Introduction

- Basics
  - Introduction to networks
  - Connectivity, reliability
- Network architecture
  - Layering and protocols, encapsulation, Internet architecture
- Performance metrics
  - Bandwidth, latency, throughput

Elements of a network

- Nodes
  - Devices that need to be connected
  - Can be end hosts, servers, routers, switches…
- Links
  - A physical medium: Coaxial cable, optical fiber, etc.
  - Anything else?
  - Vacuum?
-

Nodes in a network

- General purpose computer
  - Most often a node is a desktop PC
  - PC is connected to network via network adapter
- Router
  - Specialized hardware for forwarding data
- Switch
  - Interconnecting networking devices
- Hub, repeater, bridges, etc…

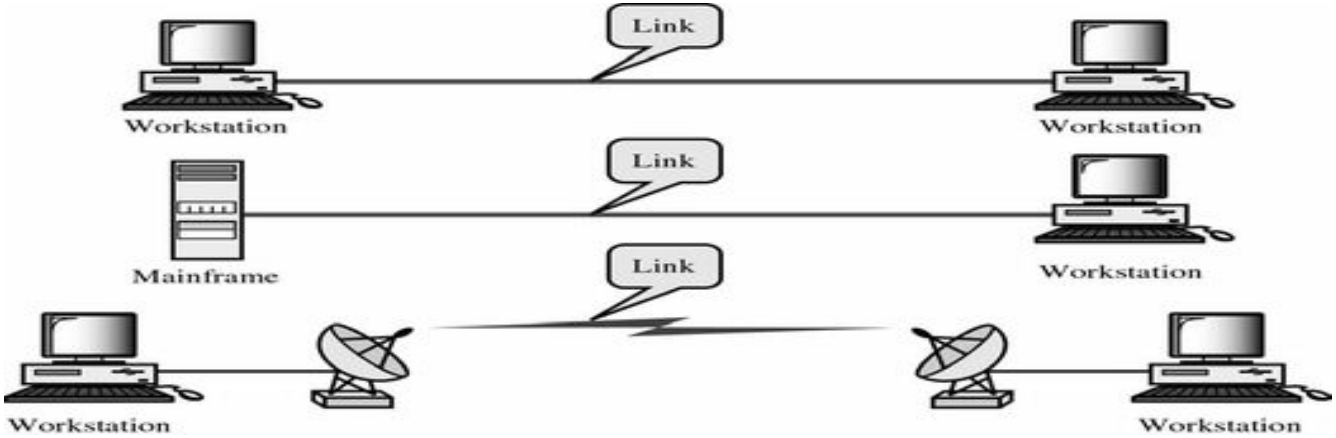
Links in a network

- Wired
  - Coaxial Cables, Optical Fiber…
- Wireless
  - Satellite, cellular..
- Speed of signal
  - In medium
    - (speed of light in medium) / frequency
  - In vacuum

Is that enough?

- Are we missing something?
  - Data needs to be transmitted to a medium before it can be received at the other end
  - Hence, it needs to be ENCODED in a signal !
- Encoding schemes
  - NRZ, NRZI
  - Manchester
  -

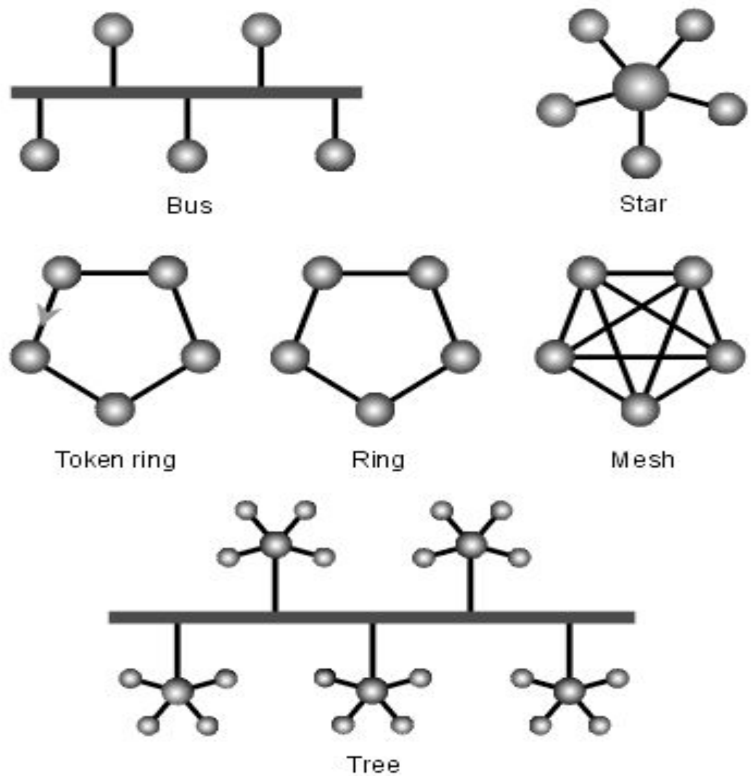
Network topologies: Point to point

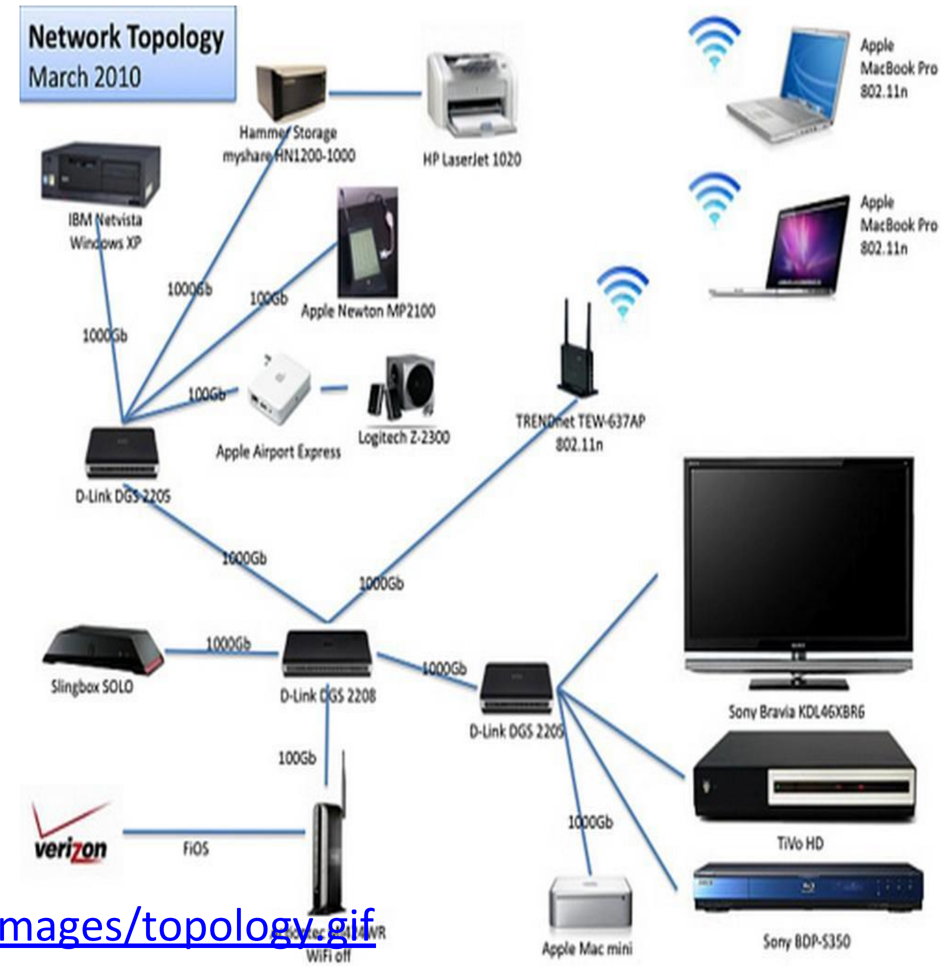


- One machine is connected to the other
- Physical medium: wired or wireless
- What if only above topology was possible?
  - LAN?
  - Scalable to Internet?
  - Cost of management?

Figure source: http://mucins.weebly.com/uploads/7/5/5/4/7554390/6295029.png?524

Topology: Multiple access





Source: http://media.techtarget.com/WhatIs/images/topology.gif

Source: http://3.bp.blogspot.com/-hZC8s_vWm9E/TtTzc8ZC4PI/AAAAAAAAEmI/hrsgQALpgss/s1600/Network+Topology4415377462_7dcefc4876.JPG

Network Cloud

- Logical representation of a network
  - Big or small network
- Instead of whole topology, a cloud is shown
- Can be treated as a node sometimes
- Switched network

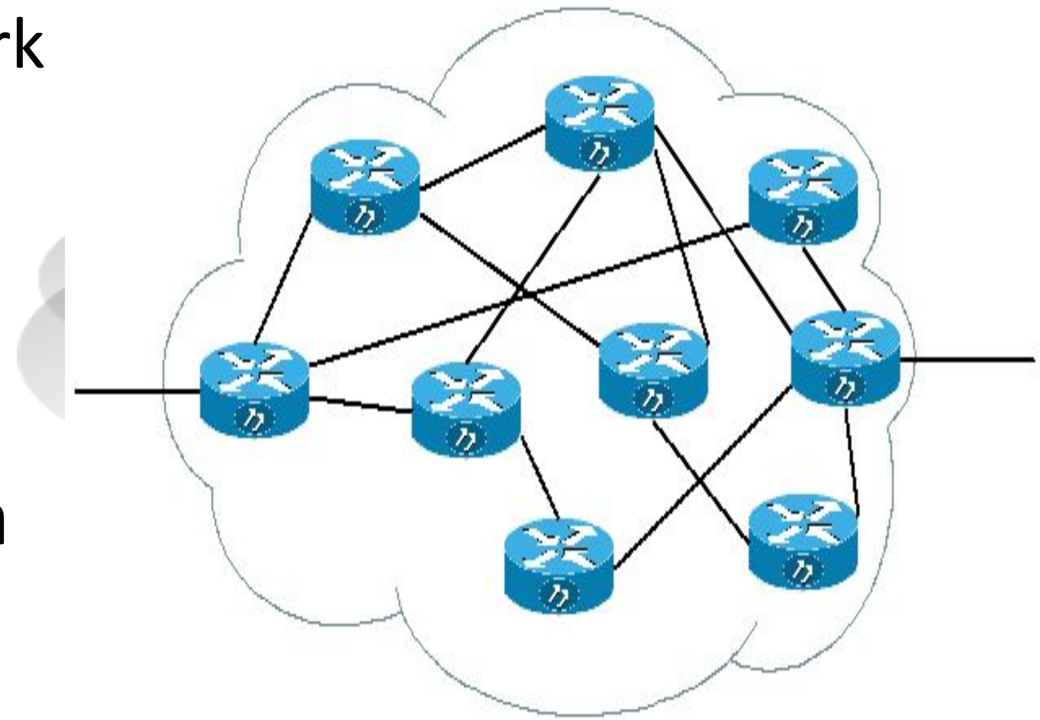


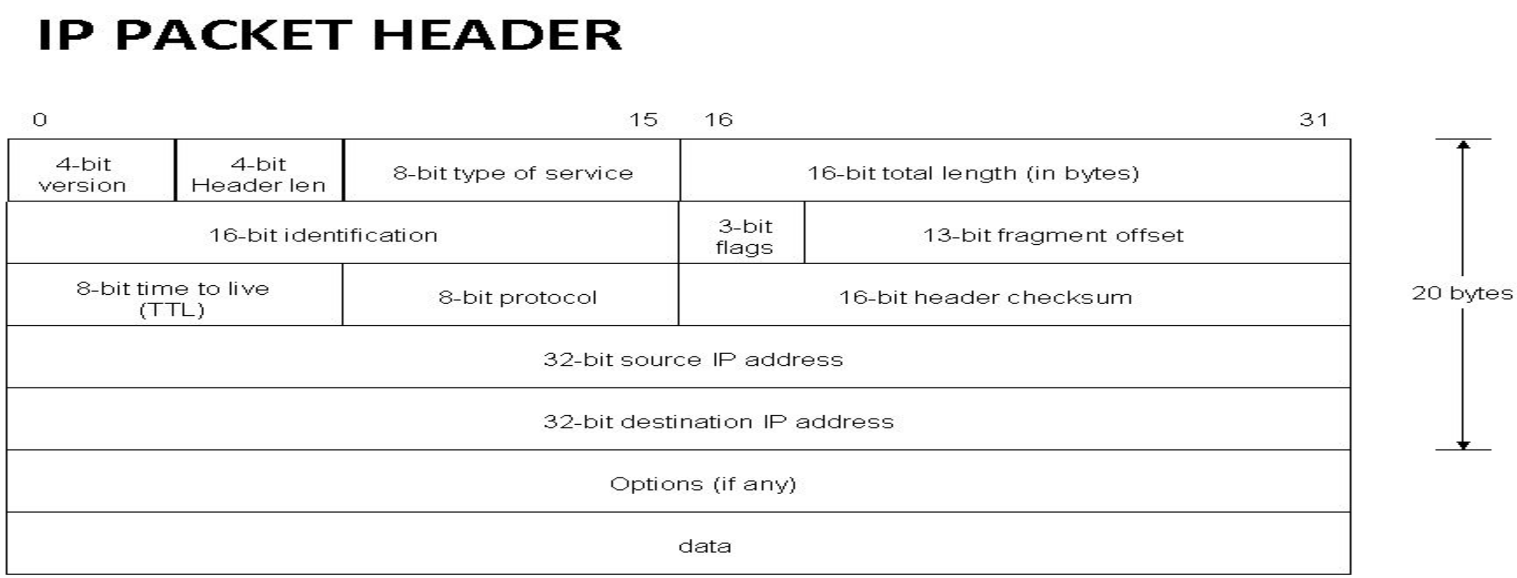
Figure Source: http://www.pwmn.net/wiki/images/c/cc/The_inside_of_a_network.png
http://www.clker.com/cliparts/b/b/2/c/11949839472059691147network_cloud_david_klan_01.svg.med.png

Data transfer

- A packet
  - Block of data that can be sent
  - Unit of transmission in computer networks
  - Even one bit cannot be sent alone
  - Has all the information needed to send data from source to destination
  - This information is put in various fields present in “header”
- Header
  - Header is a dedicated part of a packet that contains essential and optional information

A typical header

- Terminology will be clear as the course advances
  - Basically a data structure to keep information



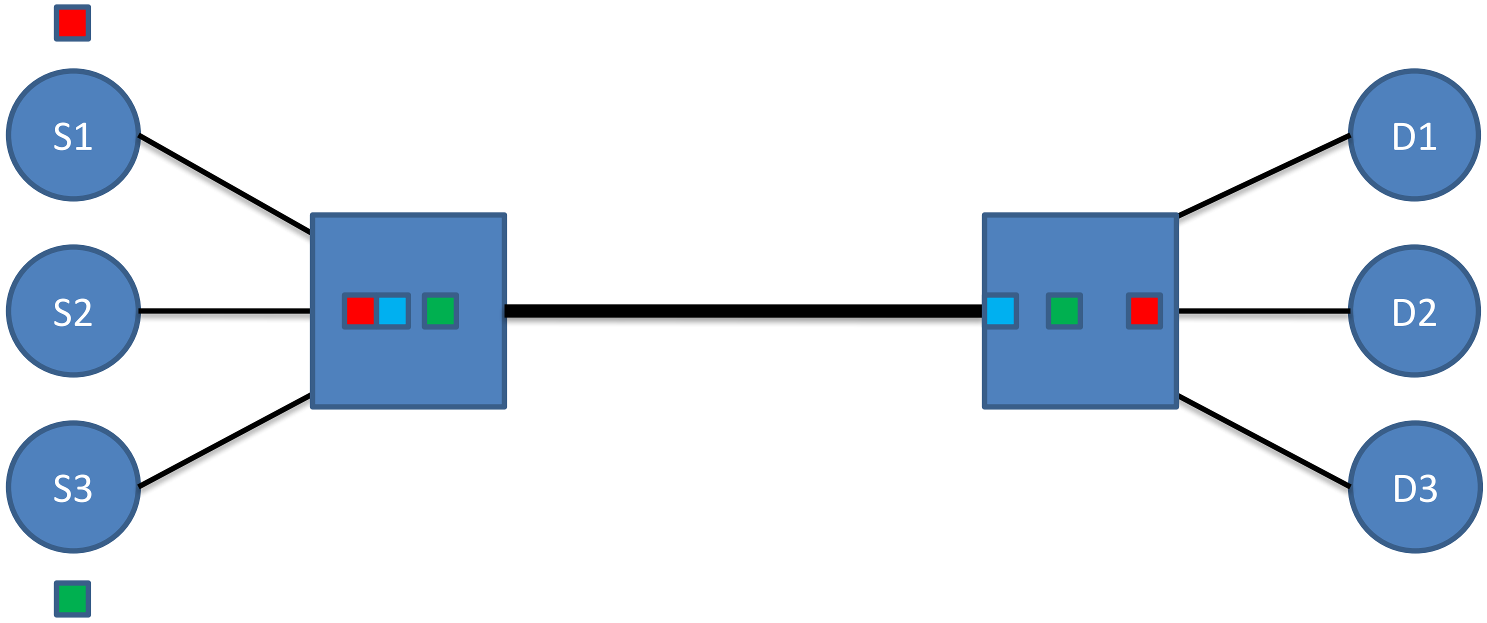
http://media.packetlife.net/media/forum/attachments/36/ip-packet.gif

Switching

- Circuit switching
  - A dedicated line to a connection
  - Telephone lines
  - A line per connection set up before the communication starts
- Packet switching
  - Not necessarily a dedicated line for a connection
  - Multiple packets belonging to different users or applications can be sent
- We focus only on packet switching in this course

An example of packet switching

- Three sources and three destinations



- Not possible in circuit switching

Issues

- Fairness has to be ensured
  - Each host gets chance to transmit
- Congestion
  - Sources are sending data too fast to handle
  - Buffer overflows
  -

Routing

- The process of delivering a packet from a source to a destination
- A destination may not be directly reachable
- Packet goes through many other intermediate nodes before it reaches destination
- Can you think of something that will help here?
  - Shortest path graph algorithms
  - How?
-

Reliability

- Do you think just reaching of a packet to its destination is sufficient?
  - How about errors in transmission?
  - Failure of nodes or links?
  - Or shuffling of the order in which source sent?
  - How does the source know if destination has received data properly?
-

Network Architecture

- Guides design and implementation of networks
  - Switching, routing, reliability, management, etc
  - This systematic design and implementation is referred to as architecture of network
- Hiding complexity from application programmers
- Above abstraction leads to layering
  - Keep a special task at each layer
  - Depend on lower layers, provide service to higher ones
-

Networking layers

- It is messy to tackle all the issues mentioned earlier at once
- Hence, networking architecture is broken into different layers with each layer as its own specialization and function
- Mainly two layering architectures:
  1. OSI networking layers
  2. TCP/IP model or Internet architecture
- Protocols: Objects that make up the layers

Advantages of layering

- An abstraction that divides functionality in different components
  - Decomposition into manageable components
    - Avoids a monolithic piece of software
  - Modularity: Little to change, if required
    - Adding new services is easier
    - Modify only one layer
  - Hide complexity of the network from programmers
- Each layer has its own set of protocols
- Protocols interact with each other to give

Layering and protocols

- Protocols interact with
  - other protocols of different layers on the same machine (service interface)
  - with the same protocol on different machines (peer-to-peer interface)
- Protocol graph: Interaction of protocols can be modeled as a graph
  - Protocols as nodes
  - “Depends on” relationship as edges

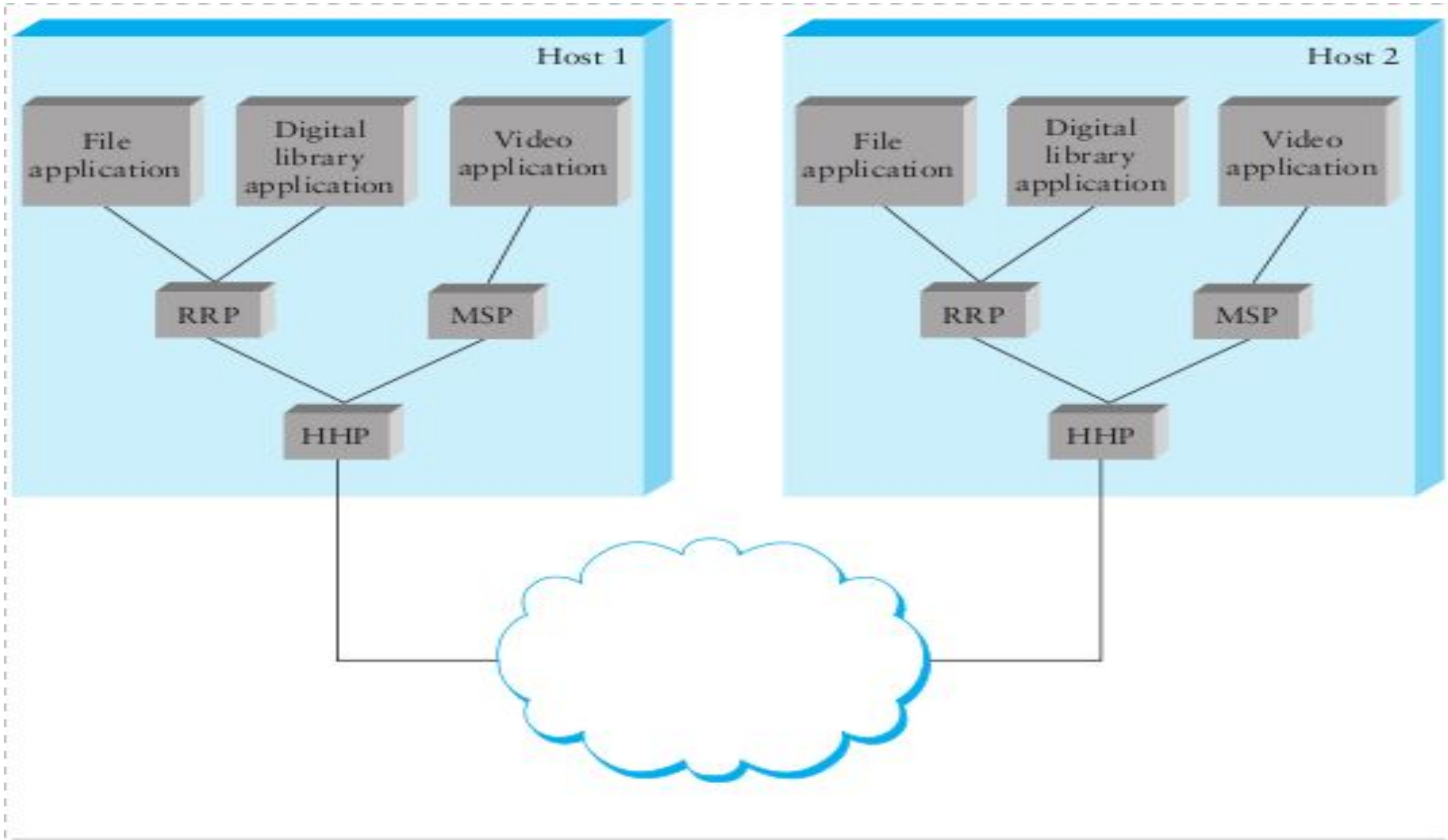


Figure 1.11 Example of a protocol graph.

Protocol graph

For example RRP does not care what kind of bits are sent to it by higher level protocols, email or a jpg image. It simply treats them as bits. Challenging is to make two different implementations of the same protocol to work to successfully exchange messages.

Issues with protocols

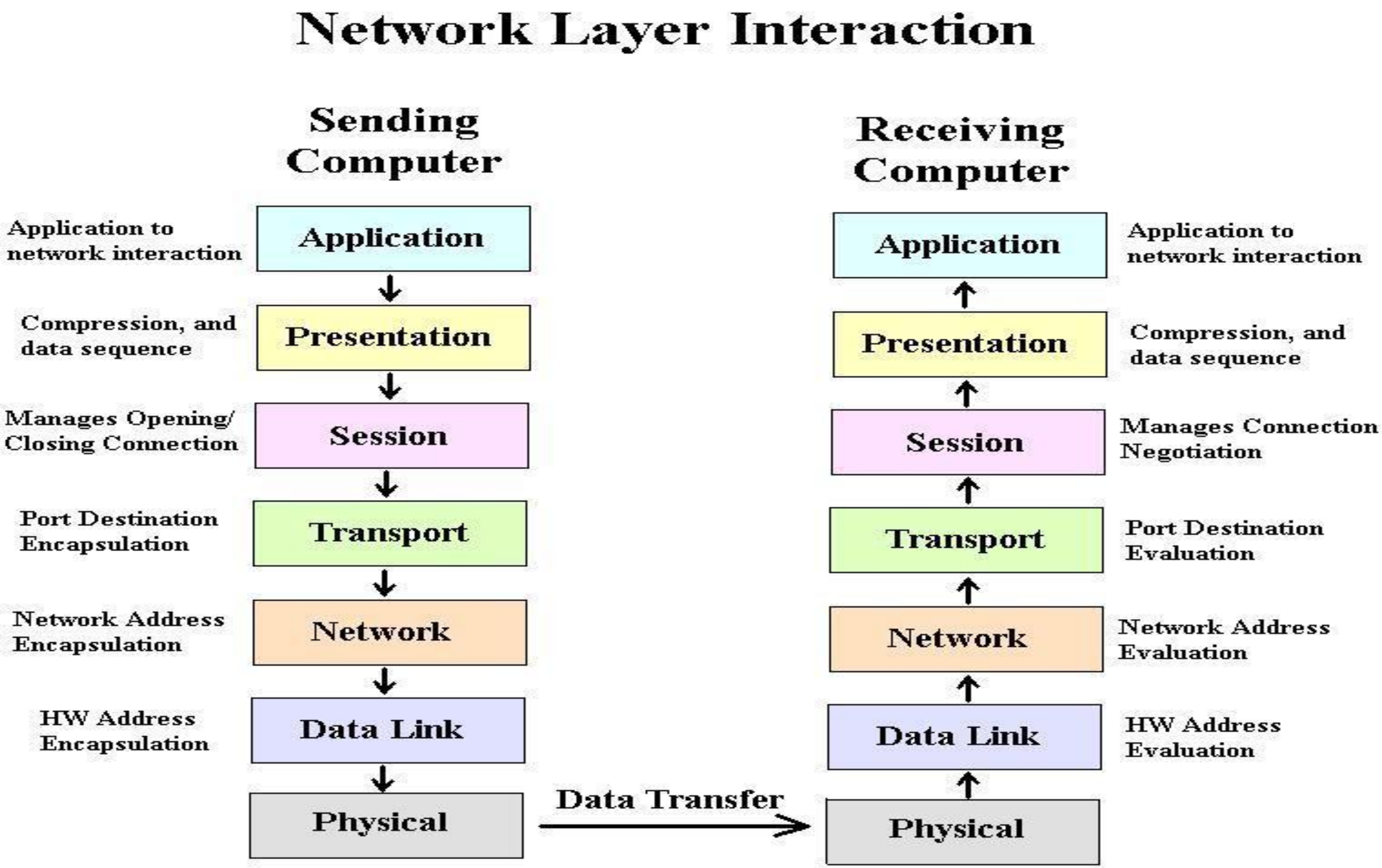
- Abstraction
  - In terms of the services it provides
  - Protocol specifications
- Actual implementation
  - Ensure that all implementations interoperate
- Standardized bodies like IETF
  - Check RFCs of protocols and network various other network specifications

Encapsulation

- Upper layers depend on the lower ones
  - For lower layers, upper layer's data is no more than a stream of bits
    - It acts as data/payload/body
    - This process is encapsulation
- Repeated at each level of protocol graph
- Advantages?
  - Reduces complexity: each layer handles its stuff
- Increases the bits actually transferred

- Header attached by protocol stores which higher layer protocol it belongs to
  - Demultiplexing key, or demux key
- Some protocols use 8 bit some 16 or 32 bit field
- Some use two for both sides of communication
- Network architecture
  - The set of rules governing the form and content of a protocol graph

OSI networking layer



TCP/IP model

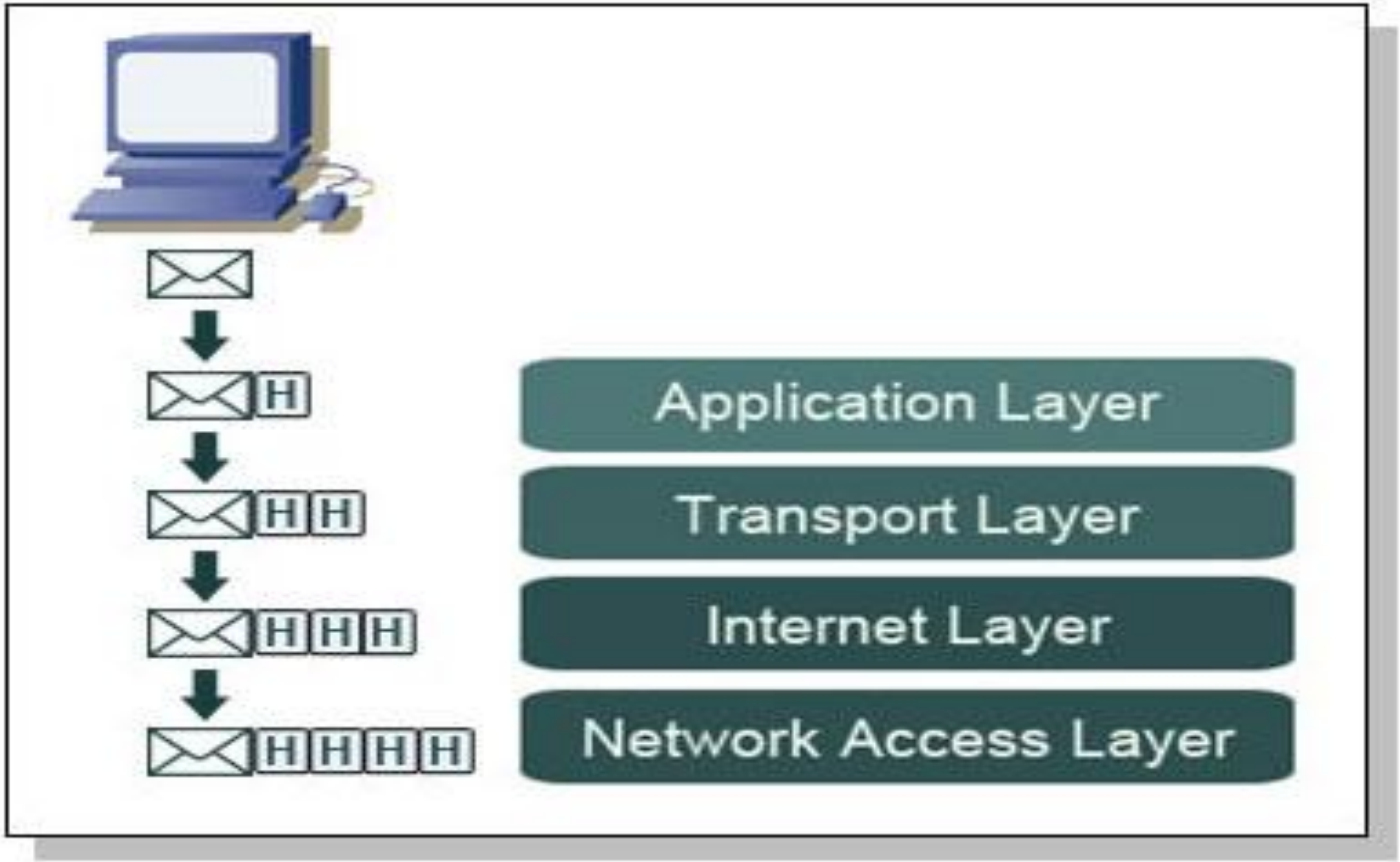


Fig. source: http://www.learn-networking.com/wp-content/oldimages/tcp-ip-headers.jpg

Comparison

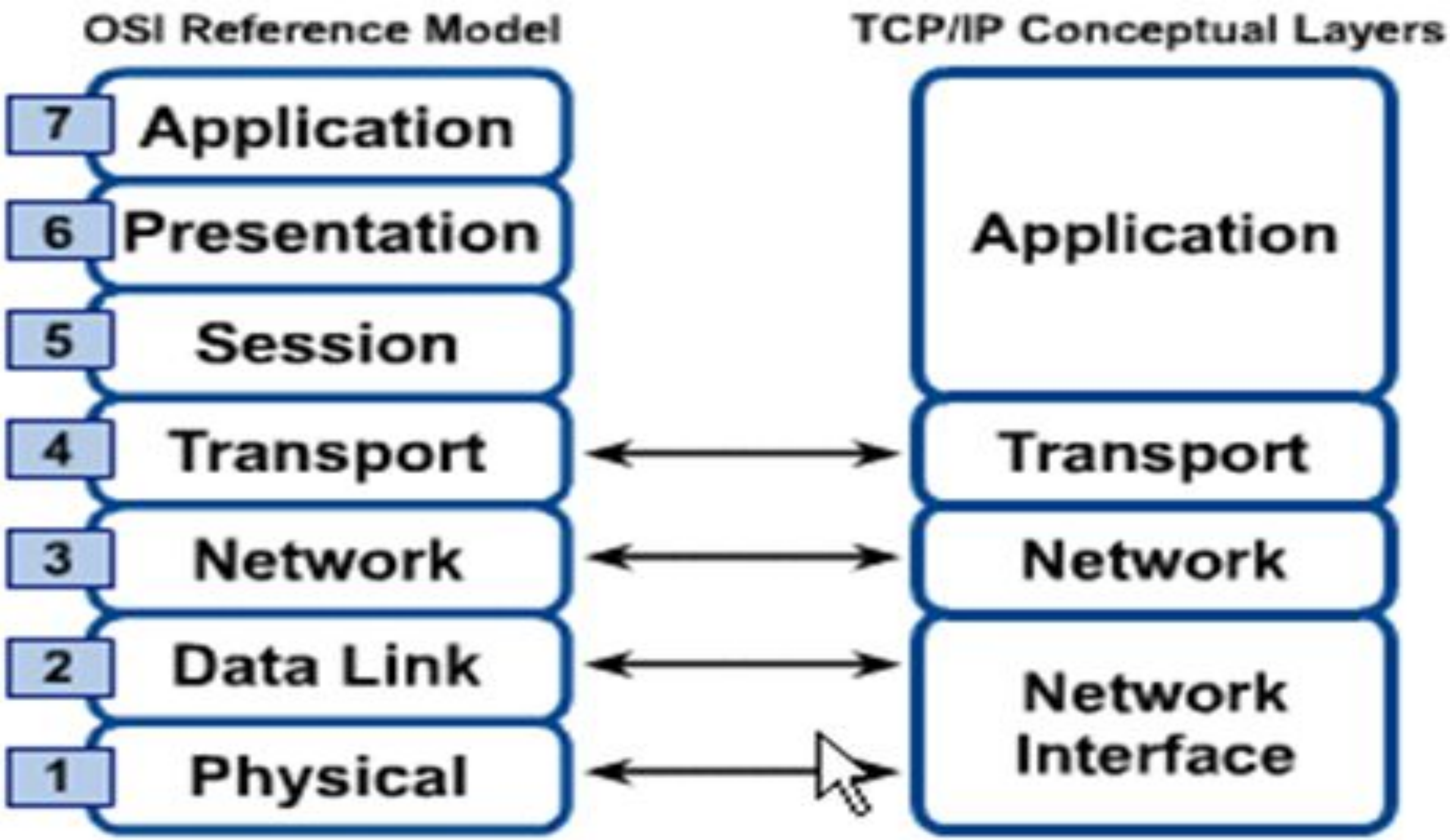


Fig. source: http://www.wtuto.com/cisco/topic-images/osi.PNG

The Internet Architecture

- *We reject kings, presidents and voting. We believe in rough consensus and running code.*

  ---- David Clark

- TCP/IP stack introduced before
- Evolution from the ARPANET
- Three main features
  1. No strict layering
  2. Hourglass shape
  3. For official inclusion of a protocol, a working group at IETF

Performance

- It is not over yet!
  - We have seen a high lever overview of network architecture
  - We need to worry about performance now
- Design for performance
  - Just designing a network is not enough
  - It must perform “well”
- It is measured in two ways:
  1. Bandwidth or throughput
  2. Latency or delay

Bandwidth and throughput

- Number of bits that can be transmitted over a network per unit time
  - E.g. 10 million bits per second (10Mbps)
- Bandwidth at different levels
  - Single physical link
  - Logical process to process channel
- Bit as a pulse of some width
  - E.g. each bit on 1 Mbps link is 1 µs wide
- Throughput: Measured performance of a system
  - A link may have bandwidth of 10 Mbps but

Latency

- How long it takes a message to travel from one end of a network to the other
- Latency of a single link or a process to process communication
- Measured in terms of time
- Time for a message to go to and fro
    - Round Trip Time (RTT)
    - *ping*

Latency

- Three components
  1. Speed of light propagation delay
  2. Time to transmit data
  3. Queuing delays
- Latency = Propagation + Transmit + Queue
  - Propagation = Distance/Speed in medium
  - Transmit = Size/Bandwidth

Bandwidth and latency

- Combine to define performance of a link or channel
- Depends on application though
  - For an application that sends say 1 byte
    - 1 ms RTT vs 100 ms RTT
  - An application that sends 25 MB
    - 10 Kbps vs 10 Mbps

Delay x bandwidth

- Maximum number of bits in transit at a given instant
- It determines how many bits should sender send before the first bit reaches destination
- What does 32 KB data transmitted over a 10 Kbps link mean?
  - What does K in both stand for?
- Increasing bandwidth does not mean latency improves!!
  - RTT of 100 ms is same for 1 Mbps or 1 Gbps link