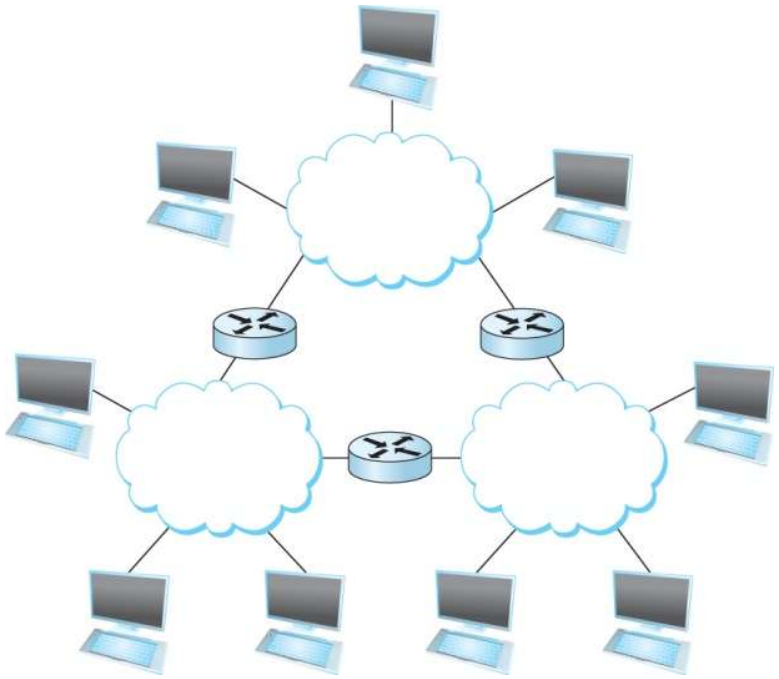
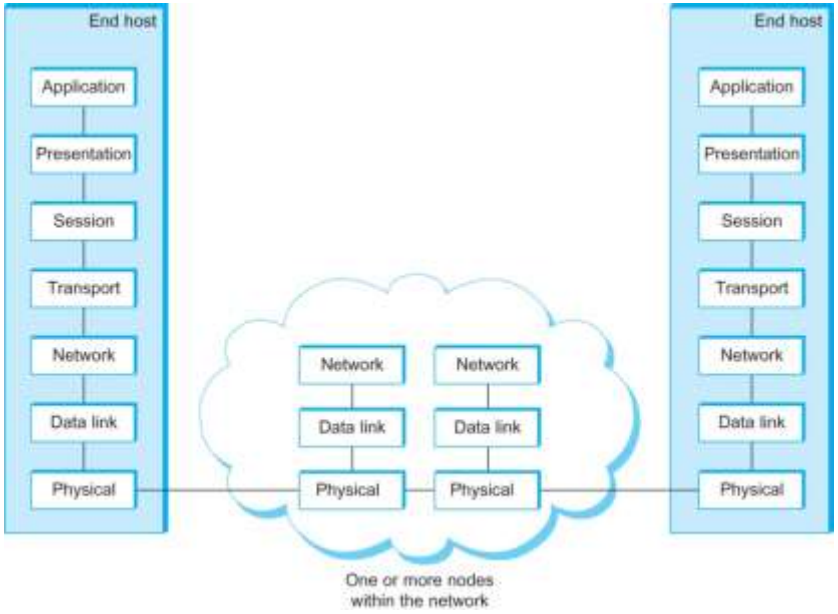
Network applications, requirements and architecture





Overview

- Network applications
  - What kinds of things do we want to do?
- Network requirements
  - What do networks need to have in order to support our desired apps?
- Network architecture
  - How do we break things down to handle the complexity?

Applications

- For most people, the Internet = applications
  - Web surfing
  - Email
  - Social networking (Facebook)
  - Broadcast audio and video (YouTube, Netflix)
  - Two-way audio and video (Skype)
  - File sharing (BitTorrent)
  - Instant messaging (Twitter)

Applications

- Virtual workplace (Amazon Mechanical Turk)
- Mega-mega mall (Amazon.com)
- Global supercomputer (SETI@home)
- Virtual reality (Second Life)
- Online gaming (World of Warcraft)
- Online voting
- Online whistleblowing
- ???

Requirement stakeholders

- Application programmer
  - Types of services needed by application
    - How much data is being sent?
    - How fast must it get there?
    - Guaranteed delivery?
    - Security?
- Network designer
  - Cost effective network with shared resources
- Network provider
  - Network that is easy to manage

Requirement stakeholders

- End user
  - How easy is it to use?
  - How expensive is it to use?
  - How fast is it to use?
  - How consistent is it to use?
  - ???

4 requirements for networks

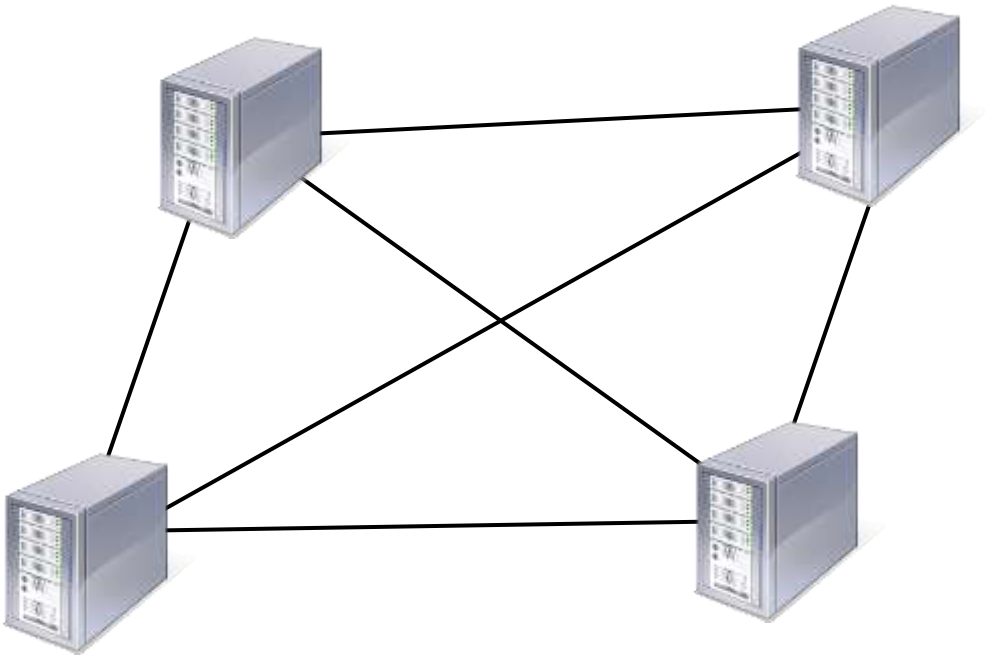
1. Scalable connectivity

- ***Nodes*** (i.e. computers) in a network must be able to communicate
- Support arbitrarily large networks



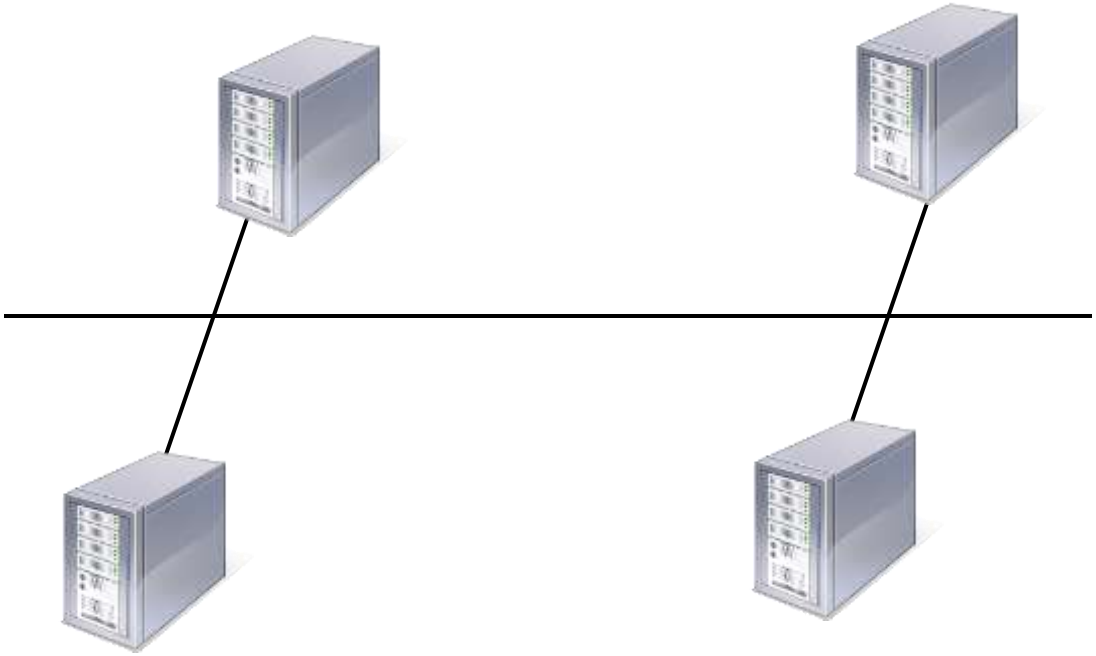

Point-to-point

- Connect with ***point-to-point*** cables
- Simple but not scalable
  - Given n computer, (n-1) ***links*** to each



Multiple-access

- Connect using ***multiple-access***
- Only one cable per computer
- Problem: contention for medium

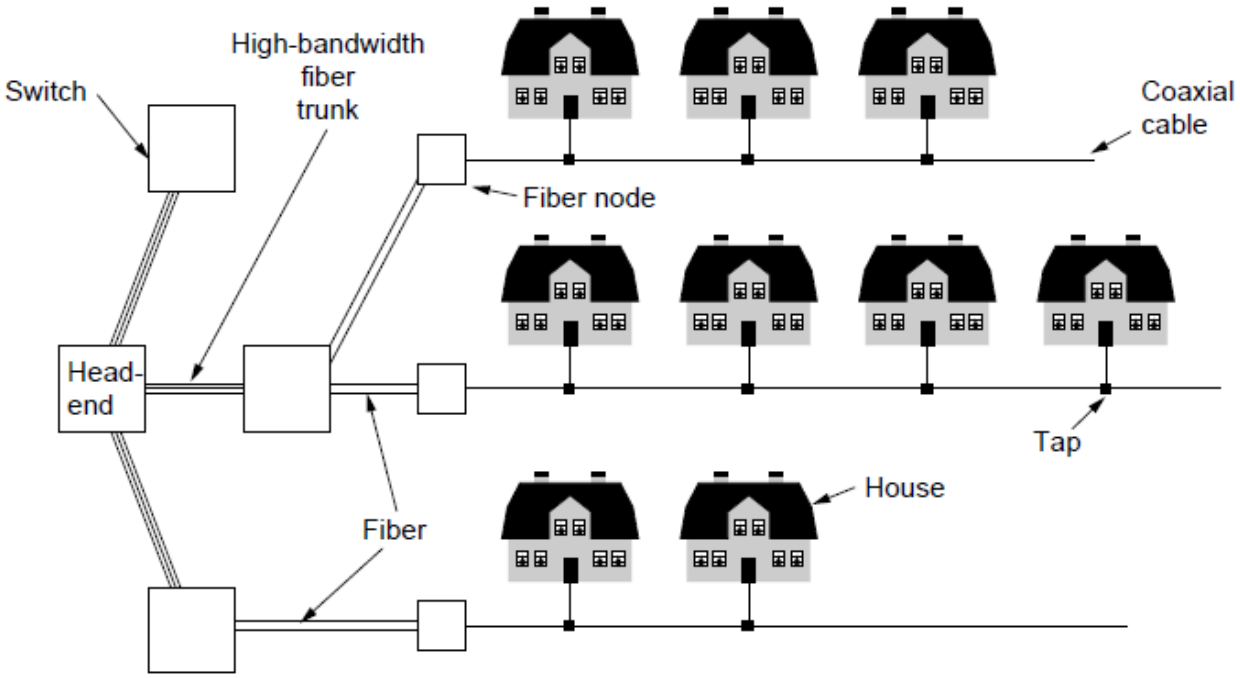


Multiple-access

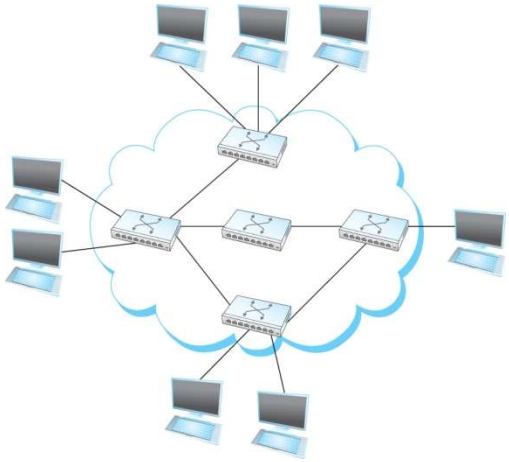
- Shared medium
  - No longer common for wired Ethernet
  - Very common for wireless

Multiple-access

- Shared medium
  - Also common for Internet over cable
  - 500-2000 houses on a cable

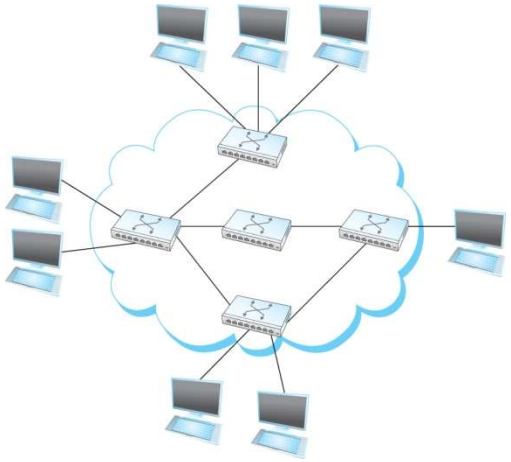


Switched networks



- ***Circuit switched***
  - Initial setup of path through network
  - Every ***packet*** between node X and Y, same route
  - Example: public switch telephone network (PSTN)
  - Also increasingly in optical networks

Switched networks

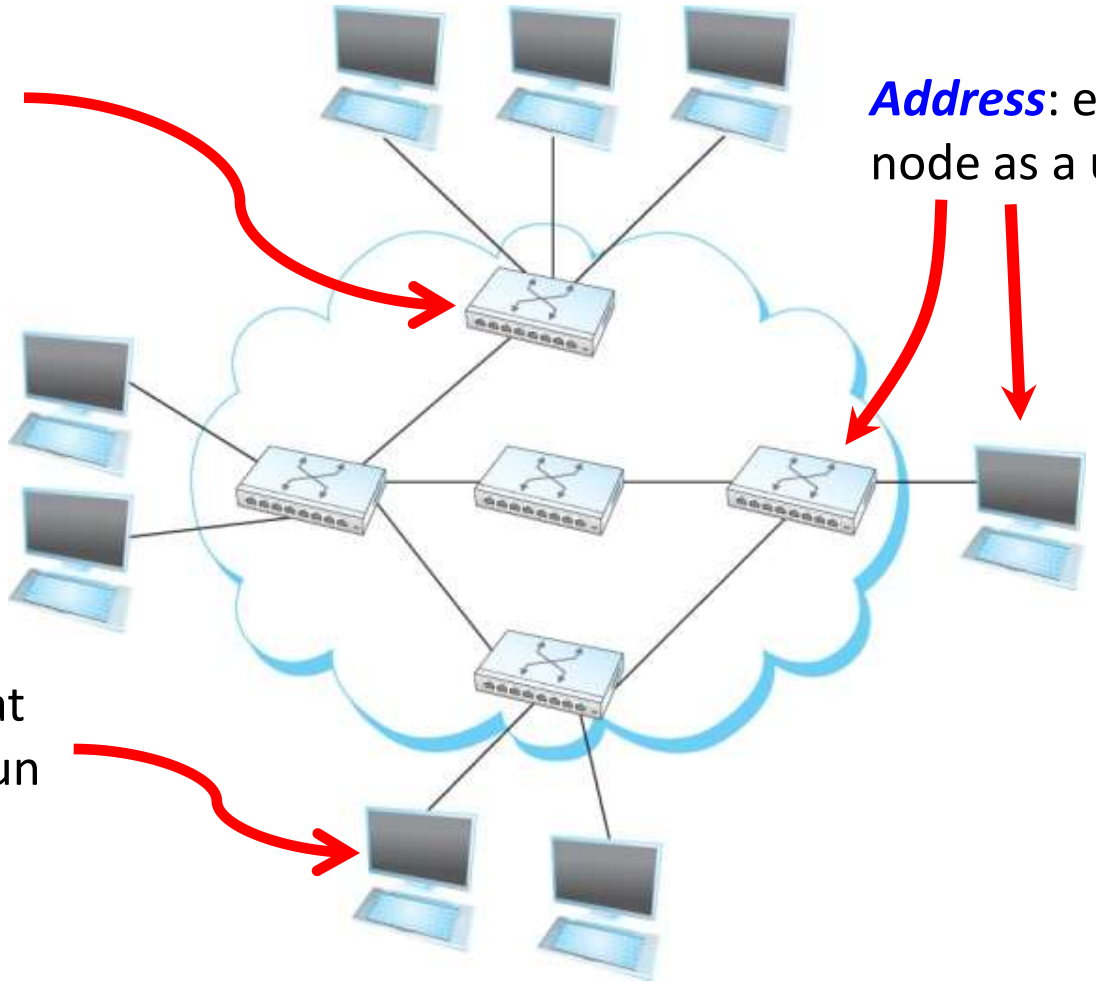


- ***Packet switched***
  - Packets between X and Y may take different routes
  - Example: most computer networks, Internet
  - Typically use ***store-and-forward***

Switched networks

Switches: nodes that implement the network

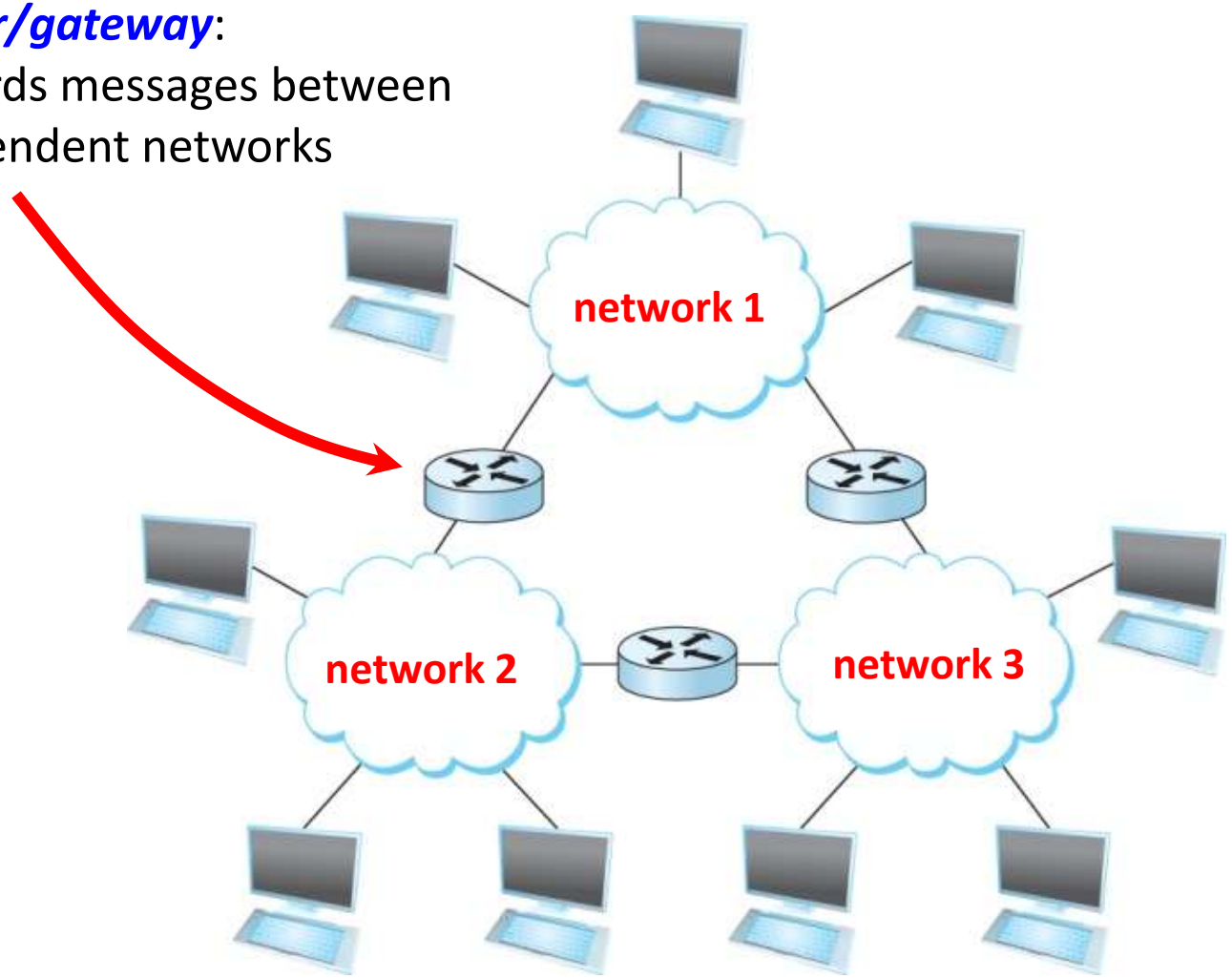
Address: every node as a unique ID



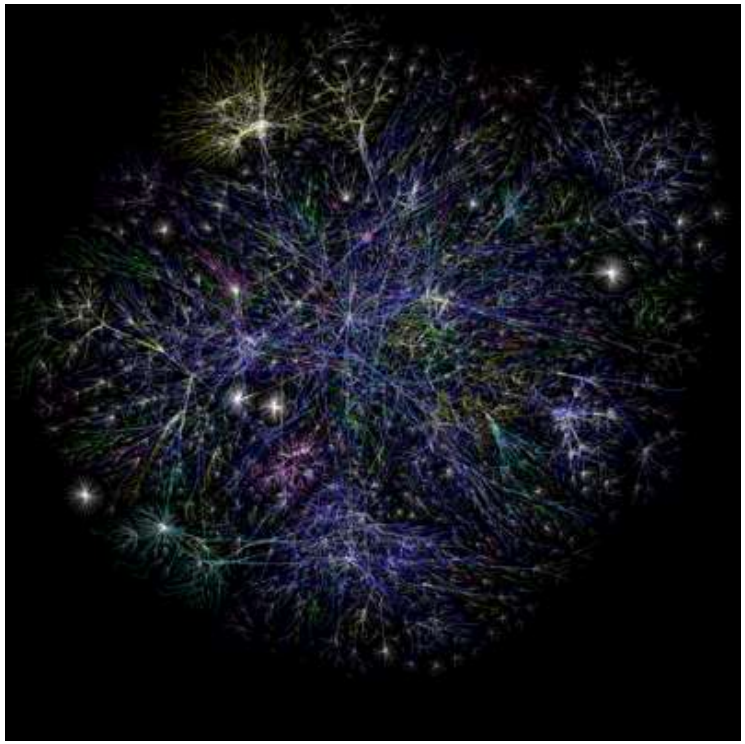
Hosts: nodes that support users, run applications

internetwork / internet

Router/gateway:
forwards messages between independent networks



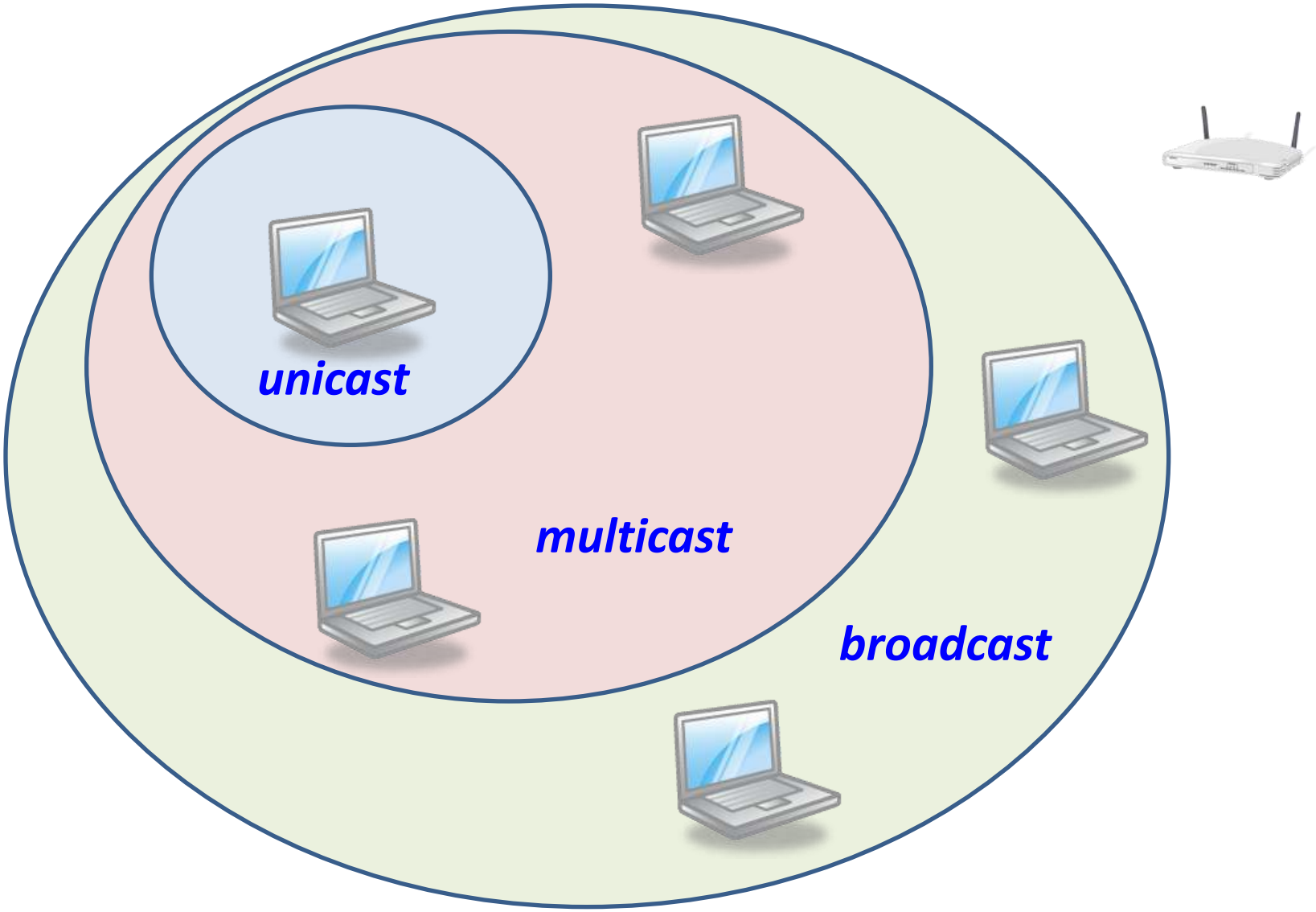
internet vs. Internet



internetwork / internet:
A set of independent networks interconnected, could be completely walled off from world.

Internet
Global public network consisting of interconnected networks running TCP/IP.

Message destination

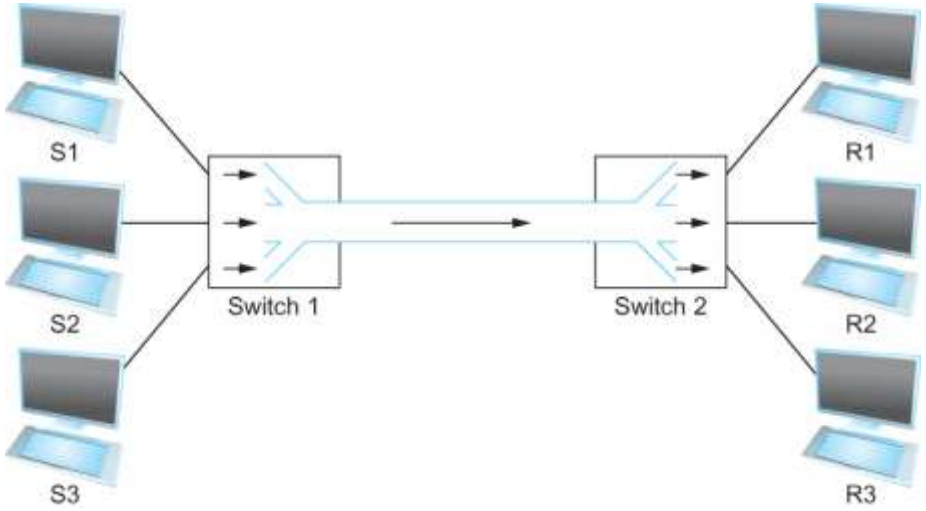


?ANs

PANs	Personal area networks A few meters
LANs	Local area networks < 1 km
MANs	Metropolitan area networks Spans a city or large campus
WANs	Wide area networks Worldwide
SANs	Storage area networks Specialized high-performance network for providing storage

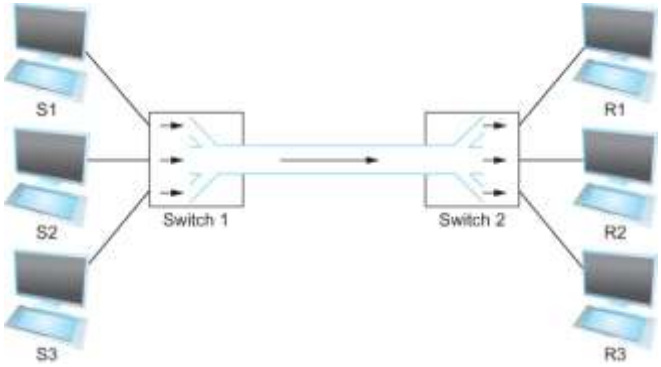
2. Cost-effective resource sharing

- How do we share a link?
  - ***Multiplexing*** - sharing a resources by combining
  - ***De-multiplexing*** - splitting back apart



Type of multiplexing

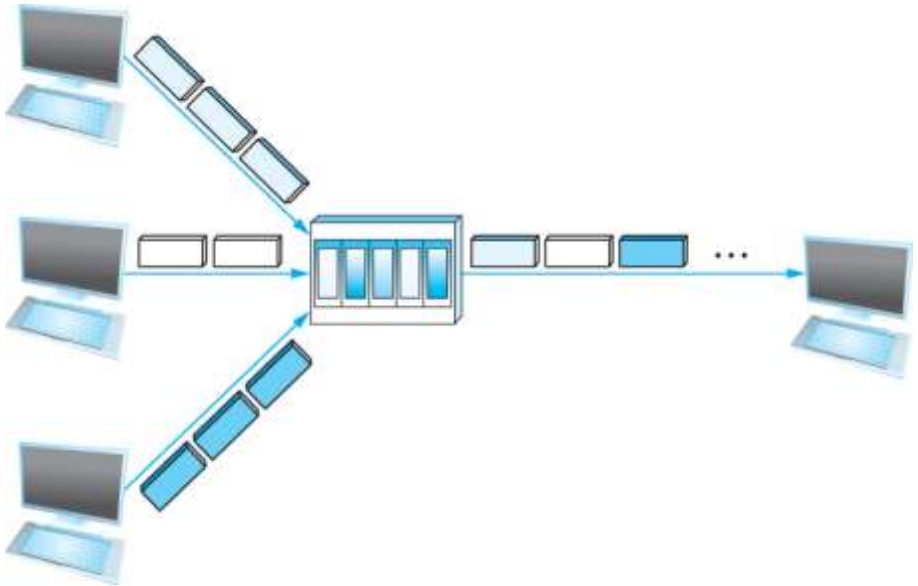
- ***Synchronous time-division multiplexing*** (STDM)
  - Divide time up and go round-robin
- ***Frequency division multiplexing*** (FDM)
- Problems with STDM, FDM:
  - Node wastes link slot if idle
  - Need to know number of nodes ahead of time



Type of multiplexing

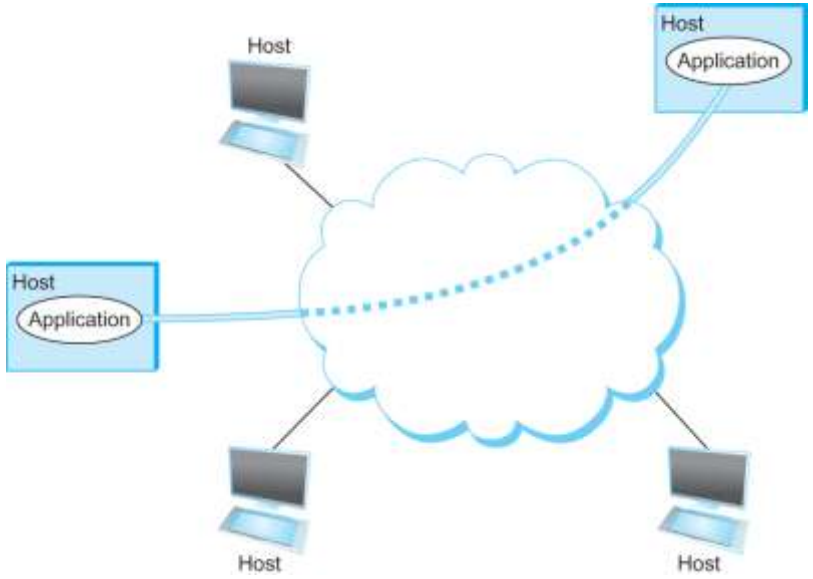
– ***Statistical multiplexing***

- Nodes split arbitrarily long ***messages*** into ***packets***
- Link decides which packet to send next
- Different scheduling algorithms: FIFO, round-robin, quality of service (QoS) based



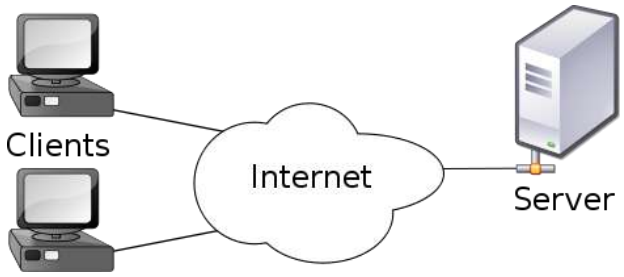
3. Support for common services

- Logical ***channels***
  - App-to-app communication path / pipe
  - Allows app to ignore details of network



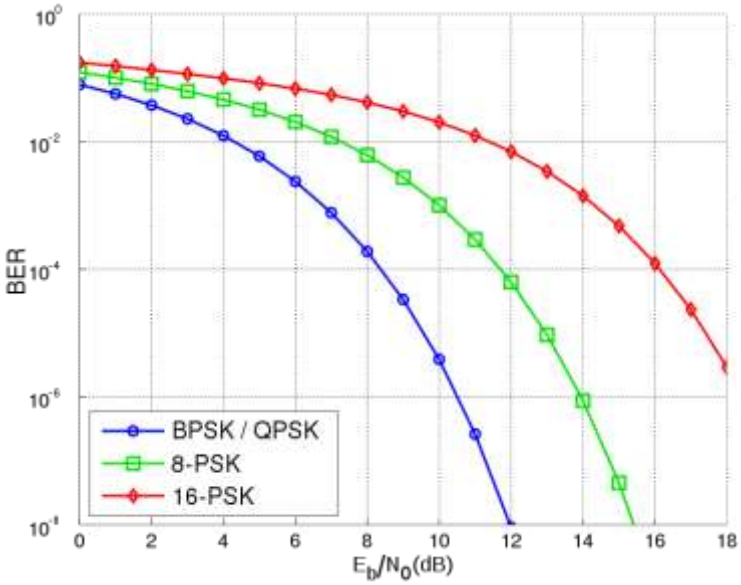
How nodes talk

- ***Client*** - makes a request
- ***Server*** - provides the data
- Communication patterns
  - ***Request/reply***
    - Guaranteed 1-to-1 delivery
  - ***Message stream***
    - May not need guaranteed delivery (e.g. videoconferencing)
    - May want multicast



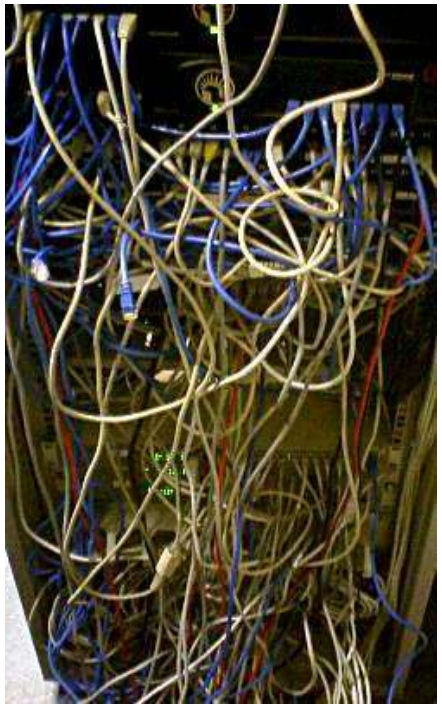
Reliability

- Network should hide errors
  - ***Bit errors*** - 1 flips to 0, 0 flips to 1
  - ***Burst errors*** - several errors in a row
- Packets lost
  - Unrecoverable bit error
  - Congestion at a switch
- Packets delayed
- Links or nodes may fail



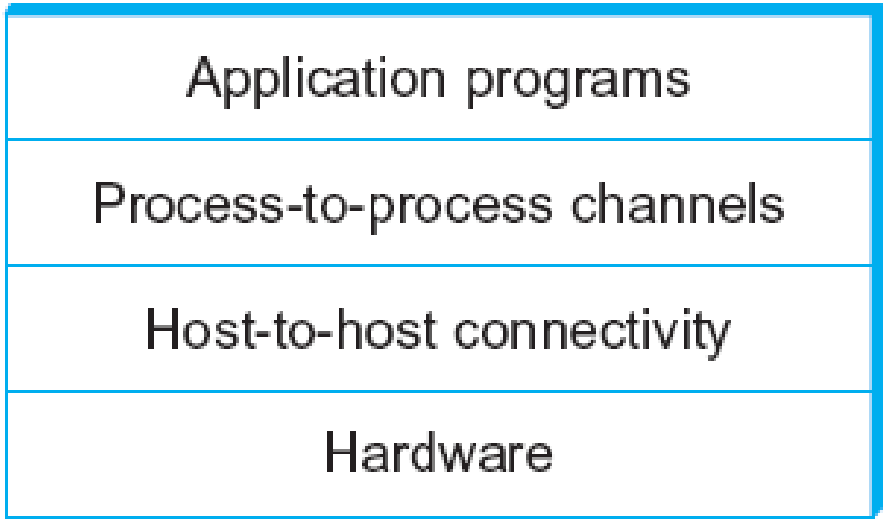
4. Manageability

- Someone has to manage the network
  - Troubleshoot failures
  - Find performance bottlenecks
  - Configure large numbers of nodes
  - May be a non-expert



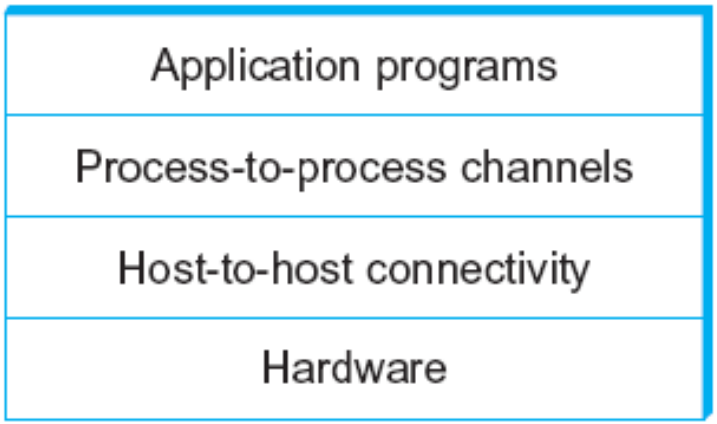
Network architecture

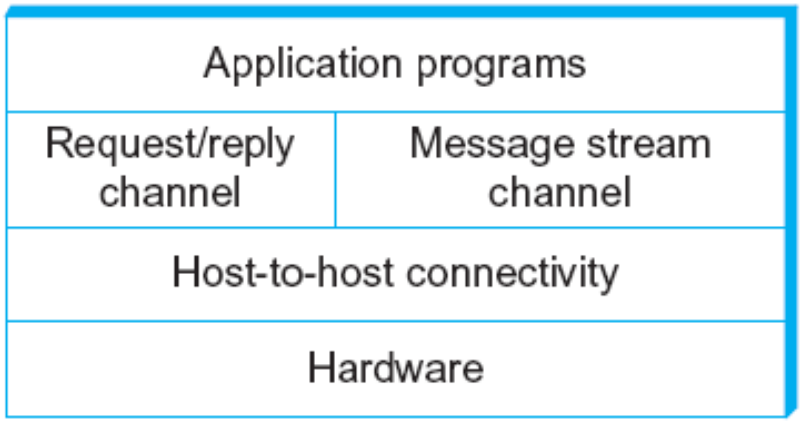
- How to manage complexity?
  - Use computer science secret sauce: abstraction
  - Layers hide details
    - Ignore details at layers above and below



Network architecture

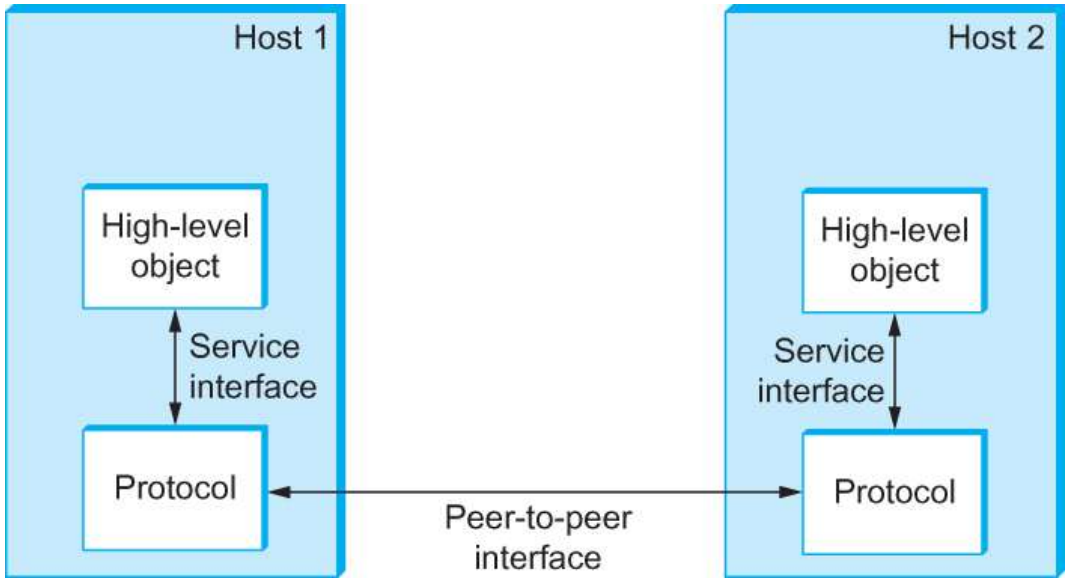
- May be alternates at a given layer:





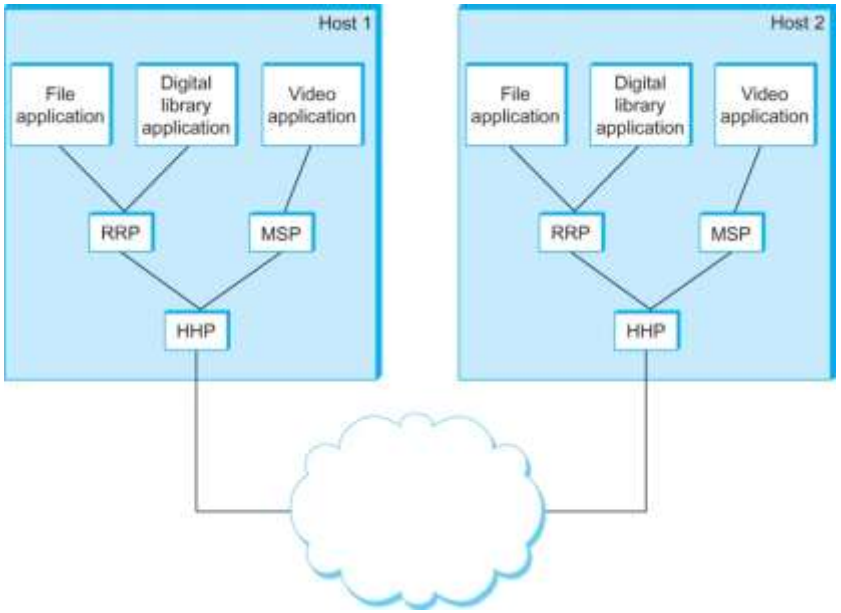
Protocols

- ***Protocol*** provides a communication service
  - ***Service interface*** - layer above talking to you
  - ***Peer-to-peer interface*** - talking to counterpart at same layer



Protocol graphs & stacks

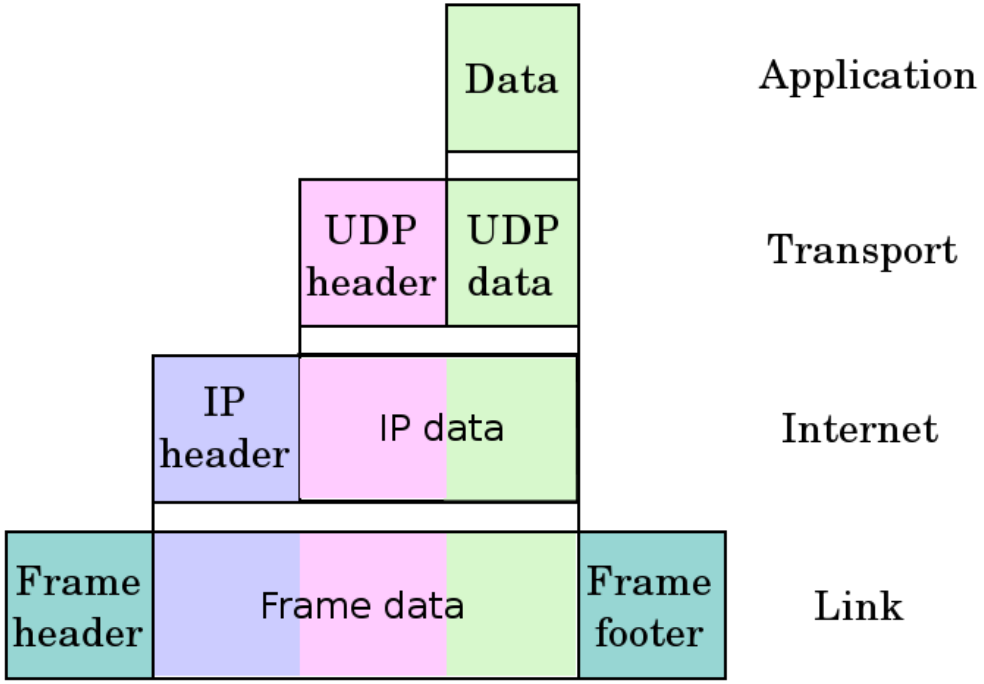
- ***Protocol graph:***



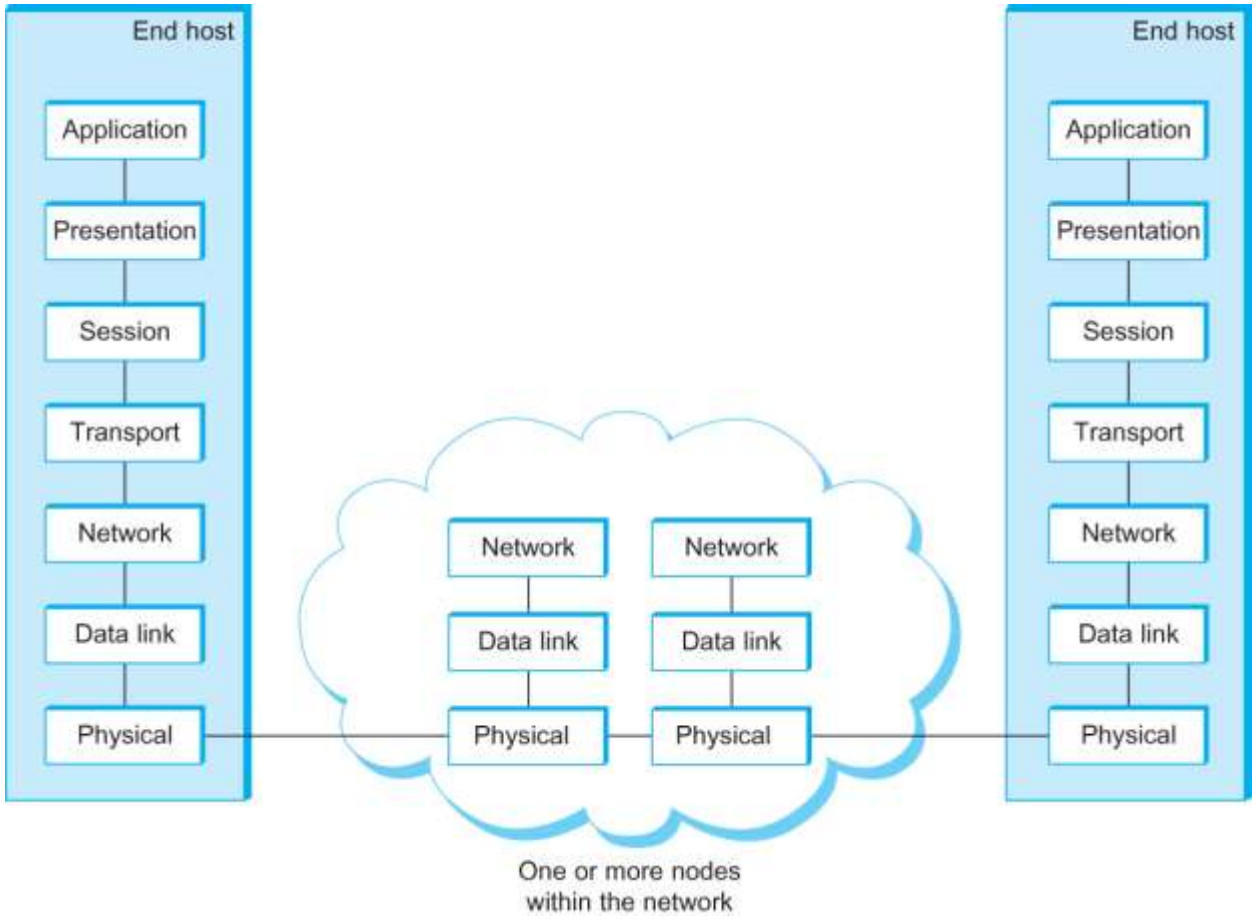
- ***Protocol stack*** - given set of protocols
  - RRP/HHP to send msg from Host 1 file app to Host 2 file app
  - e.g. HTTP/TCP/IP

Encapsulation

- High-level messages ***encapsulated*** in low-level messages
  - headers/footer get added by each layer

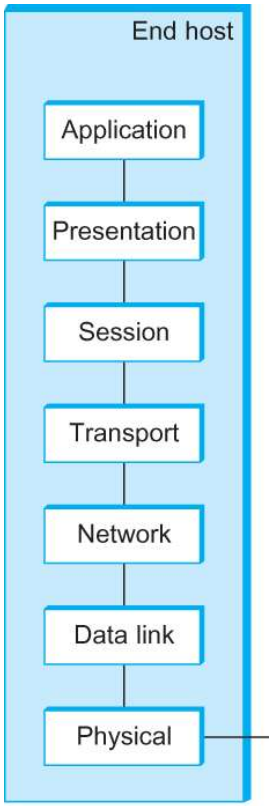


OSI 7-layer model



OSI 7-layer model

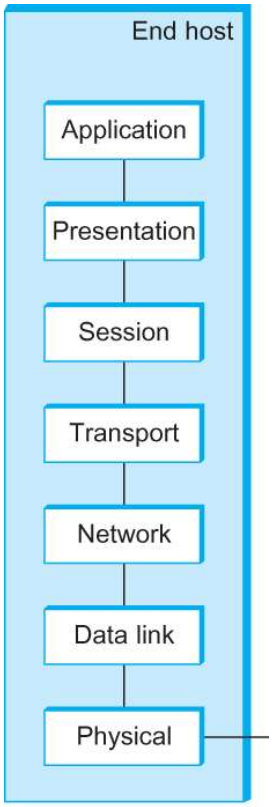
- Physical layer
  - Transmission of raw bits
- Data link layer
  - Aggregate bits into ***frames***
  - Network adapter + device driver
- Network layer
  - Message called a ***packet***
  - Routes in a packet-switched network



These three layers are implemented on all network nodes!

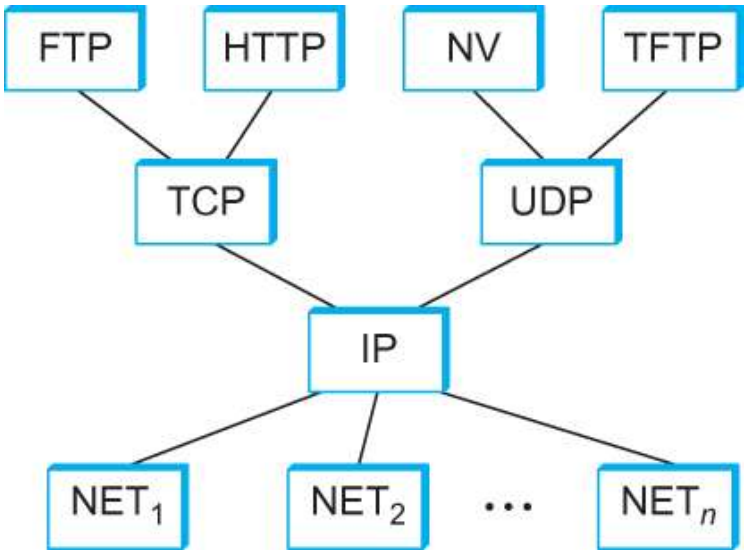
OSI 7-layer model

- Transport layer
  - Process-to-process channel
  - Unit of data called a ***message***
- Session layer
  - Name space for tying different streams
  - e.g. audio and video in a conferencing app
- Presentation layer
  - Format of data between peers
  - Big/little endian, bit width of integer
- Application layer
  - Standardize common types of exchanges
  - Protocols like HTTP, SMTP, IMAP

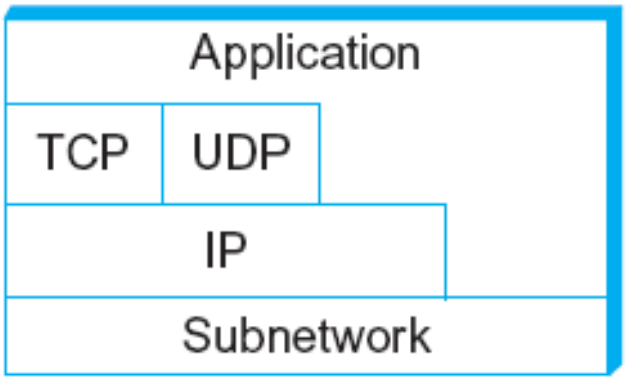


Internet architecture

- Popular 4-layer model
- All roads go through IP



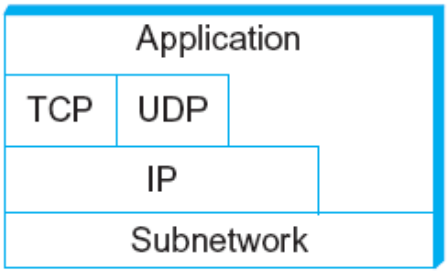
Internet protocol graph



Another view of the Internet architecture. Subnetwork is often called network or link layer.

Internet architecture

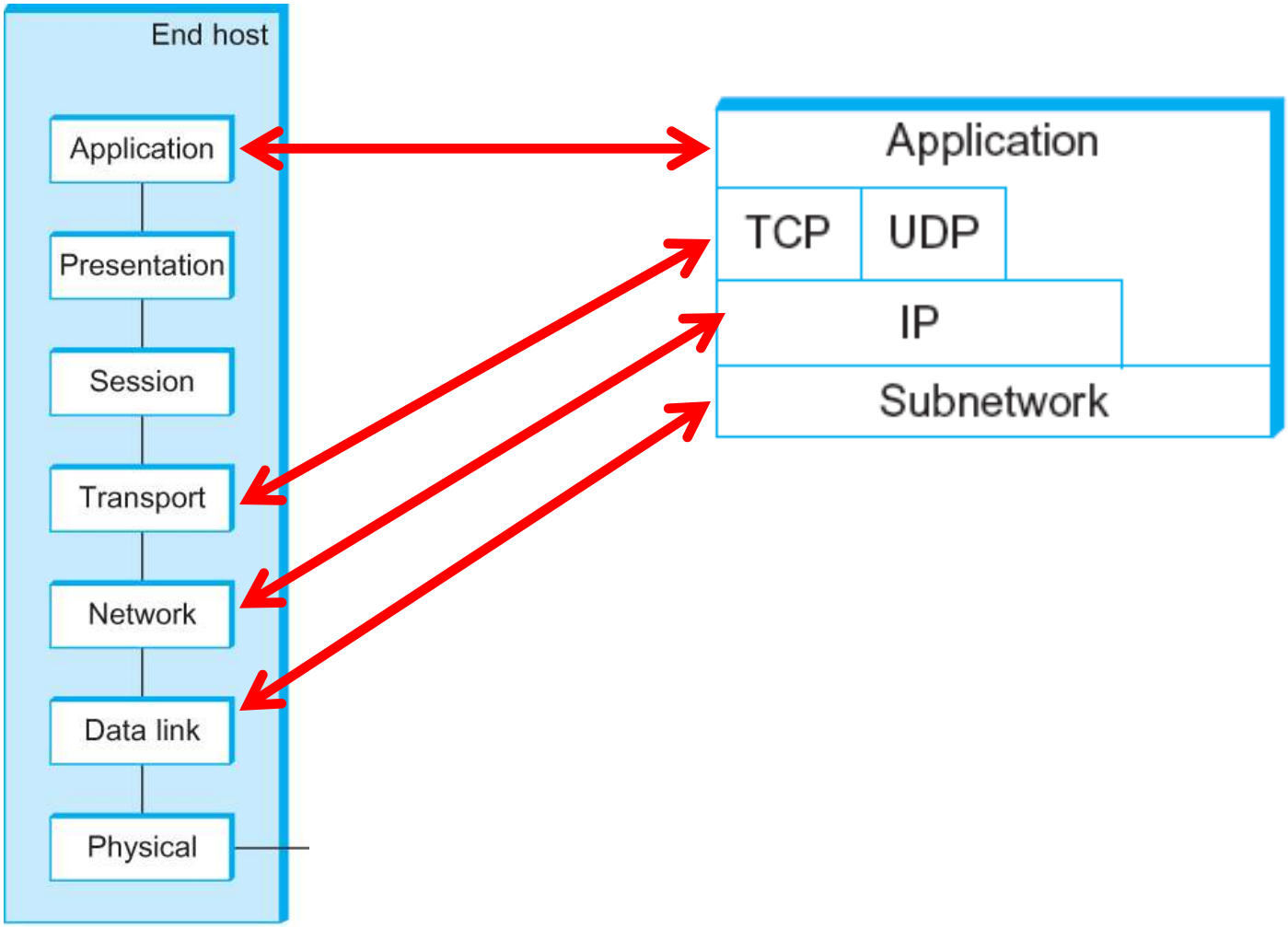
- No strict layering, application can bypass layers if need be



- New protocols: approval requires specification and working code
  - ***Internet Engineering Task Force*** (IETF)

"We reject kings, presidents and voting. We believe in rough consensus and running code."
-David Clark

Layers compared



Summary

- Networks support wide-variety of apps
- Many stakeholders:
  - Programmer, network designer/provider, end-user
- Scalable, cost-effective, reliable connectivity supporting common services
- Protocols, graphs & stacks
- 7-layer OSI and 4-layer Internet architectures