CSE331: Introduction to Networks and Security

Lecture 4

Fall 2006

Announcements

- First project: Due: 25 Sept. 2006
  - http://www.cis.upenn.edu/~cse331/project1.html
  - Please e-mail savi@seas.upenn.edu with your project group by Weds. 13 Sept.
  - If you need a group, send e-mail
- Please put "cse331" in the subject of all course-related e-mail
- Prof. Zdancewic will be away Sept. 18 & 20.
  - Class will be taught by Peng Li

Plan for Today:

- Simple network performance characteristics
- Start looking at the "protocol stack"

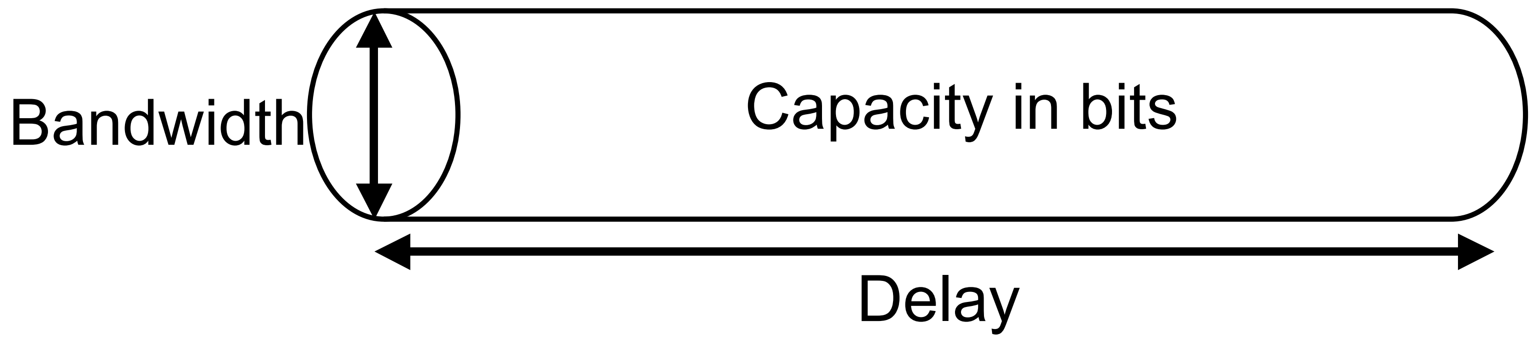
Performance

- *Bandwidth* (throughput)
  - The number of bits that can be transmitted over the network in a certain period of time.
  - Measured in bits/sec
- *Latency* (delay)
  - How long it takes a single bit to propagate from one end of the network to the other.
  - Measured in seconds
- Round Trip Time (RTT)
  - How long it takes for a bit to get from one end of the network to the other *and back*.

Connectivity: Direct Link Technologies

Wired Ethernet	10, 100 Mbps, 1, 10 Gbps
SONET fiber Synchronous Optical Network	up to 9.6 Gbps
CATV Cable TV	1-6 Mbps, asymmetric
ADSL Asymmetric Digital Subscriber Line	Downstream: 1.5-55.2 Mbps Upstream: 16-640 Kbps
ISDN Integrated Services Digital Network	64 Kbps*n with bonding
POTS Plane Old Telephone Service	56 Kbps
Wireless Ethernet	2, 11, 22, … Mbps
Infrared IrDA	115 Kbps to 4 Mbps
CDPD Cellular Digital Packet Data	19.2 Kbps

Performance: Delay x Bandwidth



Delay x Bandwidth determines the number of bits that can be "in flight".

For efficient resource usage: keep the pipe full.

Key Equations

Latency = Propagation + Transmit + Queue

Propagation = Distance / SpeedOfLight

Transmit = Size / Bandwidth

Total Latency: Direct Link

Data moves through the link at the speed of light.

Time

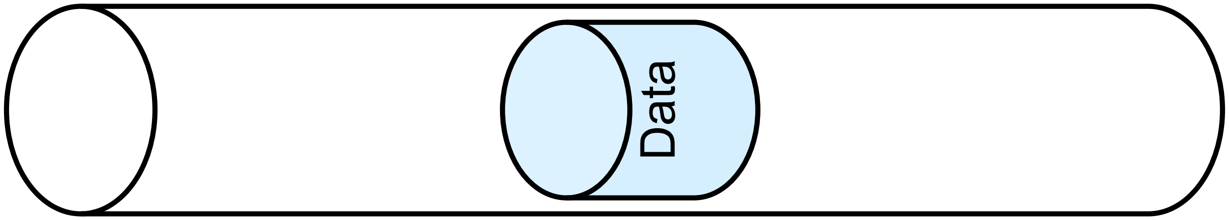
0 Data ready to be sent

Total Latency: Direct Link

Data moves through the link at the speed of light.

Time	
0	Data ready to be sent
t = Size/Bandwidth	Data in the link

Total Latency: Direct Link



Data moves through the link at the speed of light.

Time	
0	Data ready to be sent
t = Size/Bandwidth	Data in the link
t+k	Data traveling through the link

Total Latency: Direct Link

Data moves through the link at the speed of light.

Time	
0	Data ready to be sent
t = Size/Bandwidth	Data in the link
t+k	Data traveling through the link
prop = Distance/LightSpeed	First bit arrives at destination

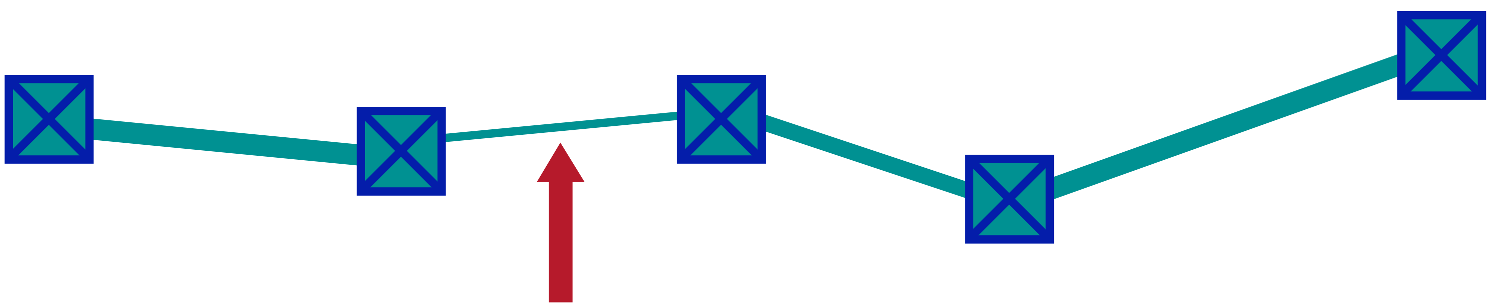
Total Latency: Direct Link

Data moves through the link at the speed of light.

Time	
0	Data ready to be sent
t = Size/Bandwidth	Data in the link
t+k	Data traveling through the link
prop = Distance/LightSpeed	First bit arrives at destination
prop + t	Last bit arrives at destination

Paths Are Made of *Links*

- Links are interconnected by zero or more *network elements*, e.g., switches, routers, hubs, bridges, etc.
- Path delay is sum of link delays plus queuing (switching) delays
- Path throughput = *bottleneck link* t'put



Tradeoffs

- RTT from Penn to Stanford is approx. 100ms
- 1.4 GHz workstation
  - 140 million cycles elapsed in that time
- Data compression
  - Trades machine cycles for bandwidth
- (Question: Why is RTT important?)

Bandwidth vs. Latency

- Which is the better deal:
  - Improve your *bandwidth* from 1 Mbps to 100 Mbps, or
  - Improve your *RTT* from 100 ms to 1 ms?
- The answer depends on what you need to send.

Latency Bound

- Send 1 byte

Transmit Time	
1 Mbps	8 μs
100 Mbps	.08 μs

Perceived Latency	100 ms	1 ms	
1 Mbps	100.008 ms	1.008 ms	99%
100 Mbps	100.00008 ms	1.00008 ms	99%
	.008%	.8%	

Bandwidth Bound

- Send 25 MB

Transmit Time	
1 Mbps	3.5 min
100 Mbps	21 sec

Perceived Latency	100 ms	1 ms	
1 Mbps	210.1 sec	210.001 sec	.05%
100 Mbps	21.1 sec	21.001 sec	.5%
	90%	90%	

Some Units and Measurements

- Mbps = 10^6 bits/sec
- byte = 8 bits
- KB = 2^{10} bytes (= 8,192 bits)
- MB = 2^{20} bytes (= 8,388,608 bits)
- ms = 10^{-3} seconds
- μs = 10^{-6} seconds

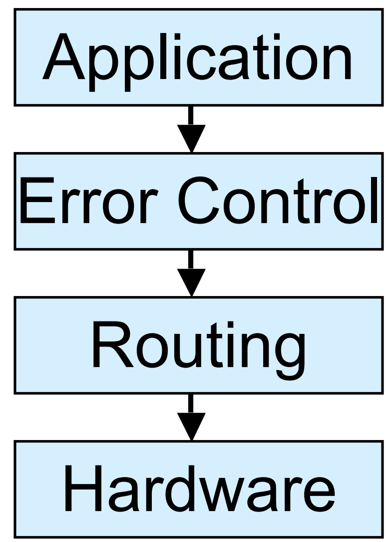
- Speed of light:
  - Vacuum : 3×10^8 m/sec
  - Copper or Fiber: 2×10^8 m/sec

Network Architecture

- General blueprints that guide the design and implementation of networks
- Goal: to deal with the complex requirements of a network
- Use *abstraction* to separate concerns
  - Identify the useful service
  - Specify the interface
  - Hide the implementation

Layering

- A result of abstraction in network design
  - A stack of services (layers)
  - Hardware service at the bottom layer
  - Higher level services are implemented by using services at lower levels



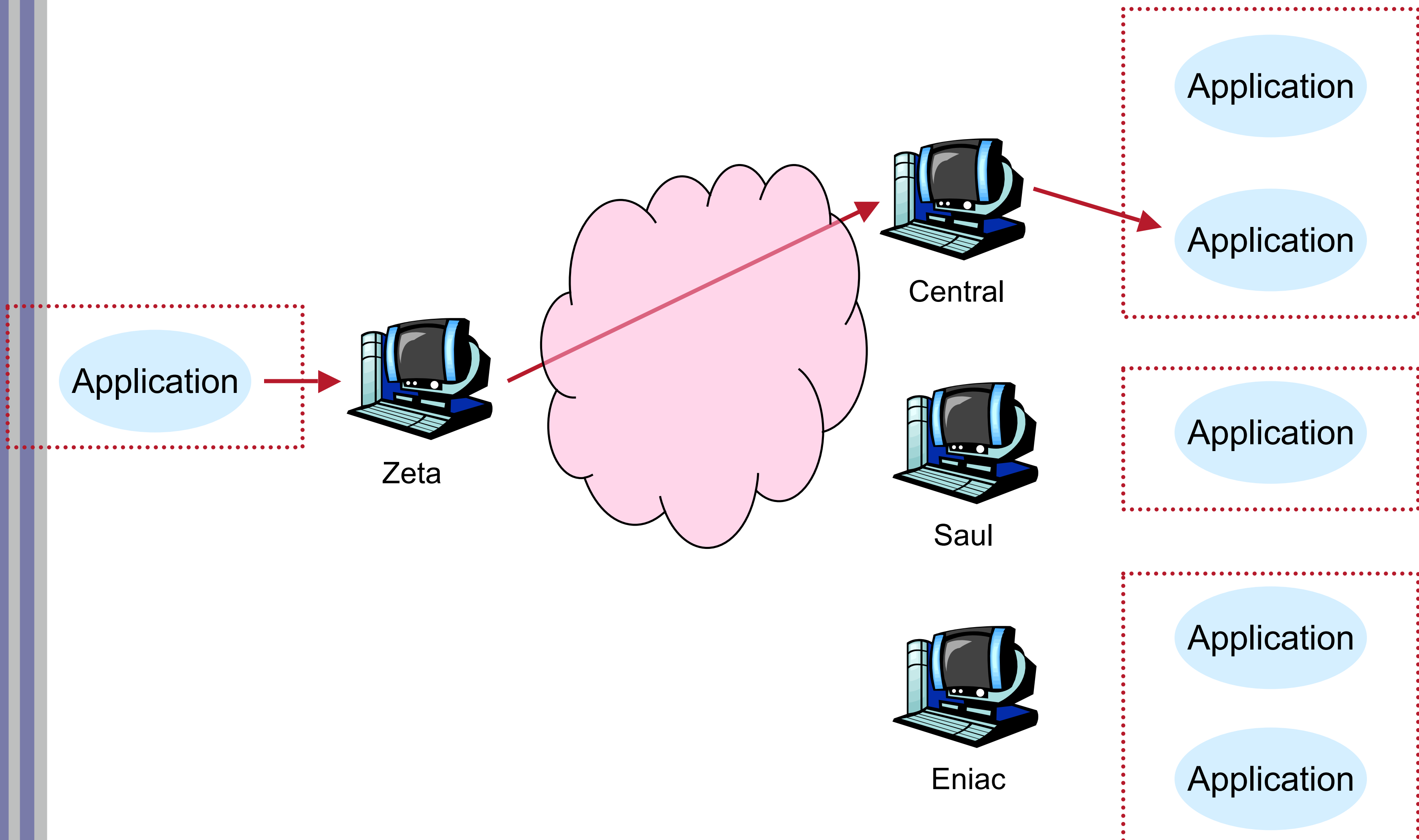
- Advantages
  - Decompose problems
  - Modular changes

Protocols

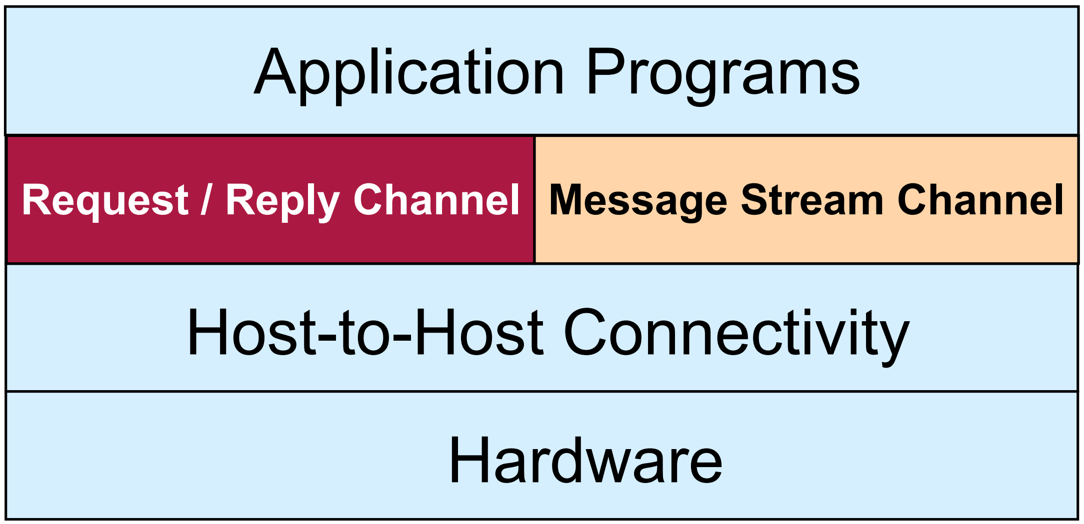
- A *protocol* is a specification of an interface between modules (often on different machines)

- Sometimes “protocol” is used to mean the implementation of the specification.

Interprocess communication

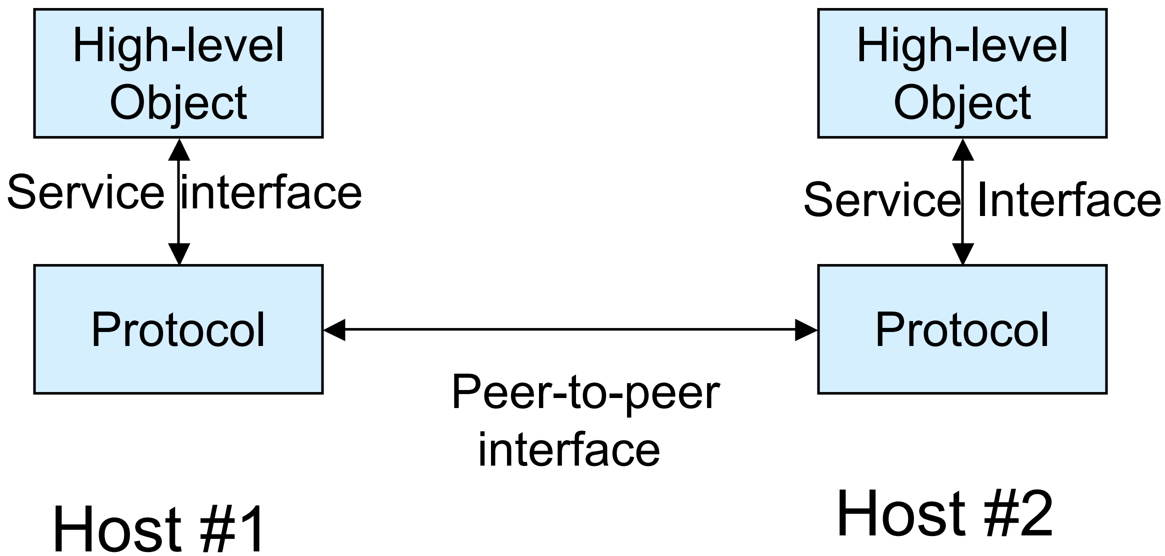


Example Protocol Stack

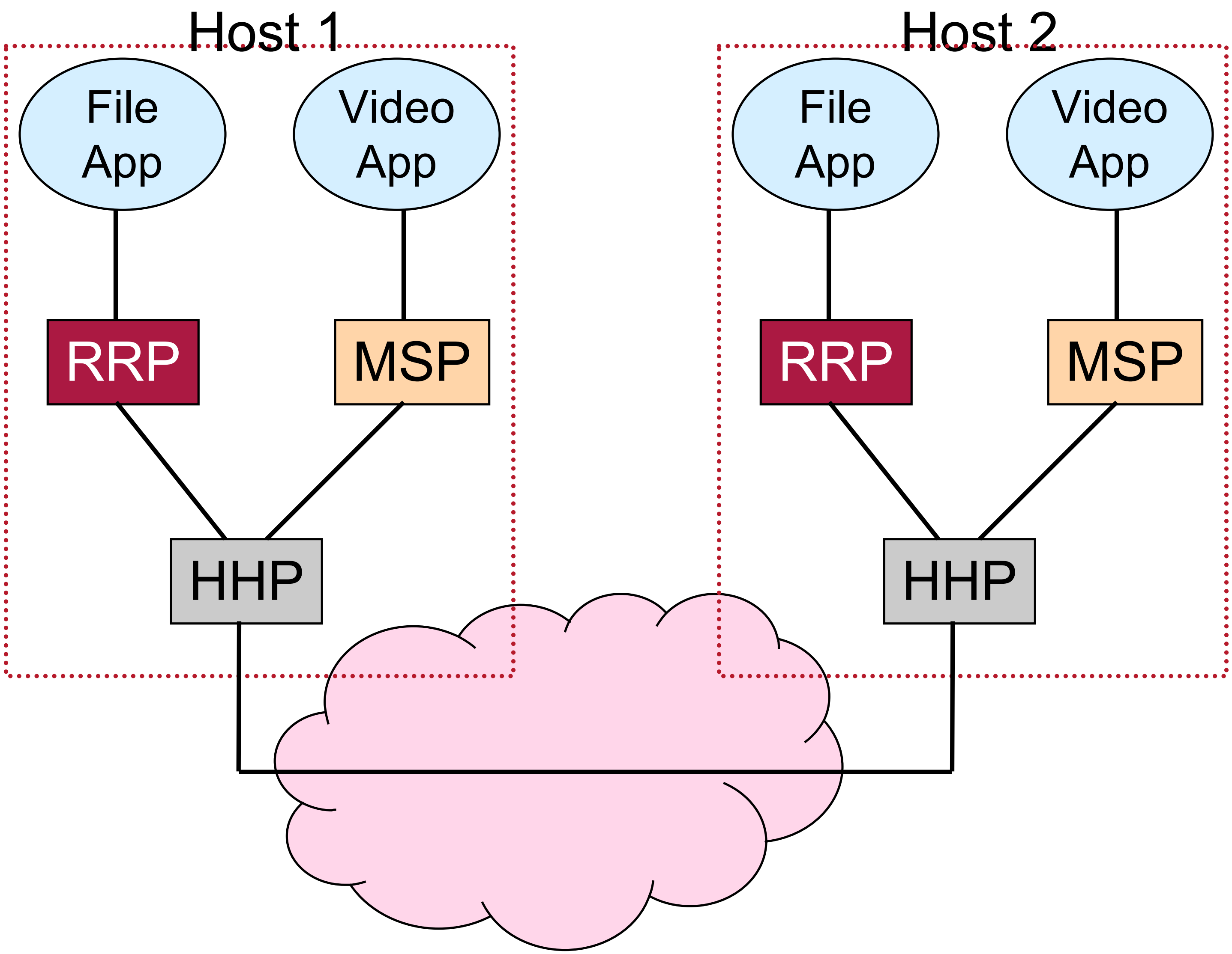


Protocol Interfaces

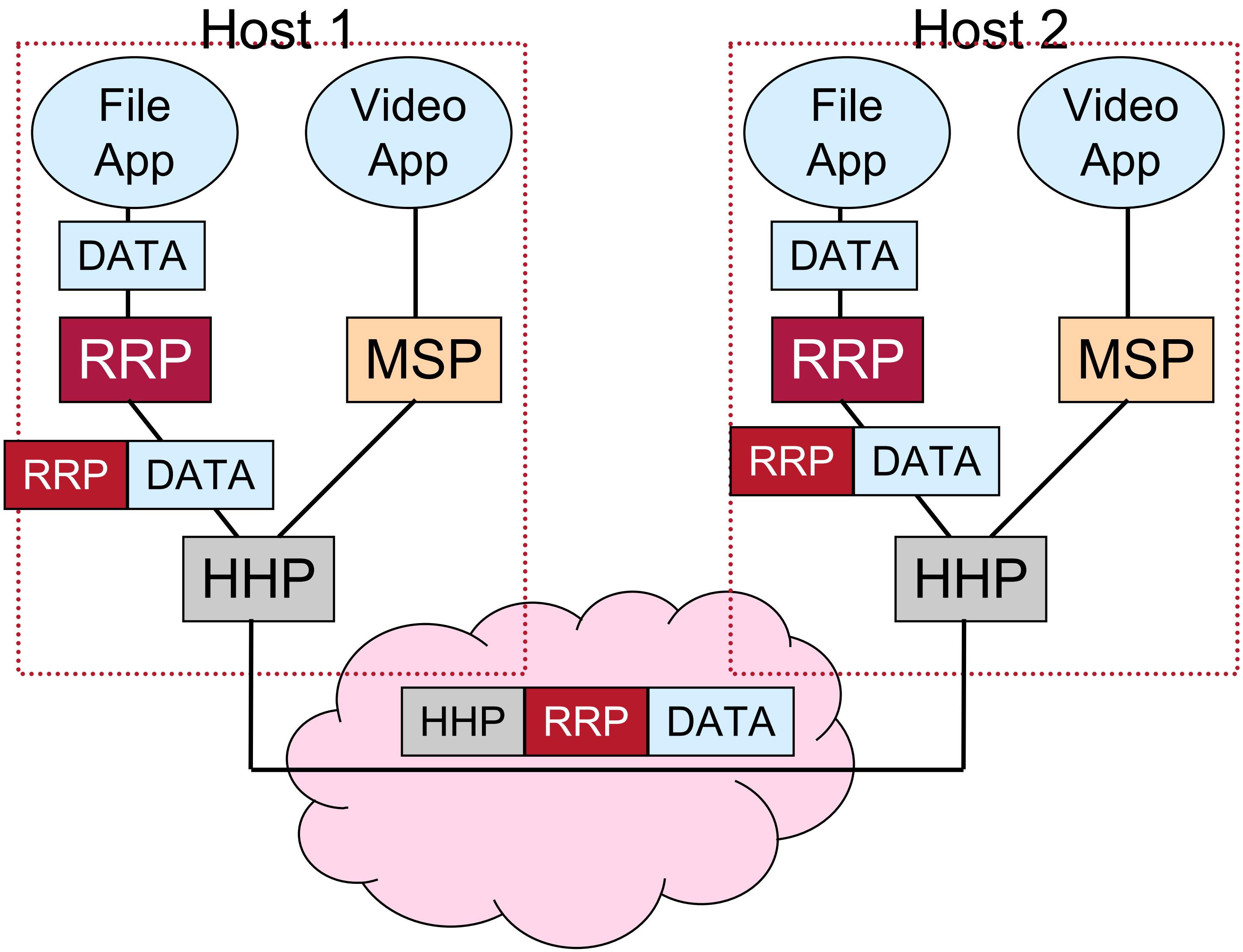
- Service Interfaces
  - Communicate up and down the stack
- Peer Interfaces
  - Communicate to counterpart on another host



Example Protocol Graph



Encapsulation

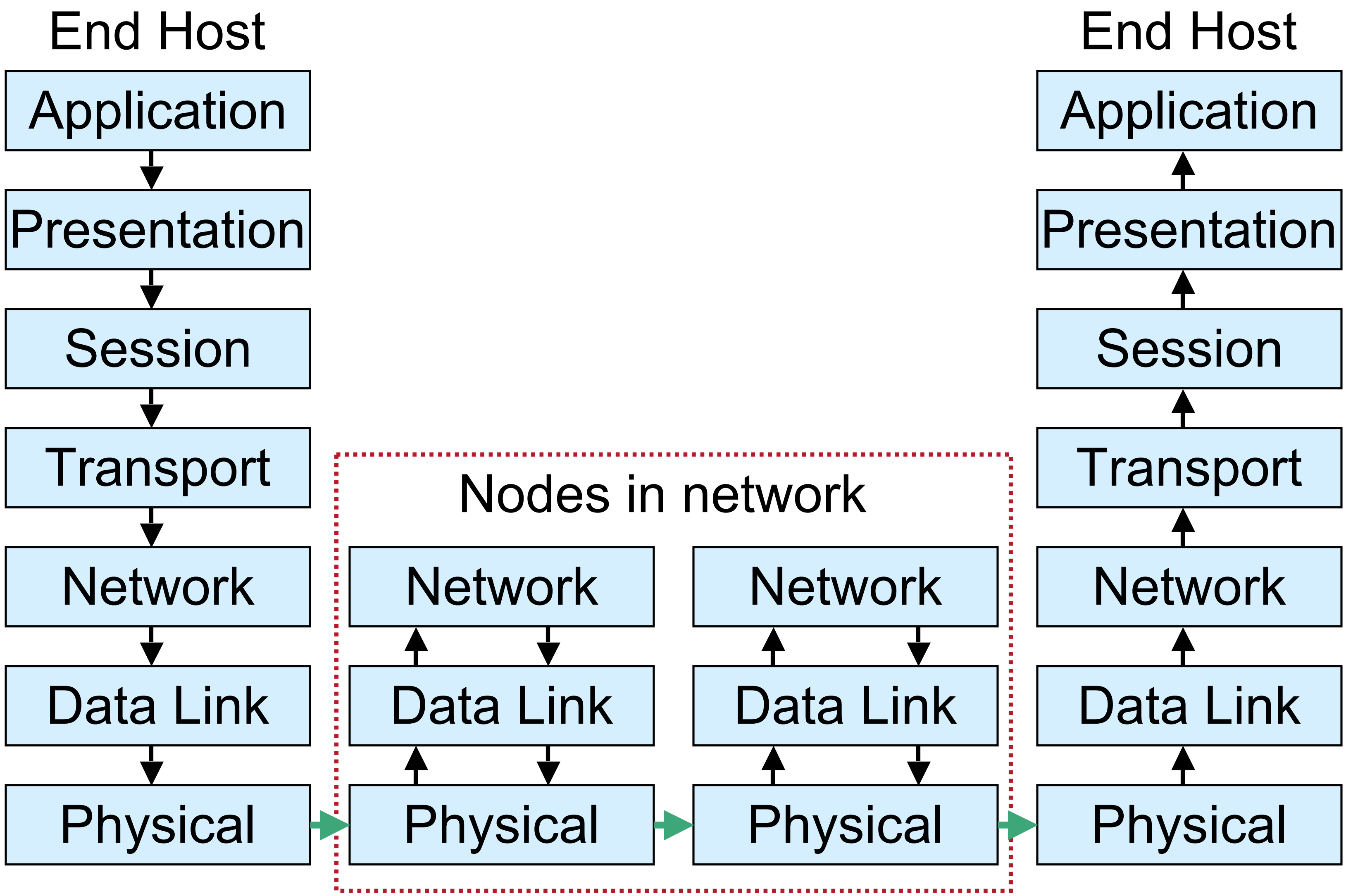


Open Systems Interconnection (OSI)

Reference model – not actual implementation.

End Host	
Application	Transmits *messages* (e.g. FTP or HTTP)
Presentation	Data format issues (e.g. big- vs. little-endian)
Session	Manages multiple streams of data
Transport	Process to process protocols
Network	Routes *packets* among nodes in network
Data Link	Packages bit streams into *frames*
Physical	Transmits raw bits over link

Open Systems Interconnection (OSI)



Internet Protocol Graph