Bandwidth, Layers Intro

13 November 2024
Lecture 2

Slides Credits: Steve Zdancewic (UPenn)

Topics for Today

- Bandwidth
- Layers

Source: Peterson and Davie 2.1 – 2.5

Some Units and Measurements

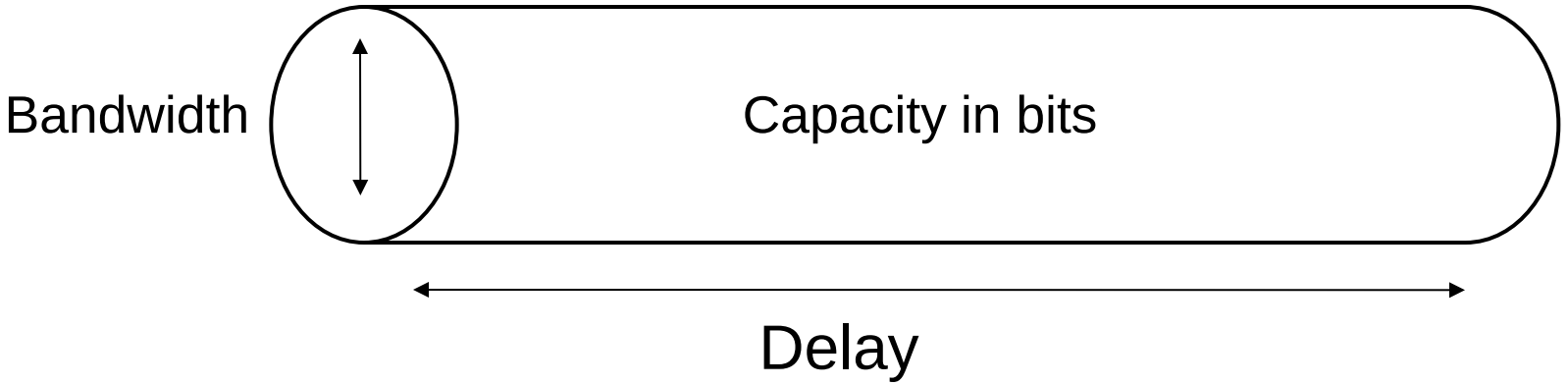
- $Mbps = 10^6\ bits/\text{sec}$
- $byte = 8\ bits$
- $KB = 2^{10}\ bytes\ (=\ 8{,}192\ bits)$
- $MB = 2^{20}\ bytes\ (=\ 8{,}388{,}608\ bits)$
- $ms = 10^{-3}\ seconds$
- $\mu s = 10^{-6}\ seconds$

- Speed of light:
  - Vacuum : $3\ x\ 10^8\ \frac{m}{\text{sec}}$
  - Copper or Fiber: $2\ x\ 10^8\ \frac{m}{\text{sec}}$

Key Equations

- `Total Sending Time = Propagation + Transmit + Queue`
- $Propagation = \frac{Distance}{SpeedOfLight}$
- $Transmit = \frac{Size}{Bandwidth}$
- $Queue = ?$

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.

Total Sending Time: Direct Link

Data moves through the link at the speed of light.

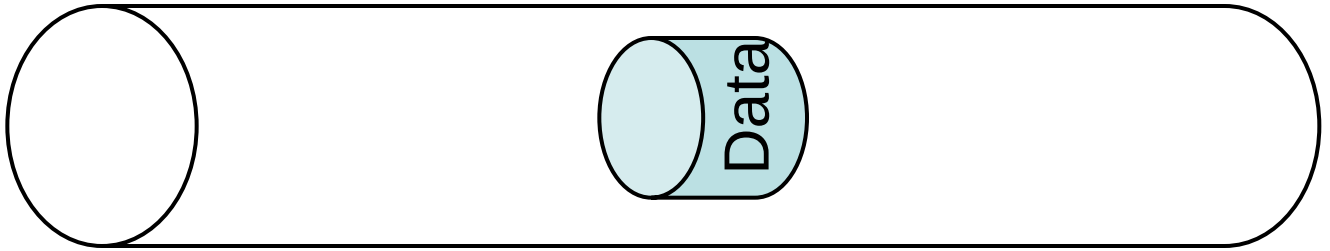
Time	
$t = 0$	Data ready to be sent

Total Sending Time: Direct Link

Data moves through the link at the speed of light.

Time	
$t = 0$	Data ready to be sent
$\mathrm{t} = \mathrm{t}ransmit$	Data finishes entering the link

Total Sending Time: Direct Link



Data moves through the link at the speed of light.

Time	
$t = 0$	Data ready to be sent
$t = transmit$	Data finishes entering the link
$t = transmit + k$	Data travels through the link ($\text{transmit} + k < propagation$)

Total Sending Time: Direct Link

Data moves through the link at the speed of light.

Time	
0	Data ready to be sent
$t = transmit$	Data finishes entering the link
$t = transmit + k$	Data travels through the link ($\text{transmit} + k < propagation$)
$t = propagation$	First bit arrives at the destination

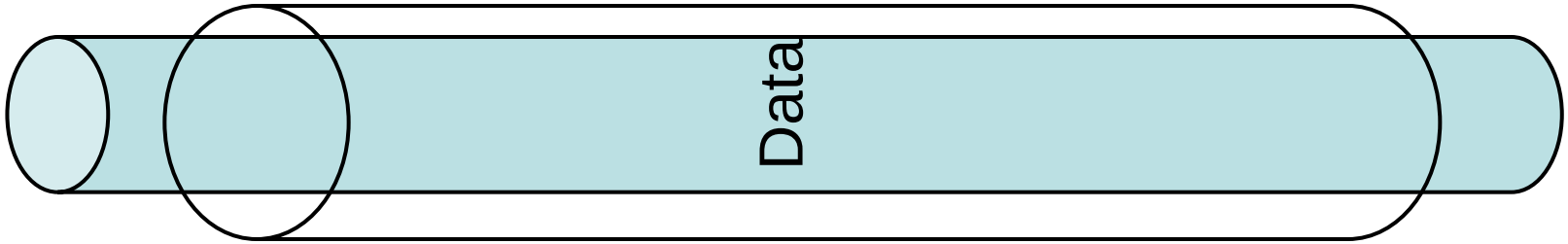
Total Sending Time: Direct Link

Data moves through the link at the speed of light.

Time	
0	Data ready to be sent
$t = transmit$	Data finishes entering the link
$t = transmit + k$	Data travels through the link ($\text{transmit} + k < propagation$)
$t = propagation$	First bit arrives at the destination
$t = transmit + propagation$	Last bit arrives at the destination

If $transmit > propagation$

First bit exits the link before the last bit finishes entering it



Data moves through the link at the speed of light.

Time	
0	Data ready to be sent
$t = transmit$	Data finishes entering the link
$t = transmit + k$	Data travels through the link ($\text{transmit} + k < propagation$)
$t = propagation$	First bit arrives at the destination
$t = transmit + propagation$	Last bit arrives at the destination

Still correct!

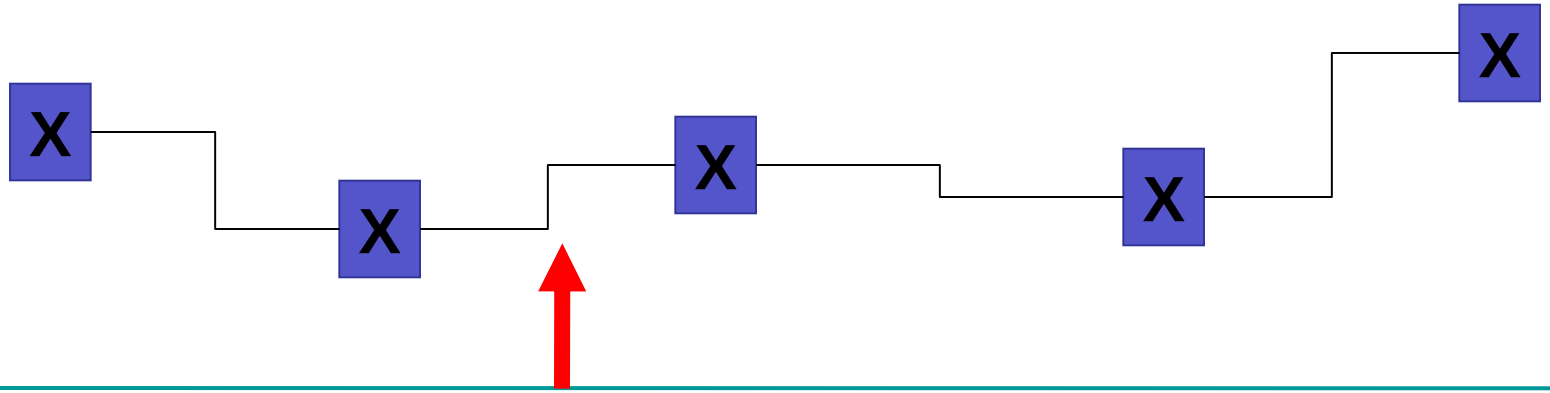
Paths Are Made of *Links*

Links are interconnected by zero or more *network elements*

- Switches, routers, hubs, bridges, etc.

Path delay is sum of link delays plus queuing (switching) delays

Path throughput = *bottleneck link* throughput



My internet is slow

What's your package?

100Mbps, 100ms ping

Two offers for you:

1. Raise you to 1 Gbps
2. Lower your ping to 10ms

Which offer is better?

Latency Bound: Send 1 Byte

Transmit

Transmit Time: 1 Byte	
100 Mbps	0.00008ms
1 Gbps	0.000008ms

Propagation

Perceived Latency	**100ms**	**10ms**	**Improvement?**
100 Mbps	100.00008ms	10.00008ms	9.9x
1 Gbps	100.0000008ms	10.0000008ms	10x
Improvement?	0.0000008x	0.000008x	

Bandwidth Bound: Send 25 MB

Transmit Time: 25MB	
100 Mbps	2.097152 s
1 Gbps	0.2097152 s

Propagation

Perceived Latency: 25MB	**100ms**	**10ms**	**Improvement?**
100 Mbps	2.147152s	2.102152s	0.021406x
1 Gbps	0.2597152s	0.2147152s	0.20957x
Improvement?	8.26733x	9.8x	

So Far

- Bandwidth
- Layers

Network Architecture

General blueprints that guide the design and implementation of networks

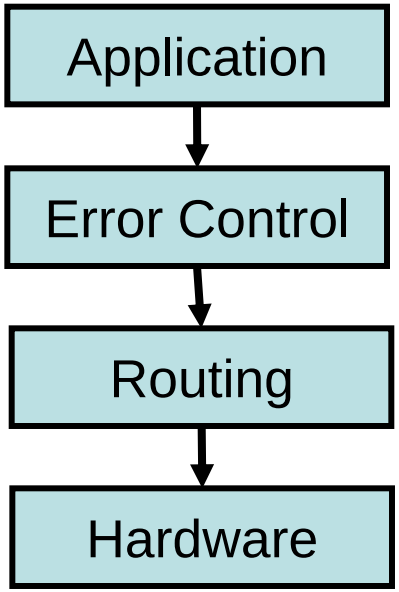
Goal: to deal with the complex requirements of a network

Use *abstraction* to separate concerns

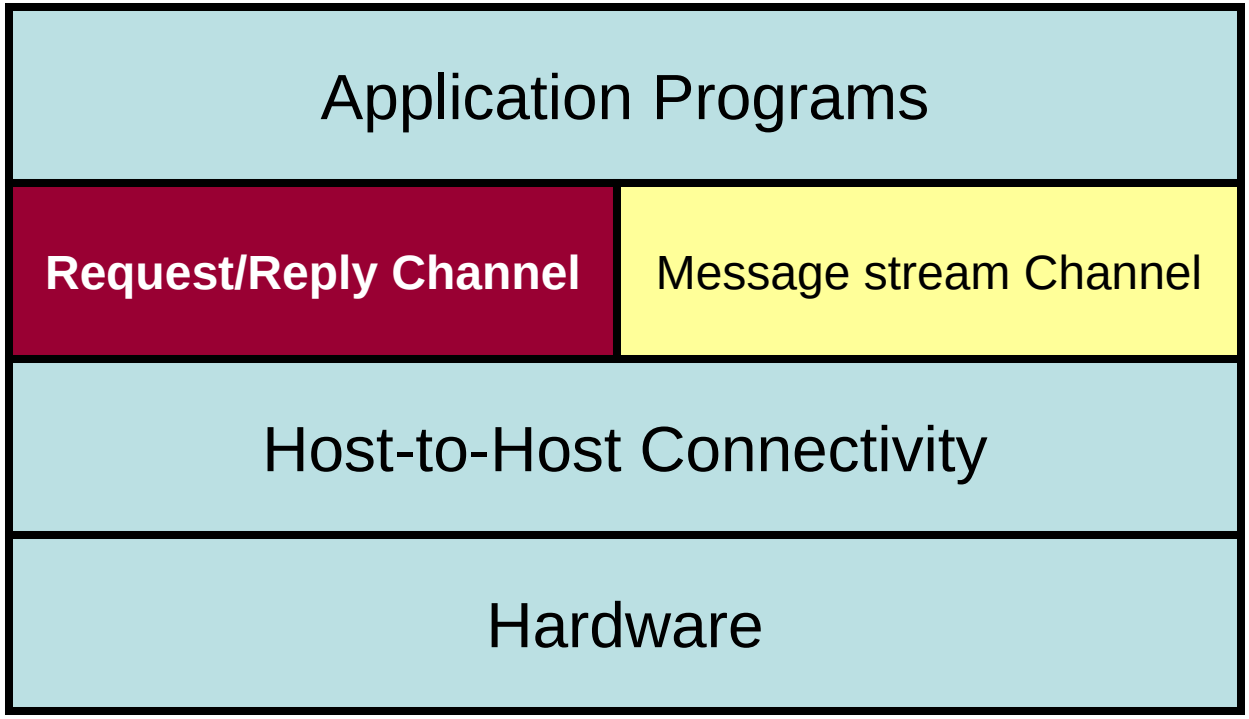
- Identify the useful service
- Specify the interface
- Hide the implementation

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** implement the layers

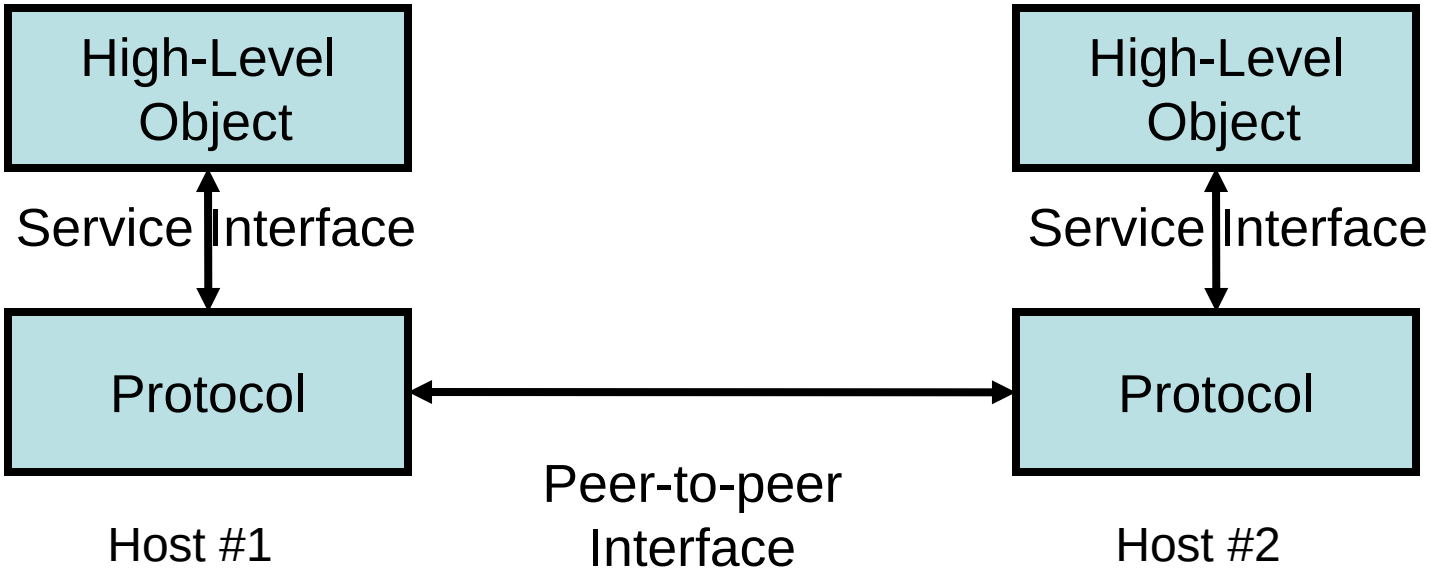


Example Protocol Stack

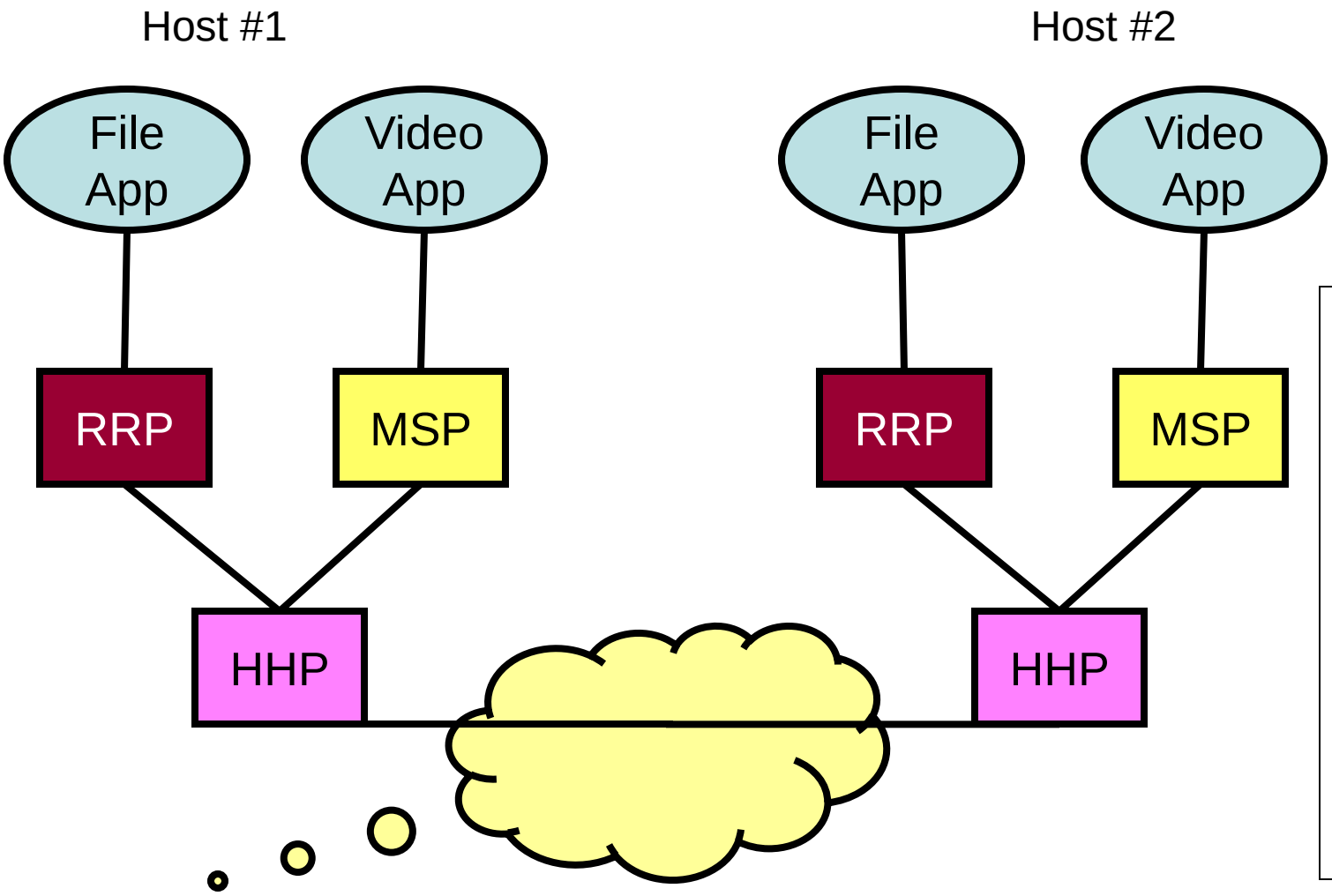


Protocol Interfaces

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



Example Protocol Graph

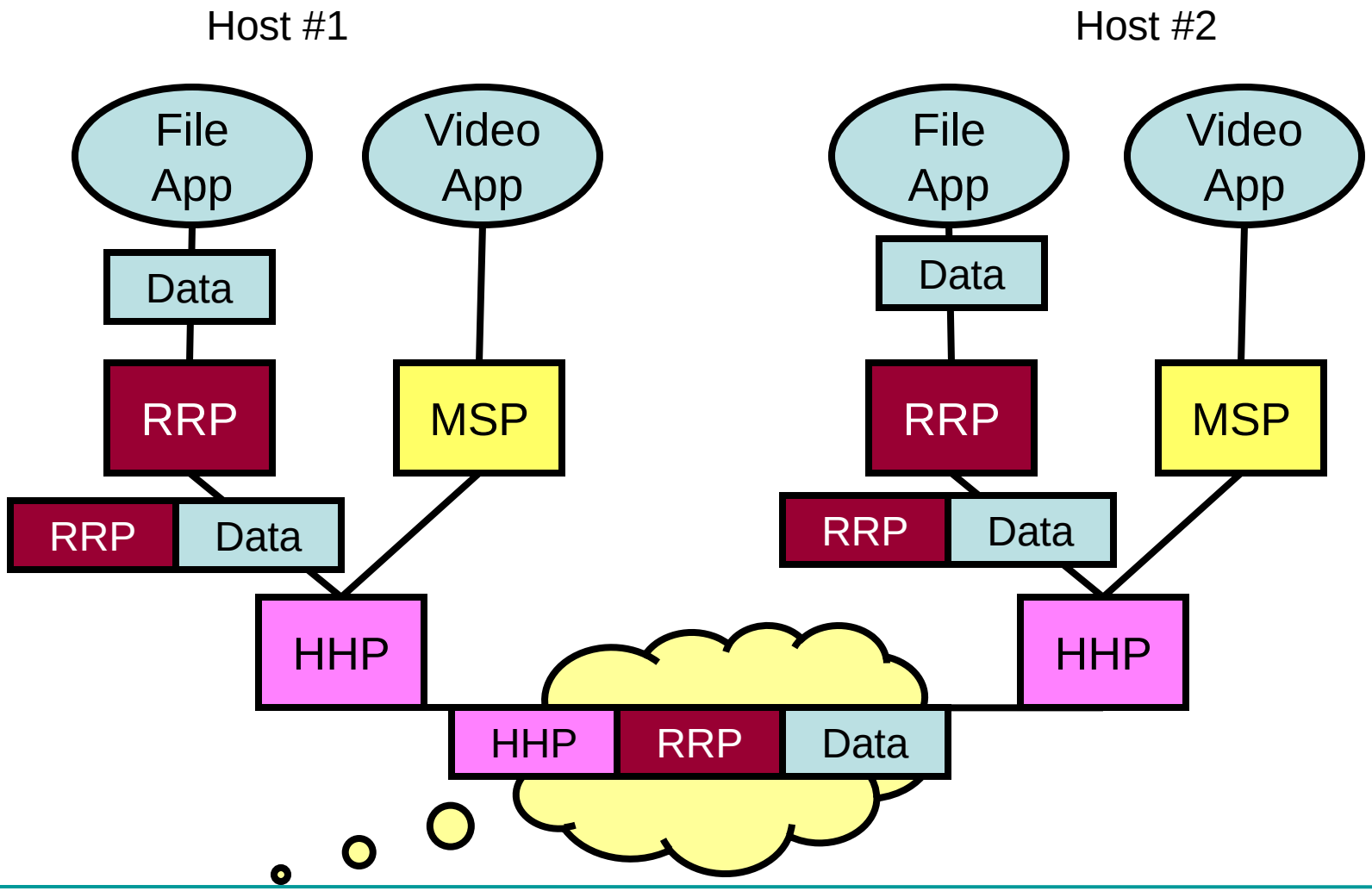


HHP: Host-to-Host Protocol

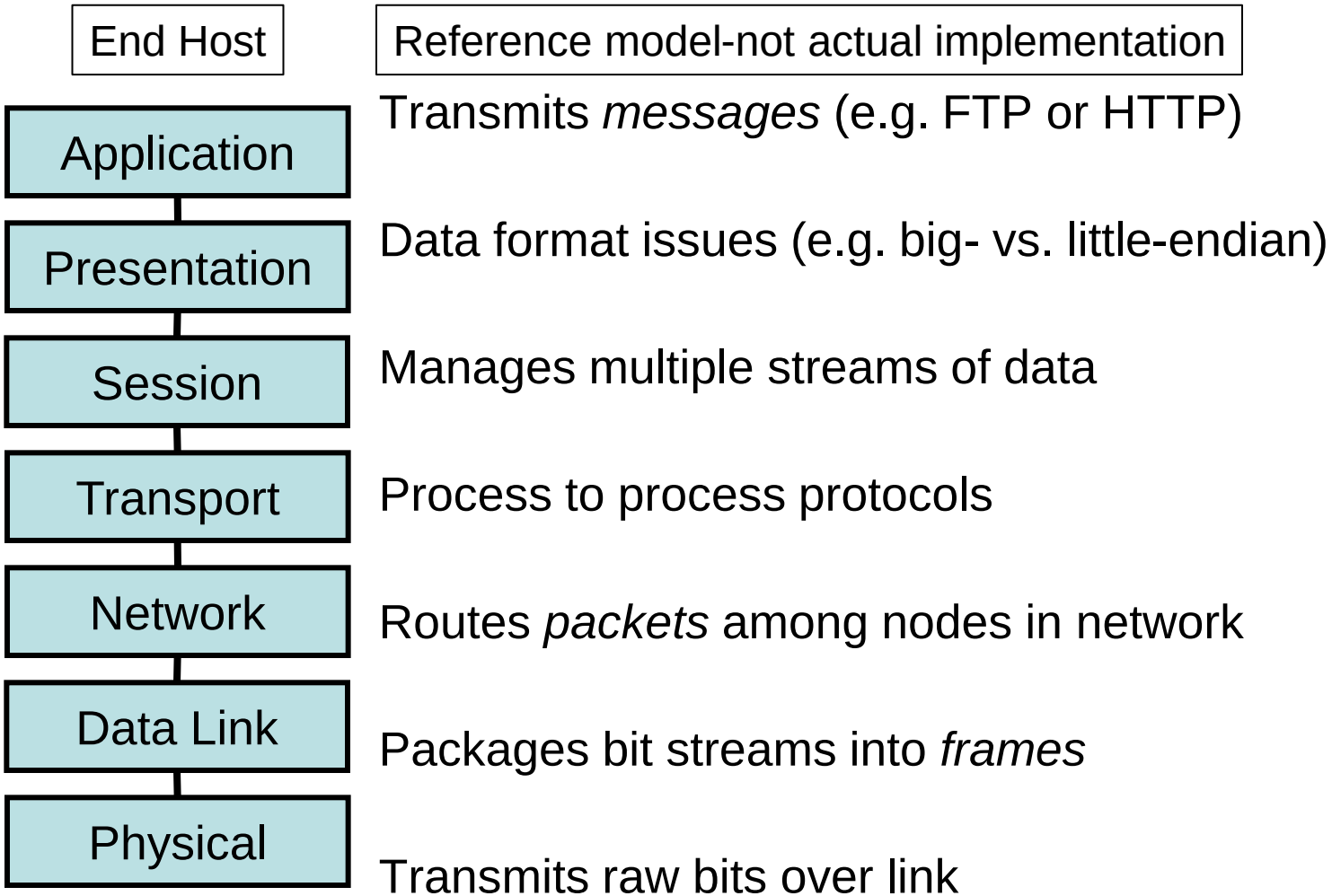
RRP: Request-Reply Protocol

MSP: Message Streaming Protocol

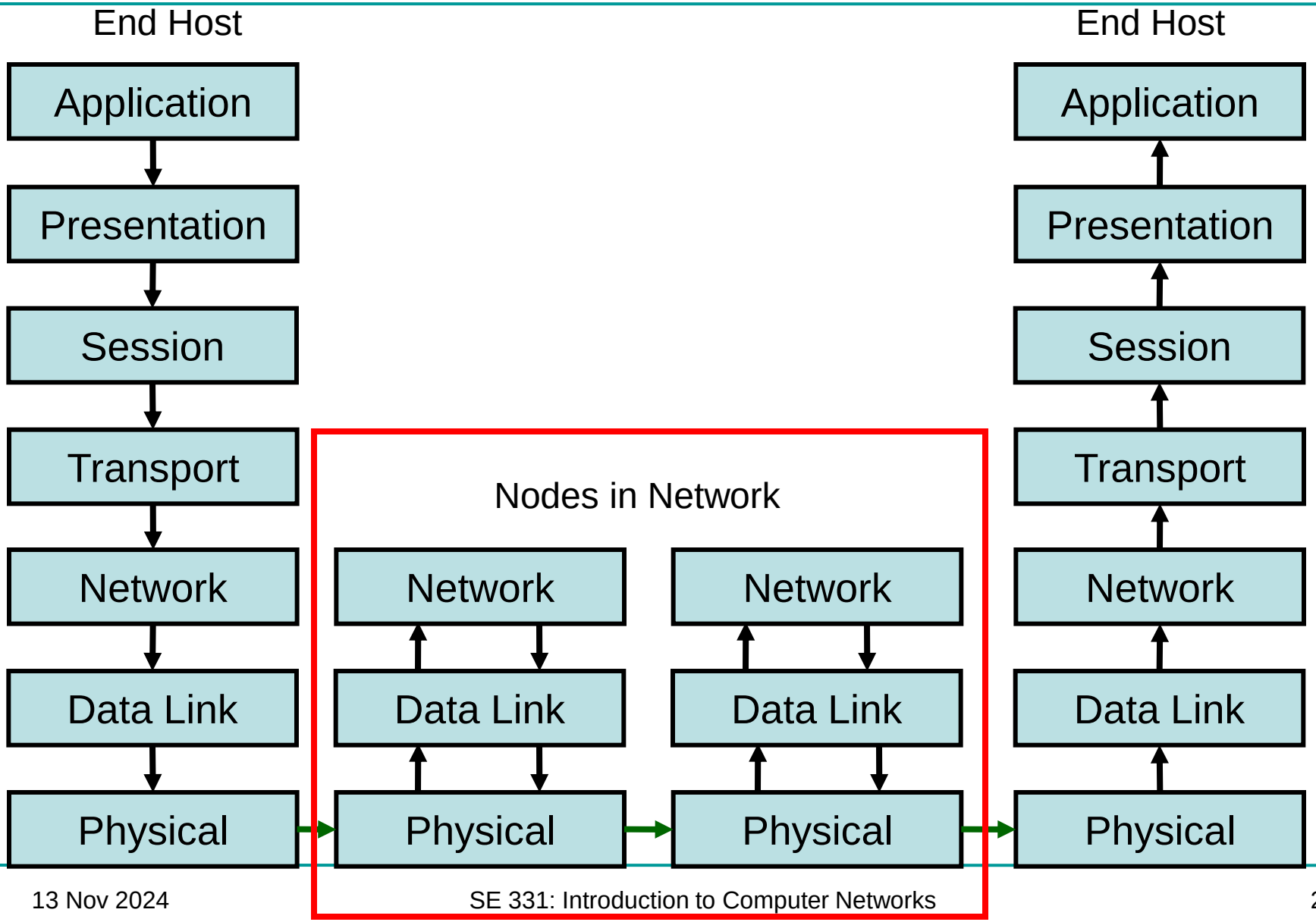
Example Protocol Graph



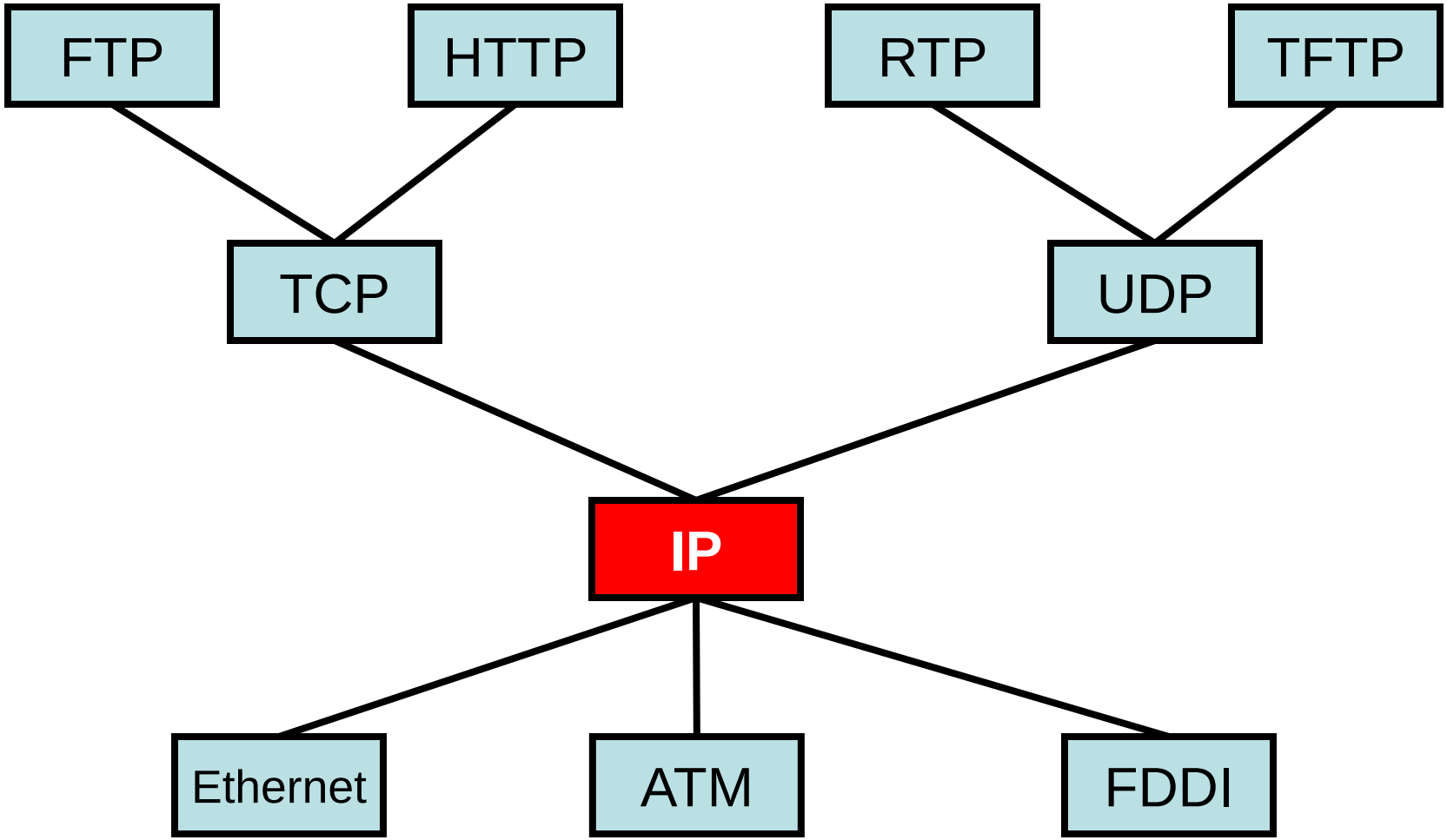
Open Systems Interconnection (OSI)



Open Systems Interconnection (OSI)



Internet Protocol Graph



Conclusion

- Bandwidth
- Layers