Protocol Architecture and Layering

Architecture of Modern Day Networks

- Current day networks have
  - Huge number of services running on them
  - Myriad of protocols, each doing something
    - Ethernet, TCP, UDP, IP, ARP, RARP, ICMP, DNS, DHCP, HTTP, SMTP, FTP, Telnet, SCTP, RTP, …..too many to specify
  - Myriad of devices (not counting end devices)
    - Access Points, L2 Switches, L3 switches, routers, load balancers, firewalls, …
    - Some of them are hardware-based (with firmware/software loaded) , some of them are software-based, some can be purchased in both forms

- Big questions
  - Can we organize all these to understand them better?
  - If I want to write a new service using the network, do I have to implement it from scratch?
    - Right down to sending the bits out???
  - If I come up with a better implementation of some existing protocol/service, does everything using it or whatever the old version was using all have to change? How do I integrate it with everything that exists?
  - While I may have administrative control over some parts (for example, CIC head has control over IIT network), I do not ask others to change just because I changed something!

- No adhoc approach will work
- Two important keywords towards a solution
  - Protocols
    - Agree on some common standards that everyone must follow for achieving specific tasks
    - Who makes everyone agree? How do you get the community to accept a new protocol?
      - Lots of organizations and well-defined processes involved
      - IETF and RFCs for Internet
  - Layering
    - Organize the protocols into layers

Protocol Architecture

- Task of communication broken up into modules or layers
- Each layer expects some service from the layer below it, and provides some service to its higher layer
- Topmost layer is application
  - Protocols implementing different services that all need some common functionalities from the "network"
- Two approaches
  - OSI Layers (Open Systems Interconnect) standardized by ISO and CCITT
  - TCP/IP Layers (de-facto standard for modern day internet)

OSI Layers

- Open Systems Interconnection
- Developed by the International Organization for Standardization (ISO)
- Seven Layers
  - Application
  - Presentation
  - Session
  - Transport
  - Network
  - Data Link
  - Physical

TCP/IP Layers

- De-facto standard
- Five Layers
  - Application
  - Transport
  - Network
  - Data Link
  - Physical

Physical Layer

- Physical interface between data transmission device (e.g. computer) and transmission medium or network
- Specifies raw transmission details like connectors, medium type, voltage levels, encodings used etc.
- Some of these we studied already
- Deals with transmission of raw bit stream, no meaning associated with the stream
- Mostly implemented in hardware
- We will not do the electrical details much, though we will look at the connector/media etc. from a practical point of view

Data Link Layer

- Ensures reliable communication between directly connected nodes (a link)
- Higher layers can think that a reliable link exists between two machines, and not worry about noise, attenuation, error etc.
- Deals with framing, flow control, error control etc.
- Mostly a combination of hardware and software
- Things we already studied so far
- Ethernet is one of the most well-known protocols in this layer
    - It also includes physical layer specifications, we will see later

Network Layer

- Primary function is routing – sending packets from a source machine to a destination machine when they are not directly connected
  - How is a path found from the sender to the receiver for a packet?
  - Efficient management of these paths
  - Actually, a machine here means a network card more specifically
- Packets may not reach in order, get lost etc.
- May have some other functionalities like fragmentation etc.
- Implemented usually in software
- IP is the most well-known protocol in this layer
- Issues we have not studied so far

Transport Layer

- End-to-end delivery between any two applications (not just machines)
  - Just think you have two web browsers open to cse.iitkgp.ac.in an browsing different links
    - Source and destination are the same
    - Network layer only gets the packets to your PC/laptop
    - How does the right things go to the right browser?
  - Basically multiplexing/demultiplexing different applications's data over the same network connection
- Can give additional guarantees like reliability, in-order delivery etc.
- Implemented usually in software
- TCP and UDP are the most well-known protocols in this layer

Application Layer

- Application specific services like email, ftp, telnet, http etc.
  - Very large number of protocols in this layer
  - Implemented usually in software
- We do not look at session and presentation layers of OSI as TCP/IP layers has become the de-facto standard
  - OSI presentation layer provides services like compression/decompression, encryption/decryption, encoding/decoding
  - OSI Session layer provides services for session creation and maintenance
  - TCP/IP subsumes some of the functionality into transport layer and leaves all others for application layer

Protocol Data Unit (PDU)

- At each layer, protocols are used to communicate to the corresponding layer at the other end
  - Peer-to-peer communication, assuming a direct logical connection between the two layers
- Data from the higher layer may be broken up into smaller data units
- Control information is added to each data unit at each layer
- PDU at a layer – control info added by the layer + data unit
- Called differently at different layers – frame in datalink layer, packet in network layer, segment in transport layer, message in application layer

Encapsulation

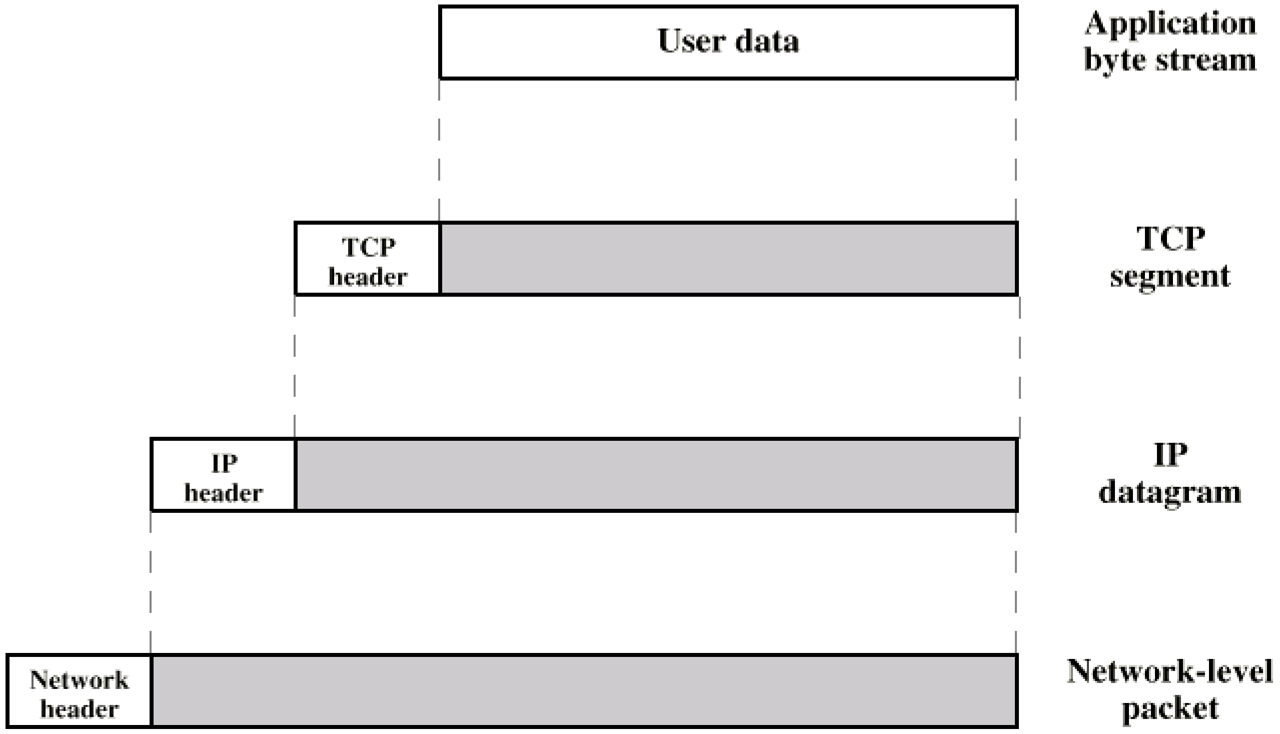
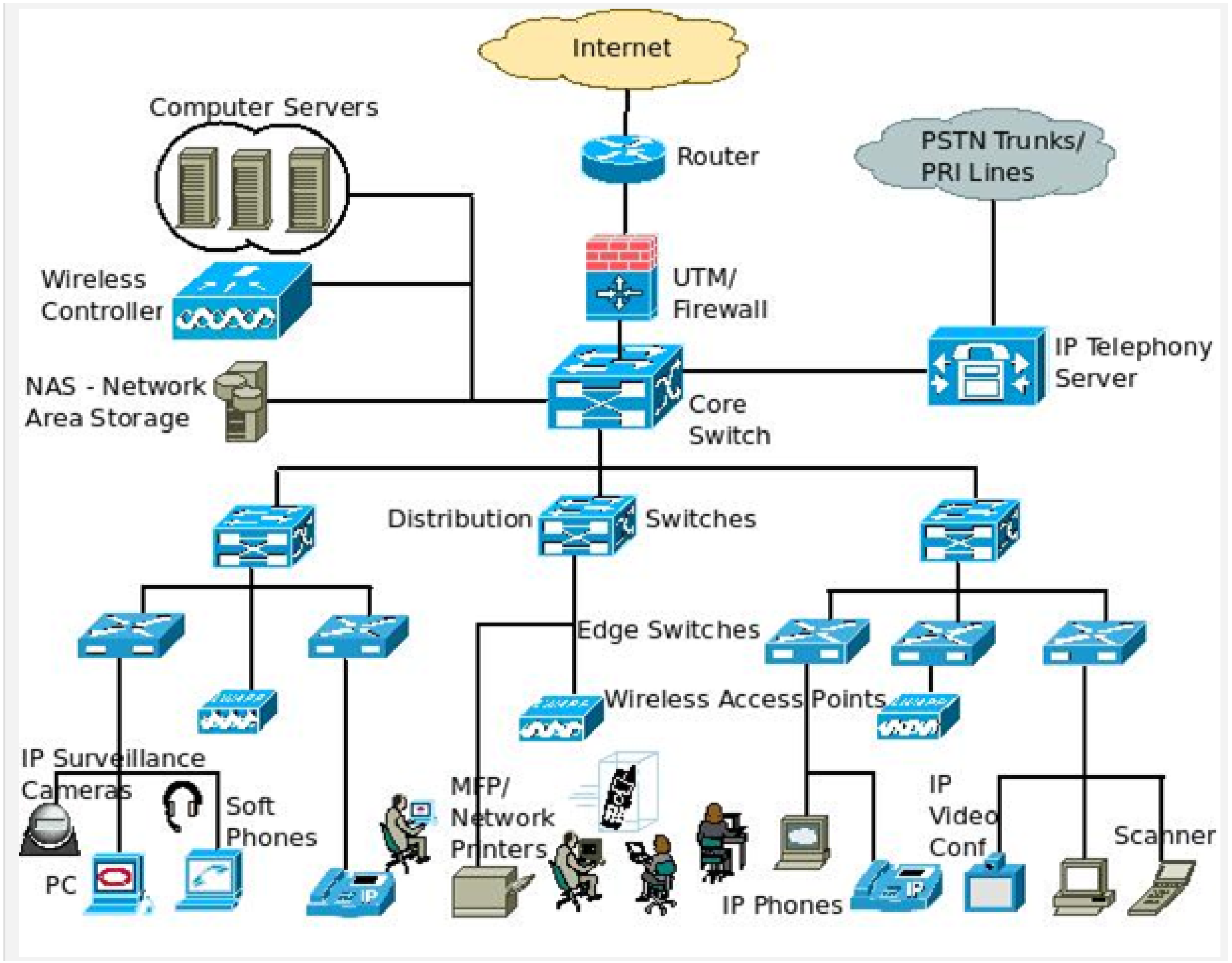


Figure 2.11 Protocol Data Units in the TCP/IP Architecture

Number of Layers?

- Each layer takes data from higher layer, possibly breaks it up into smaller chunks, and adds its own header to each chunk
- More the no. of layers, more headers are added as the data goes downwards, more wastage
  - For a total of x no. of bits actually transmitted, a significant part is in headers and so is overhead
- Too few layers – defeats the purpose of layering (isolating functionalities in layers) itself
- But why are the headers needed really?
  - We will see later when we study the protocols

Physical Architecture



- We will demystify this as we go on
- But what is there in that yellow cloud?