King Fahd University of Petroleum & Minerals
Computer Engineering Dept

COE 241 – Data and Computer Communications

Term 152

Dr. Ashraf S. Hasan Mahmoud

Rm 22-420

Ext. 1724

Email: ashraf@kfupm.edu.sa

Lecture Contents

1. The need for a protocol
2. Simple protocol architecture
3. TCP/IP Protocol Suite
4. Standardization within a protocol architecture
5. Traditional Internet-based applications
6. Multimedia (optional)

The Need For A Protocol Architecture

- Example: transfer of file between two computers
  - Connected point-to-point, or through network
- Several tasks are required:
  1. The source must either activate the direct communications path or inform the network of the identity of the desired destination system
  2. The source system must ascertain that the destination system is prepared to receive data
  3. The file transfer application on the source system must ascertain that the file management program on the destination system is prepared to accept and store the file for this particular user
  4. A format translation function may need to be performed by one or the other system if the file formats used on the two systems are different
  5. Etc.

Layered Protocol Architecture

- Tasks and subtasks needed to achieve the communication are implemented using a layered architecture
  - Modules arranged in a vertical stack
  - Each layer in the stack performs *related* functions
  - Each layer uses the services of the layer below and provides services to the layer on top
  - Peer layers (entities)
- Protocol – a set of rules that allow communication between peer entities
- Elements of Protocol:
  - Syntax (format of data blocks),
  - Sematics (includes control info for coordination and error handling), and
  - Timing (speed matching and sequencing)

Example: A Simple Protocol Architecture - Distributed data communication

- Involves:
  - Agents: (1) applications, (2) computers, and (3) network
  - Communication tasks that may be divided into relatively three independent layers:
    - Application layer - Contains logic to support applications
    - Transport layer - Collects mechanisms in a common layer shared by all applications
    - Network layer - Concerned with the exchange of data between a computer and the network to which it is attached
- Concept of Service Access Points (SAPs) or ports

Example: A Simple Protocol Architecture - Distributed data communication – cont'd

- Modules at the same levels are peer layers (or entities)
- At the application and transport levels – indirect communication
- The network access layer communicates with the communication network

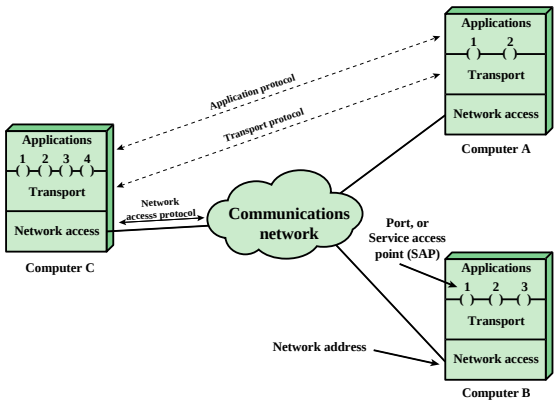


Figure 2.1 Protocol Architectures and Networks

Example: A Simple Protocol Architecture - Distributed data communication – cont'd

- The application layer uses the SAPs or Ports to access the services of the transport layer
- Peer layers exchange Protocol Data Units (PDUs) → added header + data from higher layer
- Encapsulation

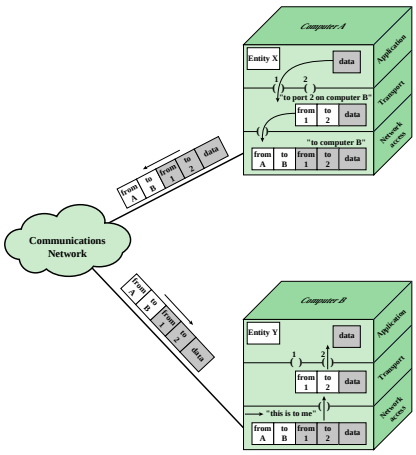


Figure 2.2 Protocols in a Simplified Architecture

Studied in details in COE 344

The TCP/IP Protocol Architecture

- **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**

Studied in details in COE 344

The TCP/IP Protocol Architecture – cont'd

- 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 o 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

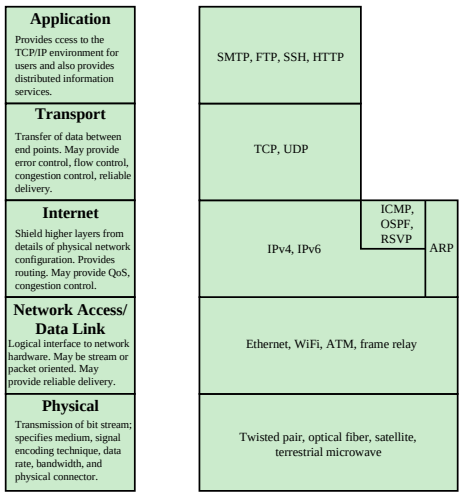


Figure 2.3 The TCP/IP Layers and Example Protocols

Studied in details in COE 344

Operation of TCP and IP

- App Y (has port 1) on host A needs to communicate to App X listening at 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)

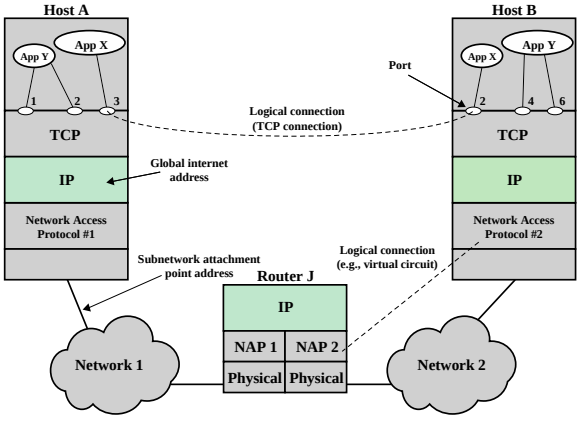
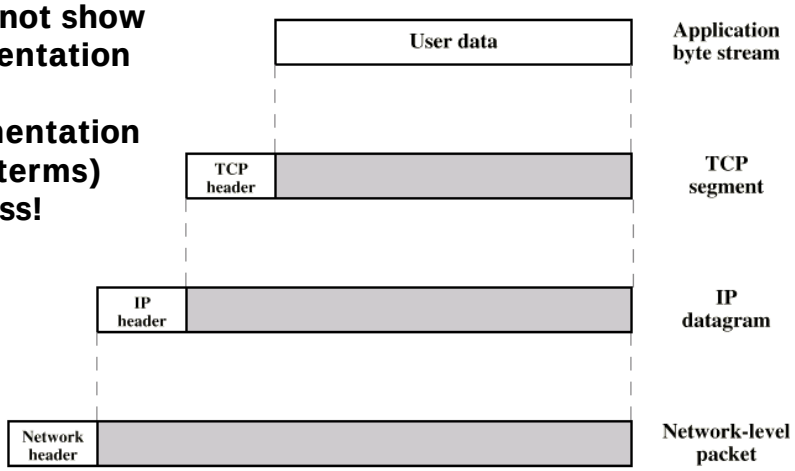


Figure 2.4 TCP/IP Concepts

Studied in details in COE 344

Operation of TCP and IP – cont'd

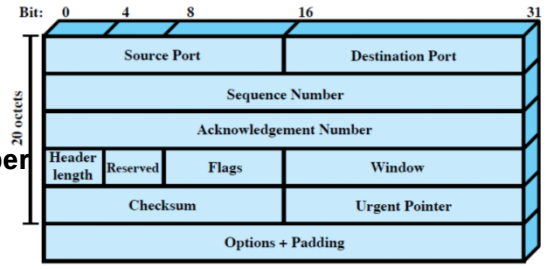
- **Does not show segmentation (or fragmentation in IP terms) process!**



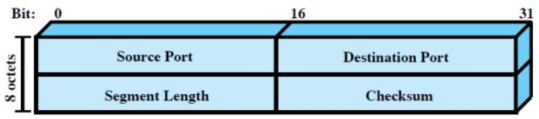
Studied in details in COE 344

TCP Headers

- **TCP control info:**
  - **Destination port number**
  - **Sequence number**
  - **Checksum**



(a) TCP Header



(b) UDP Header

Figure 2.3 TCP and UDP Headers

Studied in details in COE 344

IP Header

- **IP control Info**
  - **Version**
  - **Source Address**
  - **Destination Address**
  - **Protocol**
- **Note the 32-bit address for IPv4 versus the 128-bit address for IPv6.**

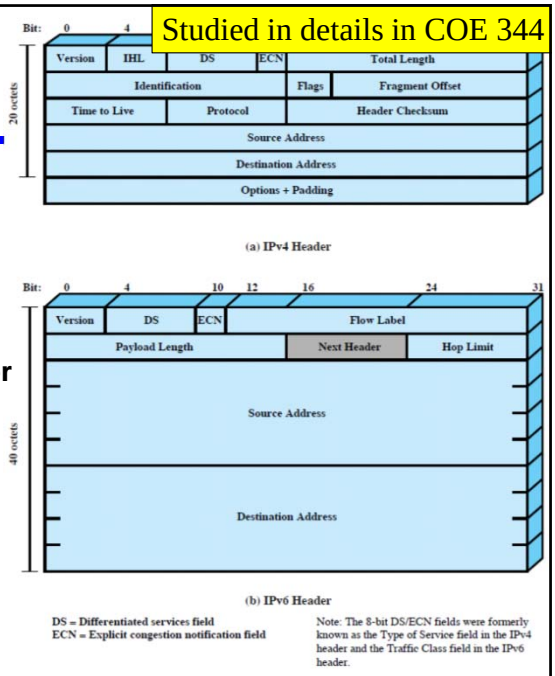
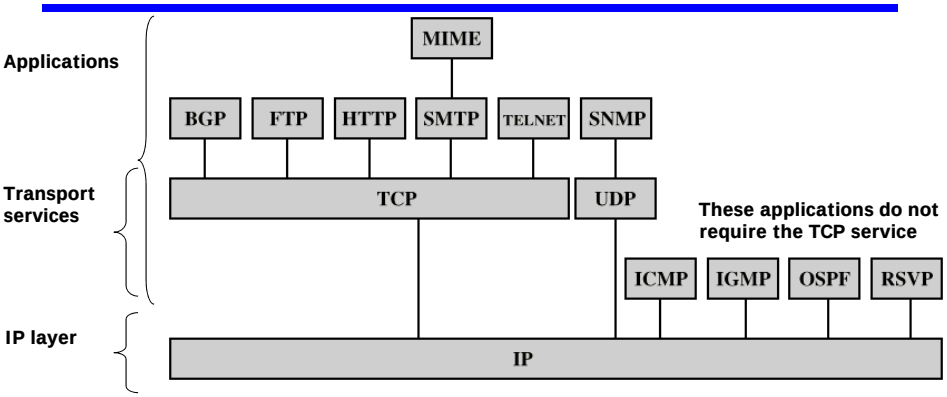


Figure 2.4 IP Headers

Studied in details in COE 344

TCP/IP Protocols



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

Standardization Within A Protocol Architecture

- Very similar to subroutine or function design in software engineering
- Layering:
  - Functions in each layer are WELL DEFNED → standards for the layers may be developed independently and simultaneously
  - Boundaries between layers are WELL DEFINED → 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

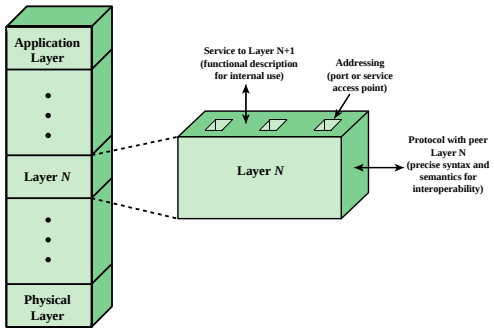
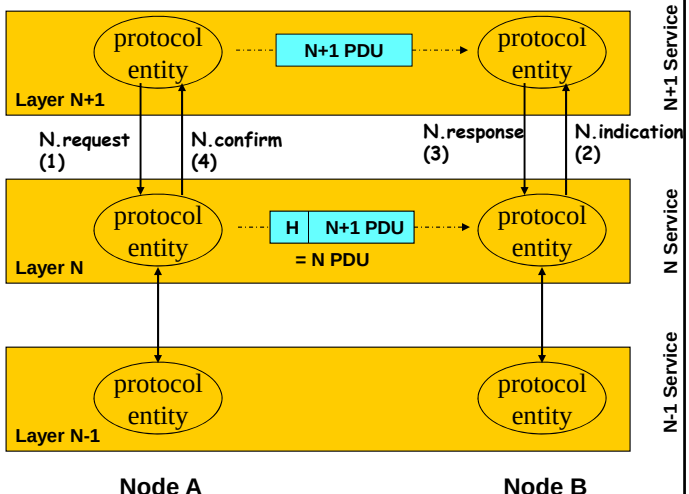


Figure 2.9 A Protocol Architecture as a Framework for Standardization

Service Primitive and Parameters

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



Studied in details in COE 344

Traditional Internet Based Applications

- Simple Mail Transfer Protocol (SMTP)
- File Transfer Protocol (FTP)
- Telnet

Studied in details in COE 344

Multimedia (Optional)

- **Media: text, still images, and video**
- **Multimedia: Human-computer interaction involving text, graphics, images, and/or audio/video**
- **Streaming Media: video and audio clips**
- **Quality of Service (QoS) Parameters include:**
  - **Throughput, delay, delay variation (jitter), packet loss, etc.**
- **Types of Traffic:**
  - **Elastic: can accept variable range of QoS levels across the internet – e.g. TCP/IP is designed for this**
  - **Inelastic – have very strict QoS levels requirements – e.g. (real-time) voice traffic**