UNIT 1

- INTERNET
  - Nuts & Bolts View
    - Network of Networks (Interconnected ISPs).
    - Millions of connected computing devices or hosts (end systems), running network applications.
    - **Commuication links**: Fiber cable, coaxial cable, radio signals, satelite transmission
    - Different transmission medias have different transmission rate measured in bits/sec.
    - **Packet**: When one end system sends data to another end systems it has to segment the data into smaller pieces and it has to add header information to these segments, The resulting packages of information known as packets are sent through the network to destination system.
    - **Packet Switches**: Takes packet arriving on one of it's incoming communication links and forwards that packet in one of it's outgoing communication links.
    - **Route (Path)** : The sequence of communication links and packet switches traversed by a packet is known as route or path throgh the network.
    - **Protocols:** protocols define format, order msg sent & received and actions taken on msg transmission.
      - 2 most famous internet protocols are TCP, IP.
      - The IP protocol specifies the format of the packets that are sent and received among routers and end systems. The Internet's principal protocols are collectively known as TCP/IP.
      - Given the importance of protocols to the Internet, it's important that everyone agree on what each and every protocol does, so that

people can create systems and products that interoperate. This is where standards come into play.

- **Internet standards**: Developed by IETF, IETF documents are called RFCs
    - RFC ⇛ Request for comments. (detailed & technical documents, more than 6000 RFCs exist.)
    - IETF ⇛ Internet Engineering Task Force.

▼ Service View

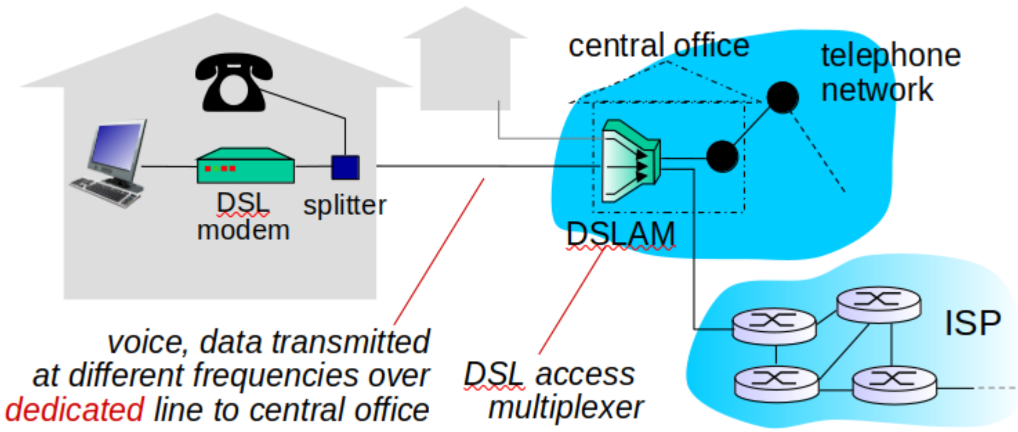
- Infrastructure that provides services to applications.
- Provides Application programming interface (API) to apps, that will allow distributed apps to communicate with each other using internet
- API : Specifies how a program running on one end system asks the internet infrastructure to deliver data to a specific destination program running on another end systems.

▼ Network Structure

- Network Edge:
    - Hosts: Client PCs or Servers in Data centers.
- Access Networks:
    - physical wires like coaxial cable, that connect end systems to the network
    - Wireless communication links.
- Network Core
    - Mesh of interconnected routers that interconnects the Internet's end systems
    - So, if a source end system or a packet switch is sending a packet of L bits over a link with transmission rate R bits/sec, then the time to transmit the packet is L/R seconds.

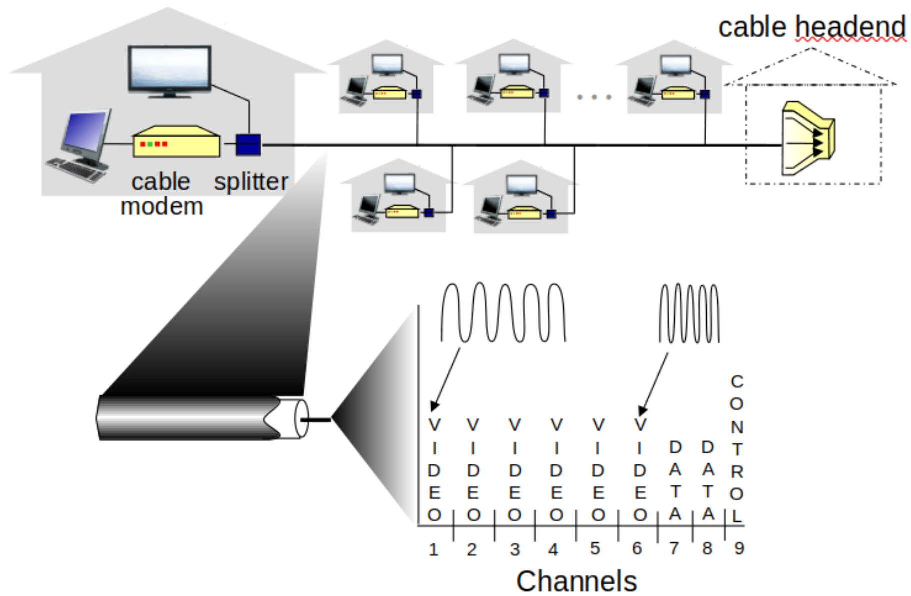
▼ Access Networks

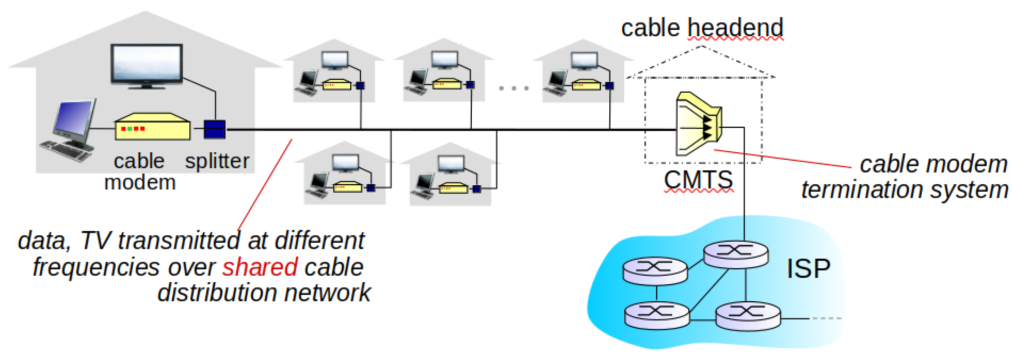
▼ DSL (Digital Subscriber Line)



- Use Existing Telephone line to central office DSLAM
    - Data over Telephone line goes to the internet
    - Voice over Telephone line goes to the telephone network.
- **Dedicated line to central office**
- **Transmission rates:**
    - **Upload speed ⇒ upto 2.5 mbps**
    - **Download speed ⇒ upto 24 mbps**

▼ Cable networks





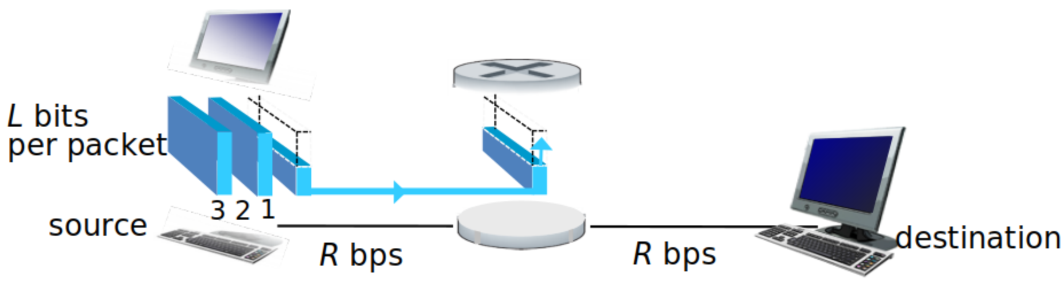
- Frequency division multiplexing:
  - Different channels transmitted over different frequency bands.
- **homes share access network to cable headend**
- **uses HFC (Hybrid Fiber Coaxial) cable.**
- **Transmission Rates:**
  - Upload Speed ⇒ upto 2 mbps
  - Download Speed ⇒ upto 30 mbps

▼ Packet Switching Networks

- Networks in which packets are traversed using packet switches like router or link layer switches are known as packet switching networks.

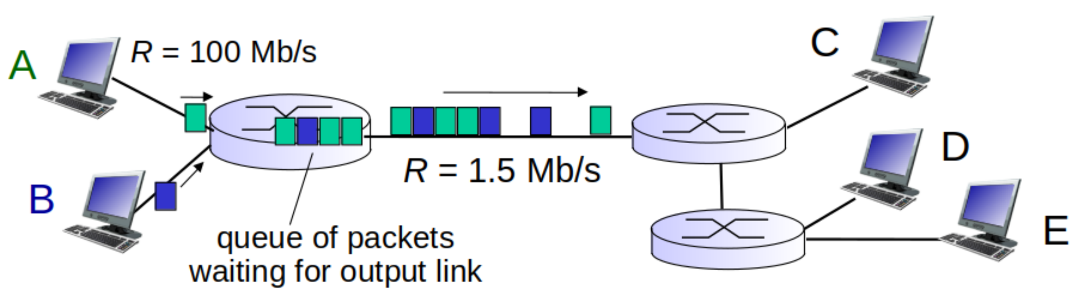
▼ Store & Forward

- Packet switched networks uses Store & Forward technique.



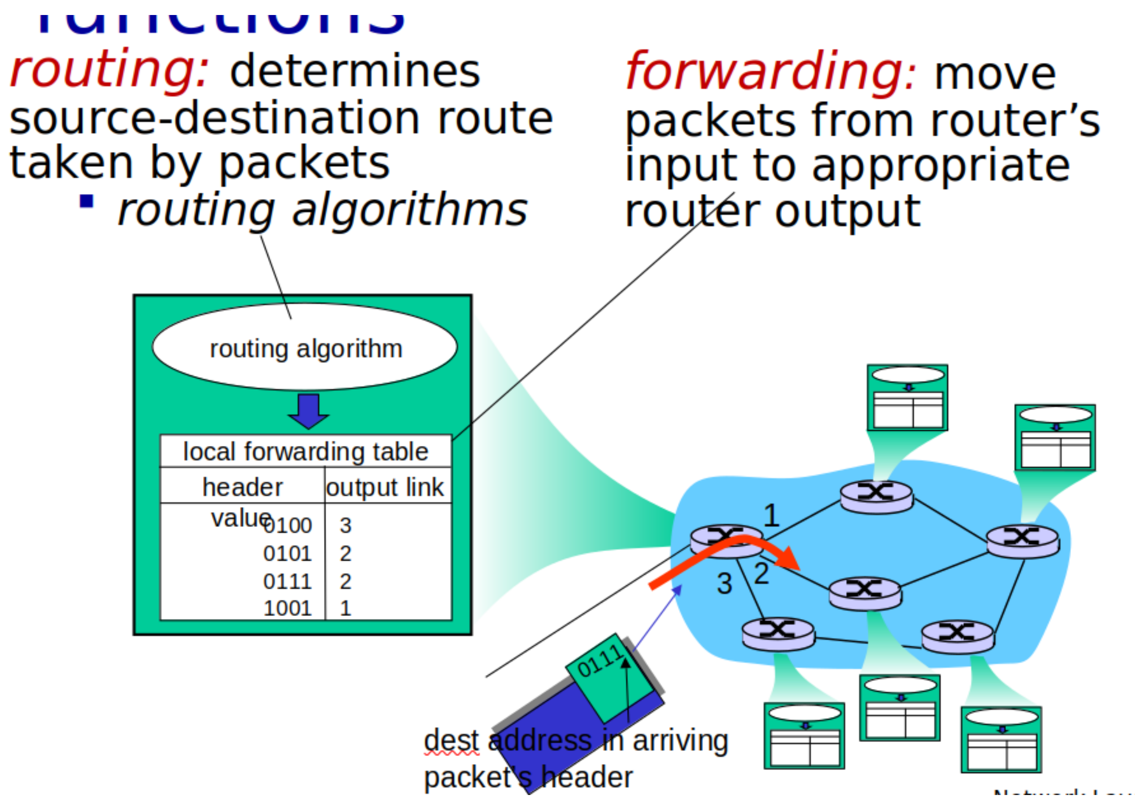
  - entire packet must arrive at router before it can be transmitted on next link
  - takes L/R seconds to transmit (push out) L-bit packet into link at R bps

▼ Queuing delay, loss



- If arrival rate (in bits) to link exceeds transmission rate of link for a period of time:
  packets will queue, wait to be transmitted on link packets can be dropped (lost) if memory (buffer) fills up

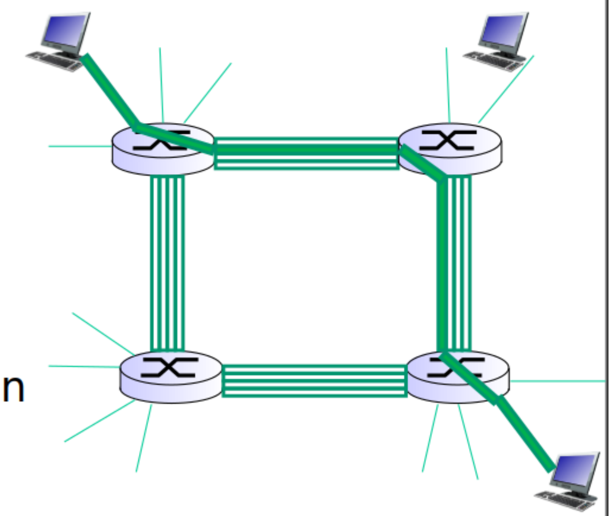
▼ Routing & Forwarding



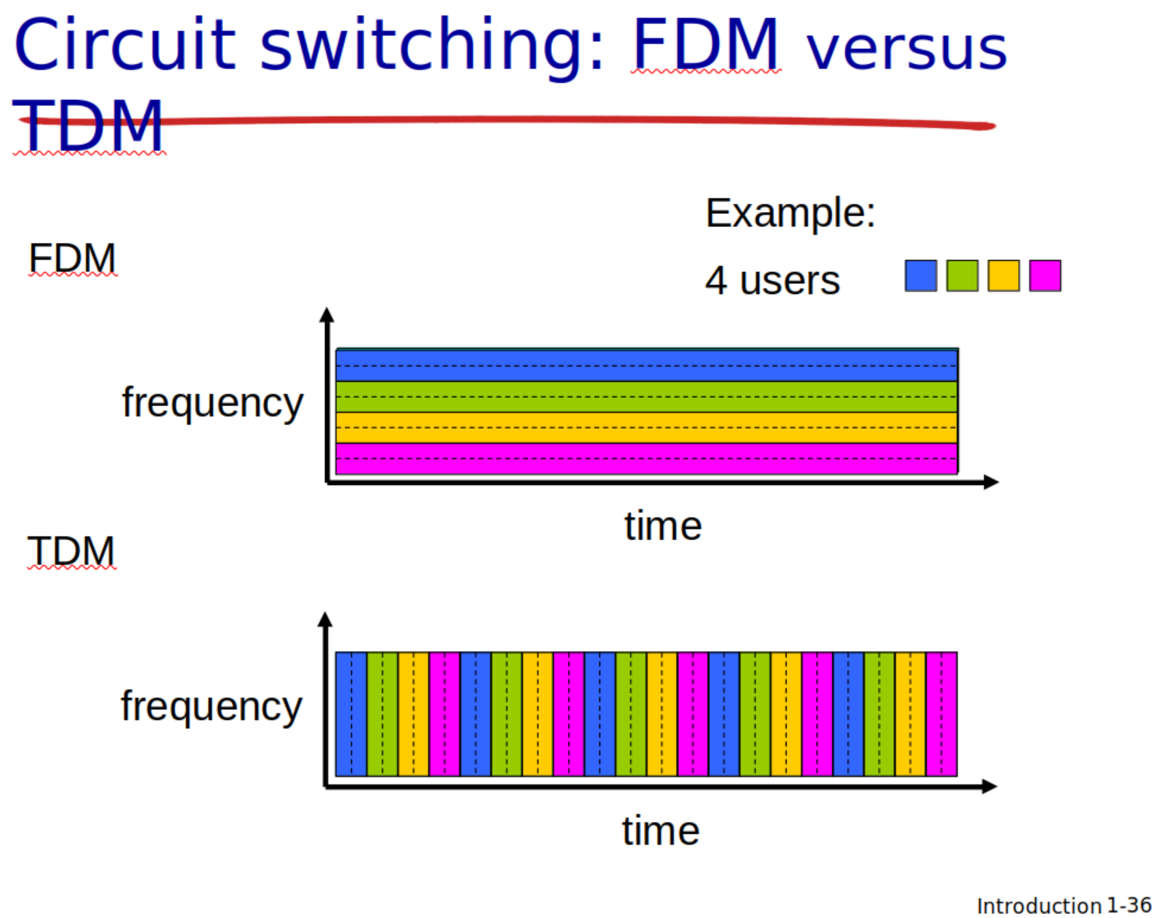
▼ Circuit switching networks

Alternative core: circuit switching

end-end resources allocated to, reserved for “call” between source & dest:



- Restaurant Analogy
- In diagram, each link has four circuits.
  - call gets 2nd circuit in top link and 1st circuit in right link.
- dedicated resources: no sharing
  - circuit-like (guaranteed) performance
- circuit segment idle if not used by call *(no sharing)*



▼ packet switching vs circuit switching

Packet switching versus circuit switching

- Critics of packet switching have often argued that packet switching is not suitable for real-time services (for example, telephone calls and video conference calls) because of its variable and unpredictable end-to-end delays (due primarily to variable and unpredictable queuing delays).

- Proponents of packet switching argue that
  (1) it offers better sharing of transmission capacity than circuit switching and
  (2) it is simpler, more efficient, and less costly to implement than circuit switching.

Introduction 1-37

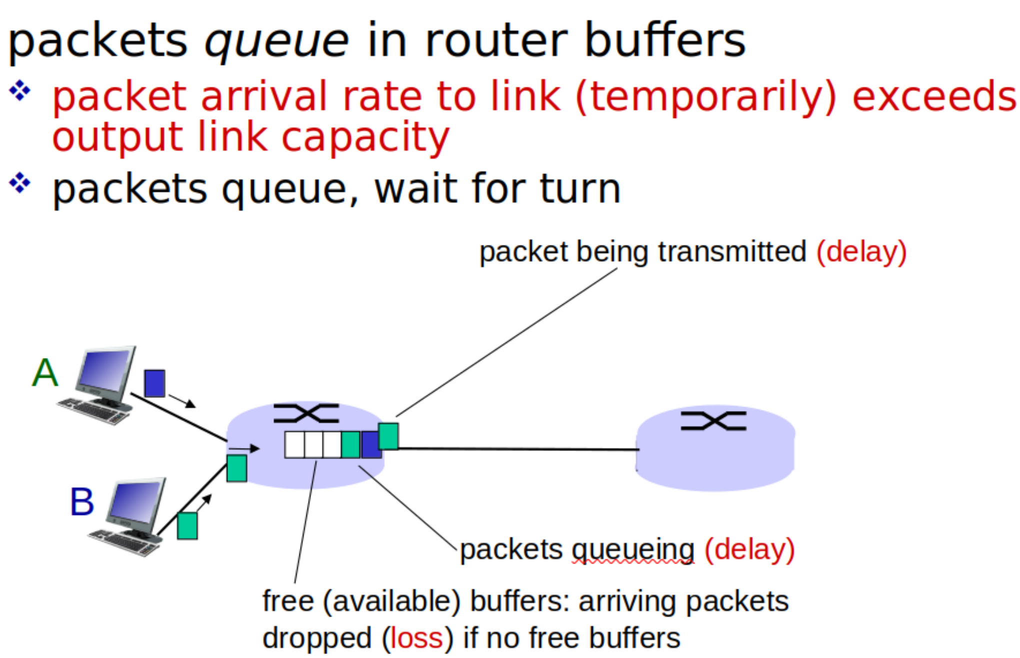
▼ Internet Structure

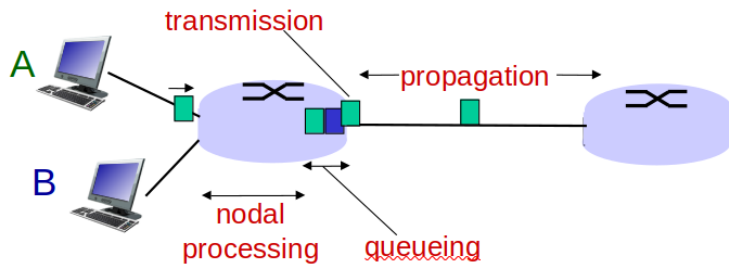
- End Systems connect to internet via Access ISPs.
- Access ISPs in turn must be interconnected.
  - So that any two hosts connected to 2 different ISPs can send packets to each other.
- Brainstorming the ways to create internet structure:
  1. just connect all Access Networks together O(n^2) connections (doesn't scale).
  2. (The right way) ⇒
     - Connect each Access ISP to global transit ISP

- because ISP is a viable business, there will be more than one ISP, so Connect these ISPs via “Internet Exchange points” and “Peering Links”.
- Regional (Local) access networks may also arise, so connect them with ISPs too.
- And there might be a private organization like Google, Which have their own content provider network to serve it’s clients directly or for better efficiency.

▼ Delay

- Ideally, we would like Internet services to be able to move as much data as we want between any two end systems, instantaneously, without any loss of data.
- This is a lofty goal, one that is unachievable in reality. Instead, computer networks necessarily constrain throughput (the amount of data per second that can be transferred) between end systems, introduce delays between end systems, and can actually lose packets.





$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

d_{proc}: nodal processing

- check bit errors
- determine output link
- typically < msec

d_{queue}: queueing delay

- time waiting at output link for transmission
- depends on congestion level of

Introduction 1-54

d_{trans}: transmission delay:

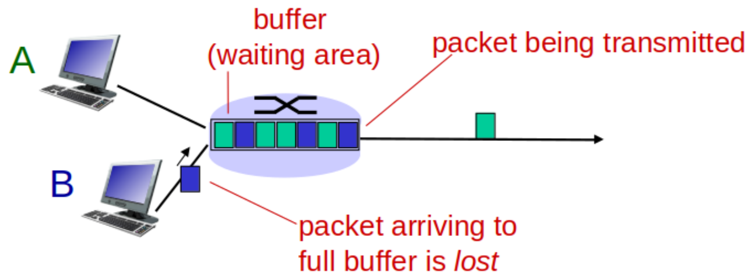
- L: packet length (bits)
- R: link *bandwidth (bps)*
- $d_{trans} = L/R$

d_{prop}: propagation delay:

- d: length of physical link
- s: propagation speed in medium (~2×10^8 m/sec)
- $d_{prop} = d/s$

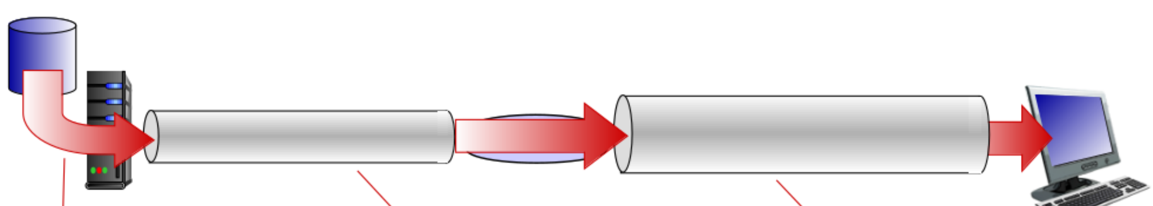
▼ Loss

- queue (aka buffer) preceding link in buffer has finite capacity
- packet arriving to full queue dropped (aka lost)
- lost packet may be retransmitted by previous node, by source end system, or not at all



▼ Throughput

- *throughput:* rate (bits/time unit) at which bits transferred between sender/receiver
  - *instantaneous:* rate at given point in time
  - *average:* rate over longer period of time



- Bottleneck : Link on end-end path that constrains end-end throughput

▼ Protocol Layering

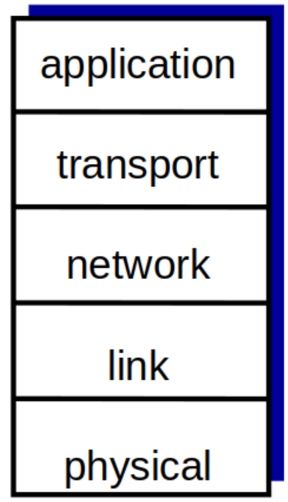
- modularization eases maintenance, updating of system change of implementation of layer’s service transparent to rest of system.

- Each protocol belongs to one of the layers.
- Each layer provides its service by
    1. performing certain actions within that layer
    2. by using the services of the layer directly below it.
- A protocol layer can be implemented in software, in hardware, or in a combination of the two.
- Application-layer protocols—such as HTTP and SMTP—are almost always implemented in software in the end systems; so are transport-layer protocols.
- Because the physical layer and data link layers are responsible for handling communication over a specific link, they are typically implemented in a network interface card (for example, Ethernet or WiFi interface cards) associated with a given link.
- The network layer is often a mixed implementation of hardware and software.
- When taken together, the protocols of the various layers are called the protocol stack.

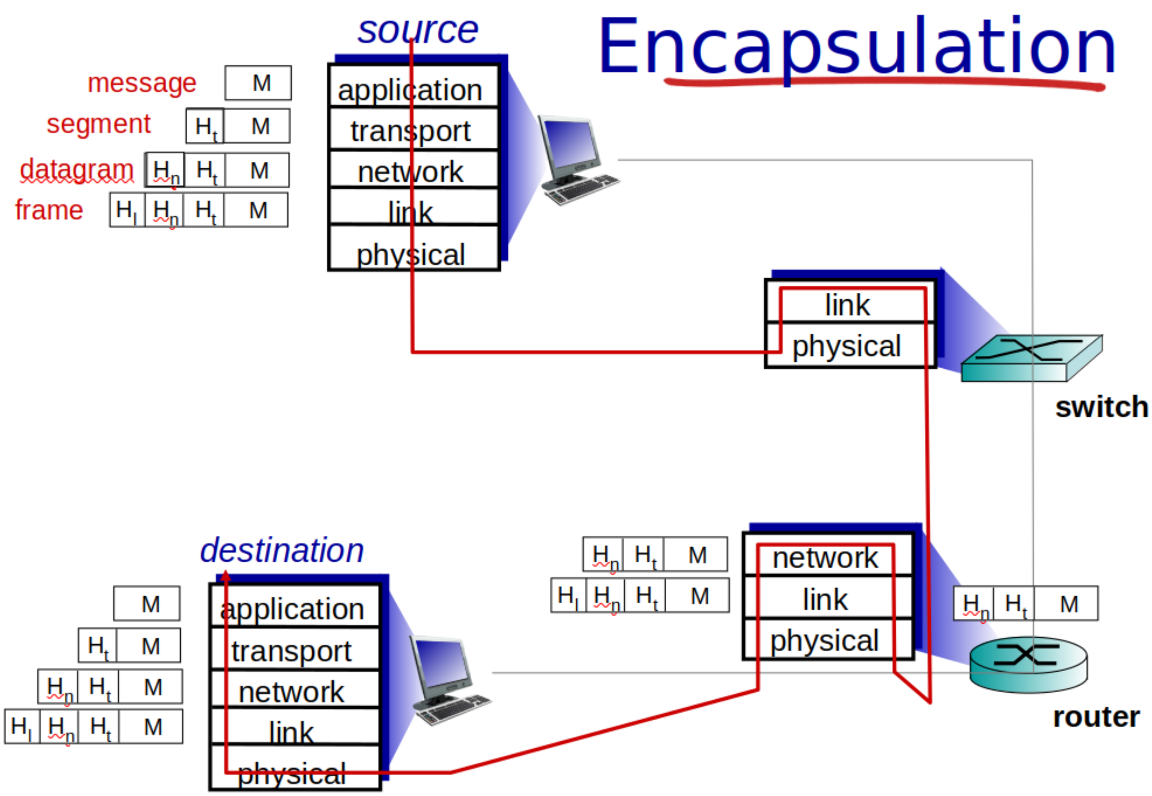
▼ **Five layers of Protocol stack**

1. Application Layer
2. Transport Layer
3. Network (IP) Layer
4. Link Layer
5. Physical Layer

- *application:* supporting network applications
  - FTP, SMTP, HTTP
- *transport:* process-process data transfer
  - TCP, UDP
- *network:* routing of datagrams from source to destination
  - IP, routing protocols
- *link:* data transfer between neighboring network elements
  - Ethernet, 802.111 (WiFi), PPP
- *physical:* bits “on the wire”



▼ Encapsulation



Not mentioned topics:

Physical media.

TCP's Each layer functionalities.