCP/IP

- Nivel Aplicație
  - Varietate de protocoale pentru transferul fișierelor și poștei, login la distanță, managementul rețelei

FTP	SMTP	Telnet	DNS	SNMP		
TCP			UDP		ICMP	EGP
IP						
ARP						
Ethernet	LLC 802.2			X.25	Packet Radio	
	MAC 802.3	MAC 802.4	MAC 802.5			
	Ethernet 802.3	Token Bus 802.4	TokenRing 802.5			



# Alte protocoale în modelul TCP

HTTP	HyperText Transfer Protocol
IIOIP	Internet Inter-Orb Protocol
WAP	Wireless Application Protocol
SOAP	Simple Object Access Protocol
LDAP	Lightweight Directory Access Protocol
SSL	Secure Sockets Layer
VPN	Virtual Private Networks
IPSEC	IP Security
PKI	Public Key Infrastructure
HTML	HyperText Markup Language
XML	Extensible Markup Language
WSDL	Web Services Description Language
UDDI	Universal Description, Discovery, and Integration





# Comparație OSI și TCP/IP

## Contra OSI

- Moment nepotrivit
- Tehnologie proastă
- Implementări rele
- Politici proaste

## Contra TCP-IP

- Nu distinge între servicii, interfețe, protocoale
- Nu este un model general
- “Nivelul” gazdă-rețea nu este un nivel
- Nu menționează nivelele fizic și legătură de date
- Protocoale minore bine înrădăcinate - greu de înlocuit



# Test

Un sistem are o ierarhie de protocoale organizate pe 6 niveluri. Aplicatiile genereaza mesaje avand dimensiunea de 45 de octeti. La fiecare nivel este adaugat un antet de 20 de octeti.

Ce fractiune din latimea benzii este ocupata de antete?

(a) ~ 64 %

(b) ~ 69 %

(c) ~ 73 %

Aratati calculul facut.

Puteți evalua eficiența acestei aplicații?



# Studiu individual

A. S. Tanenbaum Rețele de calculatoare, ed 4-a, BYBLOS 2003

- 1.4 MODELE DE REFERINȚĂ

- 1.4.1 Modelul de referință OSI
- 1.4.2 Modelul de referință TCP/IP
- 1.4.3 O comparație între modelele de referință OSI și TCP
- 1.4.4 O critică a modelului și protocoalelor OSI
- 1.4.5 O critică a modelului de referință TCP/IP

A. S. Tanenbaum Computer networks, 5-th ed. PEARSON 2011

- 1.4 REFERENCE MODELS

- 1.4.1 The OSI Reference Model
- 1.4.2 The TCP/IP Reference Model
- 1.4.3 A Comparison of the OSI and TCP/IP Reference Models
- 1.4.4 A Critique of the OSI Model and Protocols
- 1.4.5 A Critique of the TCP/IP Reference Model