King Fahd University of Petroleum & Minerals
Computer Engineering Dept

COE 342 – Data and Computer Communications

Term 021

Dr. Ashraf S. Hasan Mahmoud

Rm 22-144

Ext. 1724

Email: ashraf@ccse.kfupm.edu.sa

Lecture Contents

1. Protocols
   a. Characteristics
   b. Functions
2. OSI
   a. The model
   b. OSI layers
3. TCP/IP Protocol Suite

Protocols - Characteristics

- **What is a Protocol:**
  - **Convention between two communicating entities governing exchange of data**
- **Elements of Protocol:**
  - **Syntax: data format, signal levels, etc.**
  - **Semantics: control info coordination and error handling**
  - **Timing: matching speeds and sequencing (synchronization)**

Protocols - Characteristics

- **Characteristics:**
  - **Direct / Indirect:**
    - **Direct: e.g for point-to-point communications, RS-232**
    - **Indirect: e.g devices connected through other nodes (internetwork, internet)**
  - **Monolithic / Structured**
    - **Monolithic: One package (SW and HW) performing all functions pertaining to the comm session**
    - **Structured: modular approach ← The focus of this course**

Protocols - Characteristics

- **Characteristics – cont'd:**
  - **Symmetric / Asymmetric**
    - **Symmetric: comm between peer entities**
    - **Asymmetric: to keep one side simple**
      - **E.g. client-server model, polling methods**
  - **Standard / Nonstandard**
    - **Standard: conforming to a single agreed upon standard**
    - **Nonstandard: no conformity**
    - **Clients vs. vendors ?**

Protocols - Functions

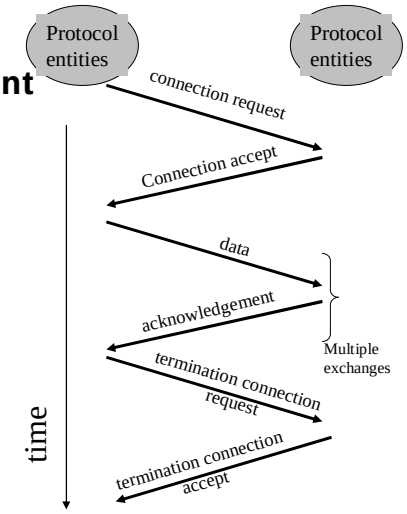
- **Encapsulation**
- **Segmentation and Assembly**
- **Connection Control**
- **Ordered Delivery**
- **Flow Control**
- **Error Control**
- **Addressing**
- **Multiplexing**
- **Transmission Services**

Protocols - Functions

- **Encapsulation:**
  - **PDU: block of data exchanged between two entities**
  - **PDU = user data + overhead (addressing, error control, protocol control)**
  - **User data is referred to as SDU**
- **Segmentation and Reassembly:**
  - **Example: ATM (53 bytes cells) core with Ethernet LAN (frames up to 1526 bytes)**

Protocols - Functions

- **Connection Control:**
  - **Connection establishment**
  - **Data transfer**
  - **Connection termination**



Protocols - Functions

- **Ordered Delivery**
  - **Ordered delivery of PDUs**
  - **Requires buffering, sequence numbers**
- **Flow control:**
  - **Limit amount of flow – e.g stop and wait procedure – receiving entity must acknowledge block before transmitter sends the next one in line**

Protocols - Functions

- **Error Control**
  - **To combat corruption of transmitted data**
  - **Transmitters inserts overhead info to detect corruption**
  - **Receiver checks overhead bits and finds outs if block is corrupted or not**
    - **Corrupted – may be correctable or request another copy**
    - **OK – accept block**

Protocols - Functions

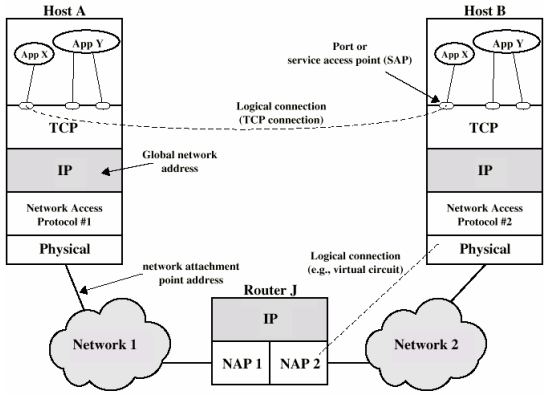
- **Addressing:**
  - **Level, Scope, Connection Identifiers, and Mode**

Level:
-Process or application address (e.g. SAP)
-Network level address (e.g. IP)

Scope:
-IP is globally unique and applicable
-Network attachment point address (e.g. MAC frames, X.25)

CI:
-Connection vs. connectionless
-Connection: use connection name (e.g. virtual circuit identifier)
-Connectionless: with each transmission the global address is used

Mode:
-Unicast
-Multicast
-Broadcast



Protocols - Functions

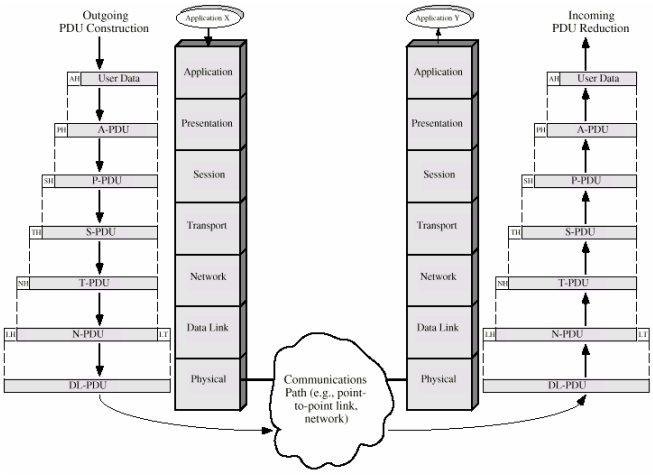
- **Multiplexing:**
  - **Multiple virtual circuits on one physical link (e.g X.25 – from one end system to another)**
  - **Mapping connections from one level (layer) to another:**
    - **E.g. Multiple service points carried on one virtual circuit (called upward or inward multiplexing)**
    - **Downward multiplexing: one high level connection is split or served by multiple lower level connections (for reliability and performance issues)**

The OSI Model

- **Software model and abstraction**
- **Defines set of layers and the services at these layers necessary to perform communication**
- **Promotes compatibility of network designs**
- **Logical partitioning:**
  - **Manageability and scalability**

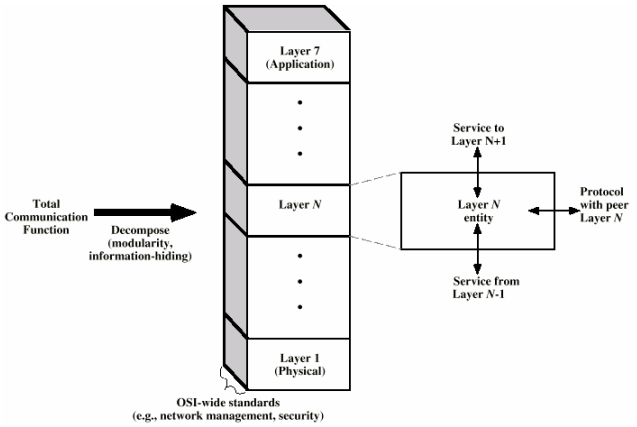
The OSI Model - Environment

- **Layer i establishes a PEER relationship with layer i on the target node**
- **This means Layer i requires service from layer i-1**
- **And so on**
- **The use of the PDUs**
- **No direct communication except for the physical layer – all other communication is indirect or virtual**
- **Encapsulation of user data**
- **Each layer may segment SDU to accommodate its own requirement – These are reassembled at the other end**



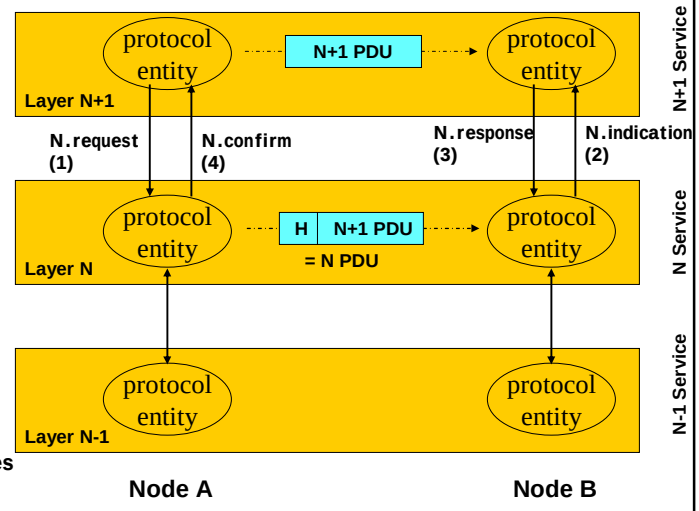
The OSI Model - Framework

- Very similar to subroutine or function design in software engineering
- Boundaries and functionalities are well designed – development of one layer has little or no effect on other layers
- Protocol specification:
  - Format of PDUs, and the semantic of each field
- Service definition:
  - What are the services provided to upper layer and the lower one
- Addressing:
  - E.g: NSAP is the address of an entity in the transport layer who uses the network service



The OSI Model - Framework

- Service Primitives:
  - Request:
  - Indication:
  - Respond
  - Confirm
- Note:
  - Encapsulation
  - Peer communication is virtual (dashed lines) – except at physical layer
  - Figure shows confirmed services case – For non confirmed services, the initiated receives no confirmation.



The OSI Model – Physical Layer

- **Specifications:**
  - **Mechanical: dimensions, connectors, etc.**
  - **Electrical: signal levels, rates of change, etc**
  - **Functional: functions performed by each circuit**
  - **Procedural: steps required to transport bits from one end to the other**
- **Provides service to do "transmission of raw bits"**

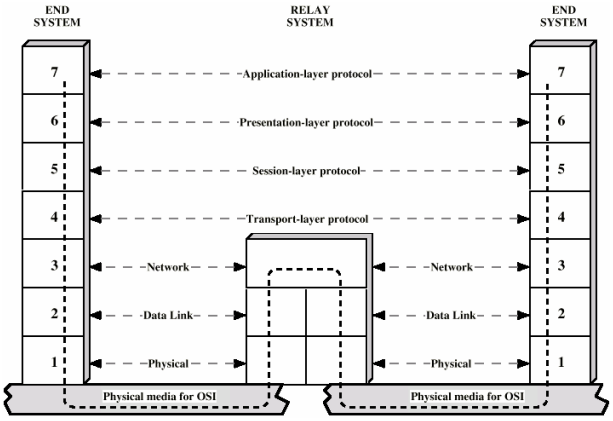
The OSI Model – Data Link Layer

- **Coverts the raw bit stream service provided by the physical layer to a reliable stream:**
  - **Performs error detection and error control**
- **Examples: HDLC, LAPB, LLC, etc**

The OSI Model – Network Layer

- **Service: transfer of information between two end systems across communication network – End to end delivery of packets**

- **Two end systems may be connected by:**
  - **Point-2-point: no need for network layer**
  - **Same network (see figure)**
  - **Different network**



9/30/2002

The OSI Model – Transport Layer

- **Service: mechanism of exchanging data (or messages) between the two end systems:**
  - **For connection oriented networks:**
    - **Error-free delivery**
    - **Ordered delivery**
    - **No loss or duplication**
    - **Attempts to provide a certain quality of service (QoS) {certain max error rate, delay jitter, etc) through optimizing the the network layer services**
- **Example: TCP (connection oriented), UDP (connectionless)**

The OSI Model – Session Layer

- **Service: mechanism of controlling the dialogue between applications at end systems**
  - **Dialogue Discipline**
  - **Grouping**
  - **Recovery**

The OSI Model – Presentation

- **Service: defines format of data (format, encryption, and compression) to be exchanged between applications**

The OSI Model – Application

- **Service: A means for user applications (email, ftp, etc) to access the services provided by the OSI model**

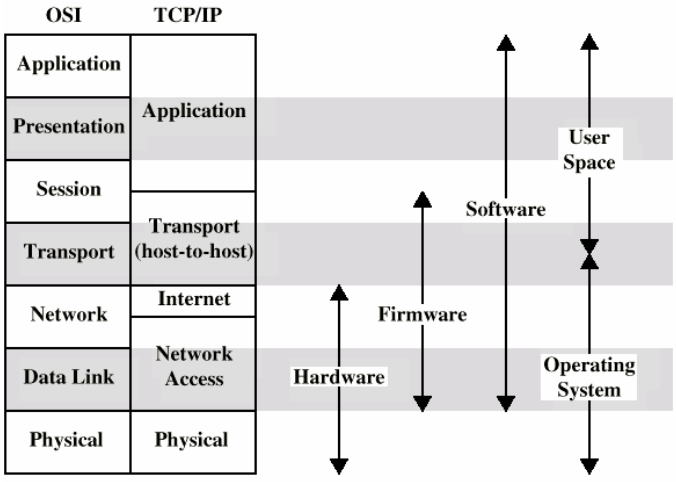
The TCP/IP Model

- **TCP/IP is the result of R&D conducted on experimental packet switched network (ARPANET) and funded by Defense Advanced Research Agency (DARPA)**
- **TCP/IP is NOW the dominant commercial architecture – The foundation of the internet and its applications**

The TCP/IP Model

- **Model has five independent layers:**
  - **Application layer: comm between processes or applications on separate hosts**
  - **Transport layer: end-2-end transfer service – may include reliability mechanisms**
  - **Internet layer: routing data from source to destination through one or more networks**
  - **Network access layer: logical interface between end systems and the network**
  - **Physical layer: defines mechanism of transmitting raw bits depending on media characteristic**

The TCP/IP Model (using the OSI Model as a reference)

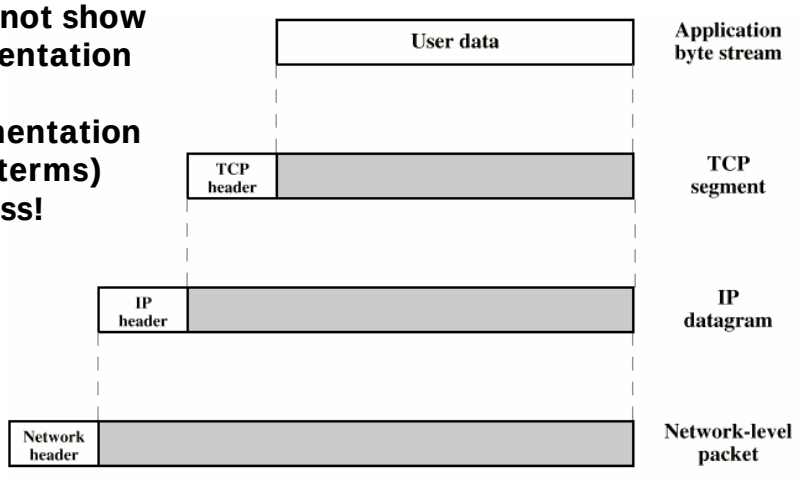


Example of TCP/IP Communications

- A process (has port 1) on host A needs to communicate to another process: port 2 at host B
- The application layer on A hands the msg down to TCP with instructions to deliver it to (port2,host B)
- TCP hands msg down to IP with instructions to send it to host B:
  - The IP layer knows how to reach host B (or at least the first hop of the route) – does not care about port info
- IP hands down packets to network access (say Ethernet) with instructions to pass it to next router (first hop on the way to B)

Example of TCP/IP Communications

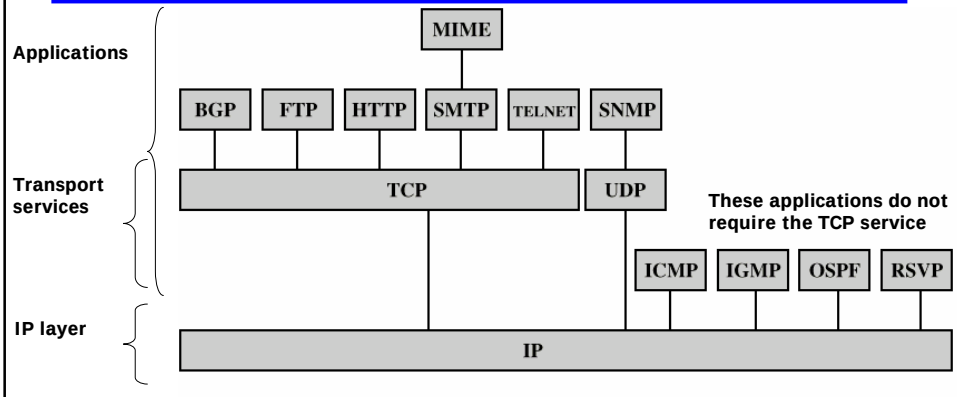
- Does not show segmentation (or fragmentation in IP terms) process!



TCP/IP Control Information (Partial)

- **TCP control info:**
  - **Destination port number**
  - **Sequence number**
  - **Checksum**
- **IP control info:**
  - **IP address**
- **Network Access control info:**
  - **Destination network access address (this is not the IP!!)**
  - **Facilities request (e.g. priorities)**

TCP/IP Control Information (Partial)



BGP = Border Gateway Protocol
FTP = File Transfer Protocol
HTTP = Hypertext Transfer Protocol
ICMP = Internet Control Message Protocol
IGMP = Internet Group Management Protocol
IP = Internet Protocol
MIME = Multi-Purpose Internet Mail Extension
OSPF = Open Shortest Path First
RSVP = Resource ReSerVation Protocol
SMTP = Simple Mail Transfer Protocol
SNMP = Simple Network Management Protocol
TCP = Transmission Control Protocol
UDP = User Datagram Protocol