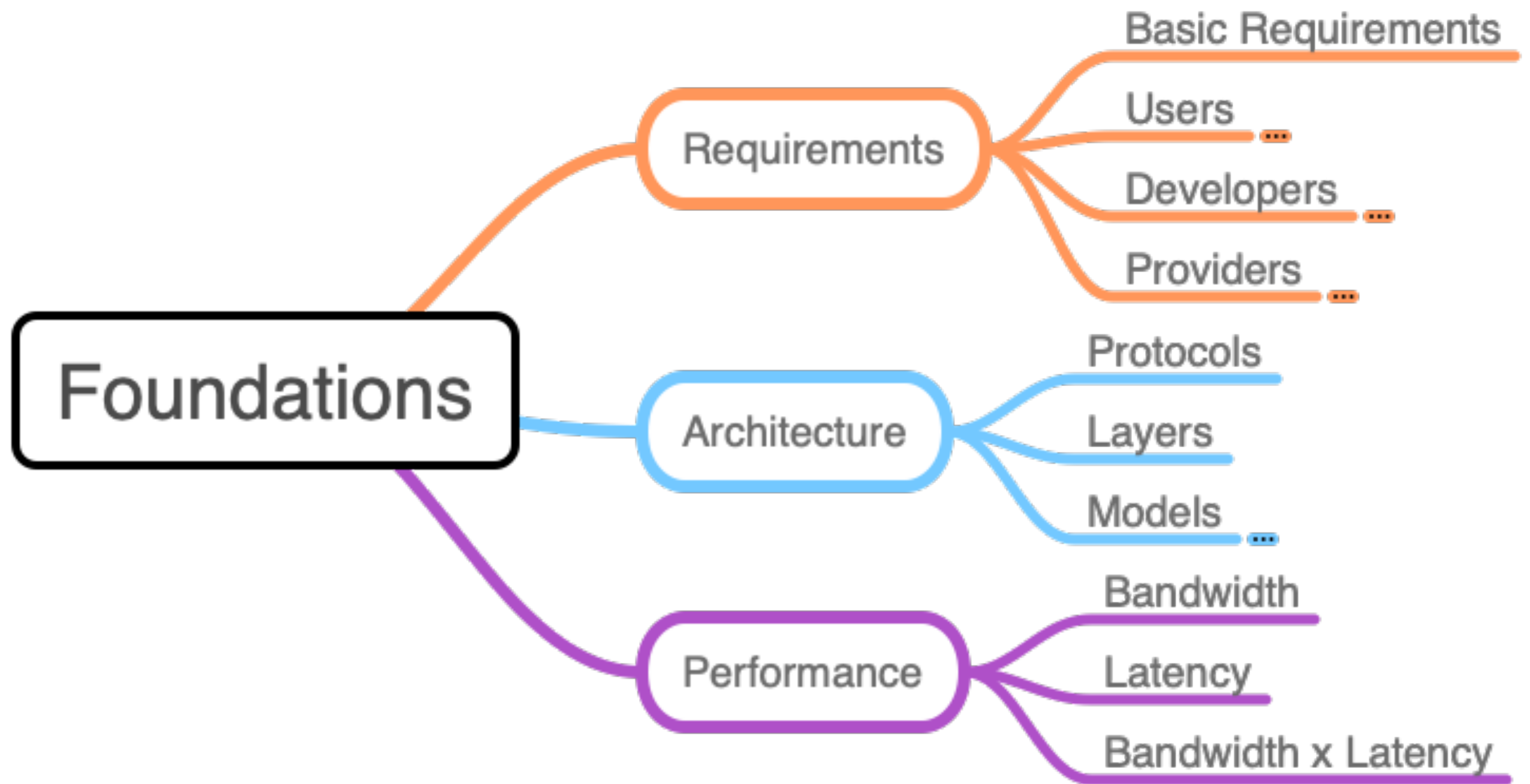


# Foundations

COS 460 - Computer Networks  
University of Southern Maine  
Fall 2019



# Goals

*Do that thing Computer Scientists do...*

1. **Reflect on and collect requirements**  
(the basis of our “network problem”)

Requirements

2. **Introduce the idea of Architecture, Layers, and Protocols**  
(apply structured problem solving)

Architecture

3. **Define performance metrics for network**  
(analyze our results)

Performance

**“How did I get here?”**

*–Talking Heads*

# Applications

## **The Web**

Based on Hypertext Transfer Protocol (HTTP)

Designed for marked up text, Hypertext Markup Language (HTML)

Includes linking, descriptive formatting, style, images, video, ...

Incredibly flexible, often used by higher-level applications

# Applications

The Web

The first *killer app* for the Internet

**Email**

- Simple Mail Transport Protocol (SMTP)
- Post Office Protocol (POP)
- Internet Message Access Protocol (IMAP)

Now more common to access email via web (HTTP).

# Applications

The Web

Even before *the cloud* there was network storage

Email

**File Storage**

- Network File System (NFS)
- Common Internet File System (CIFS)
- IP Small Computer System Interconnect (iSCSI)

...and of course, over HTTP these days

# Applications

The Web

Often hidden behind web browser or media player

Email

Motion Picture Experts Group (MPEG) is most common

File Storage

**Multimedia**

- Audio - Layers 1, 2, 3 (mp3)
- Images - JPEG
- Video - MPEG-2, ..., MPEG-4



# Applications

The Web

There are too many other applications to enumerate

Email

- Social Media ([facebook](#), [twitter](#), [instagram](#))

File Storage

- News ([reddit](#), [feedly](#))

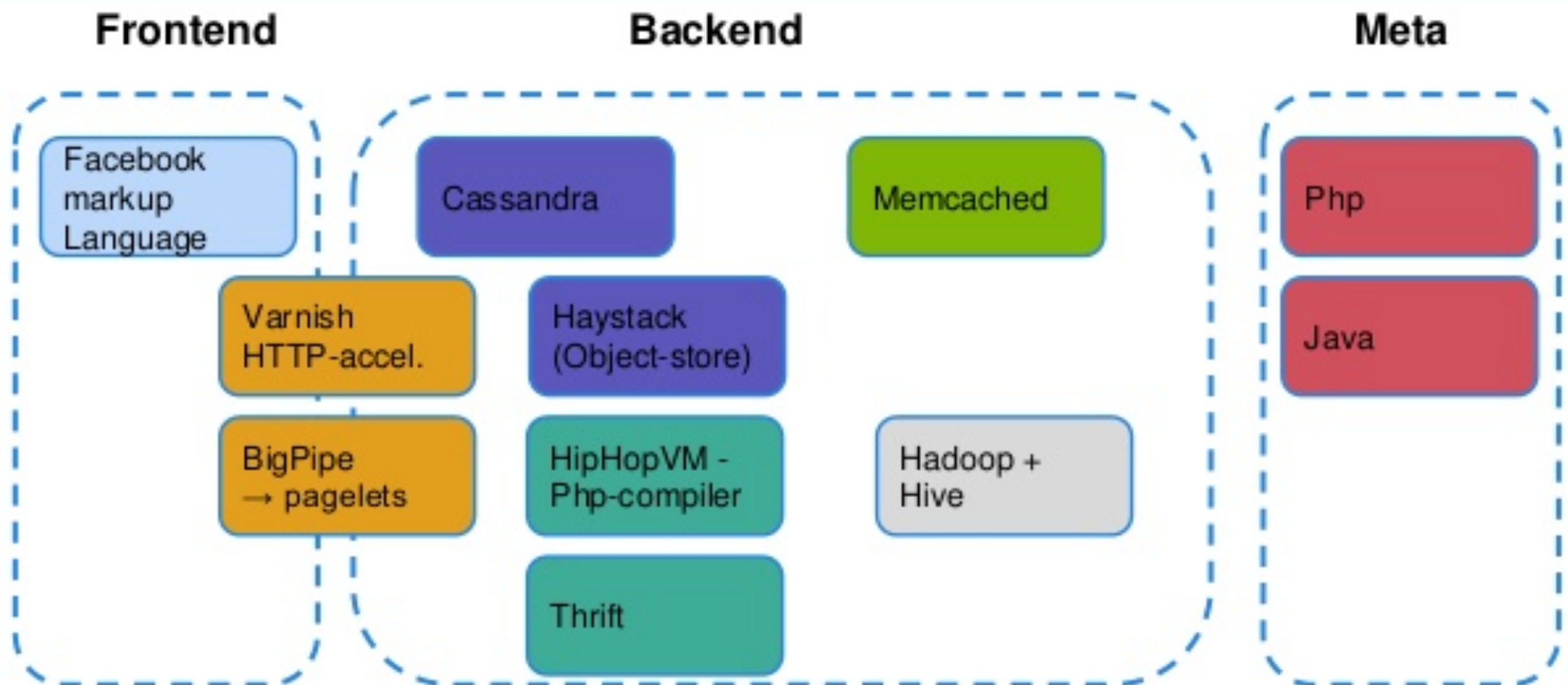
Multimedia

- Document and Spreadsheet editing ([Google](#), [Microsoft](#), ...)

More...

- Slide presentations ([Google](#), [Microsoft](#), ...)

# Facebook



# Tech Stacks

In groups of 2-3

Drill down into the “tech stack” of a web application

What technologies do they use and what for?

Which are ‘network’ technologies?

**5:00**

# Tech Stacks

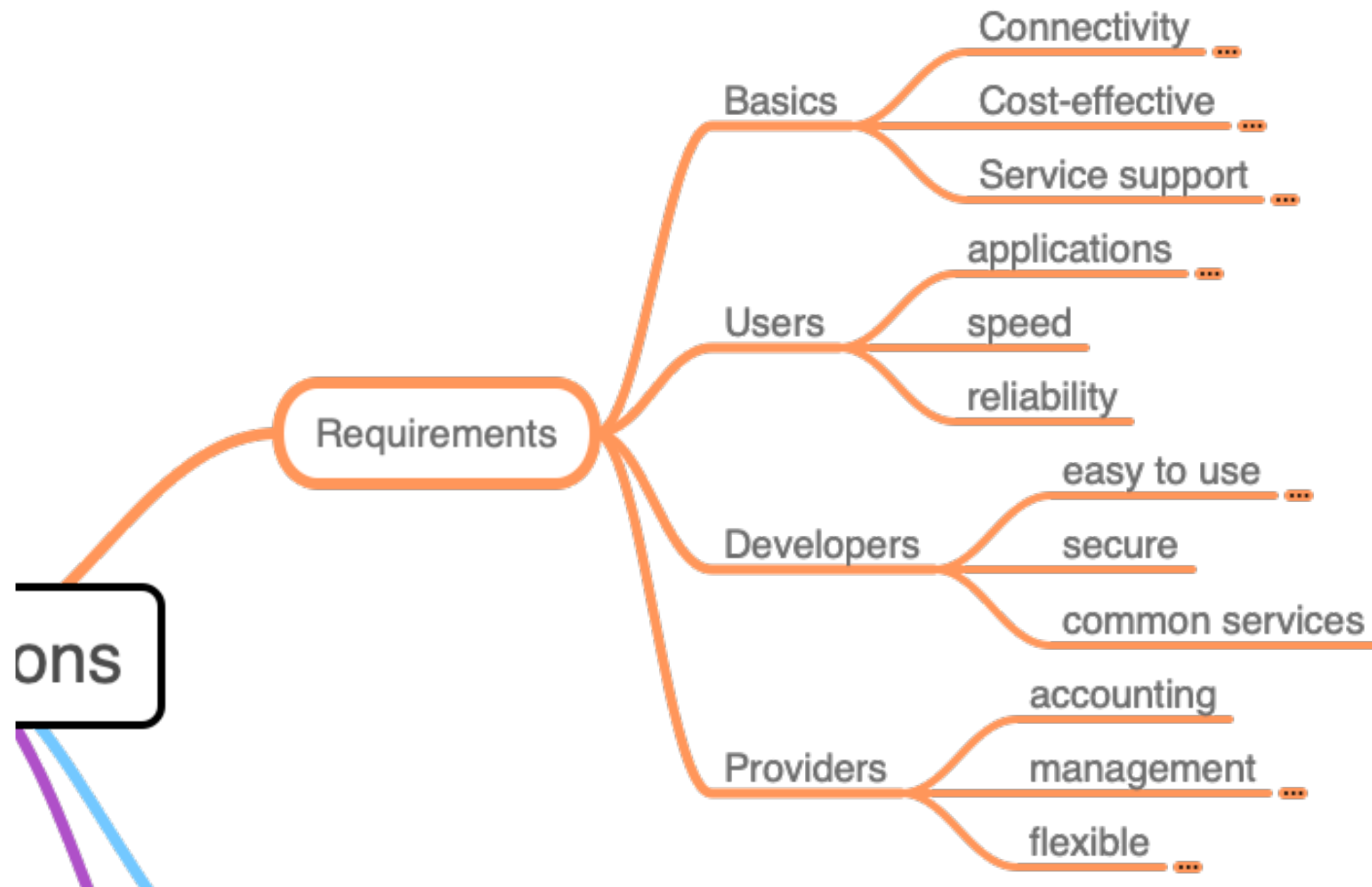
In groups of 2-3

Drill down into the “tech stack” of a web application

What technologies do they use and what for?

Which are ‘network’ technologies?

**5:00**

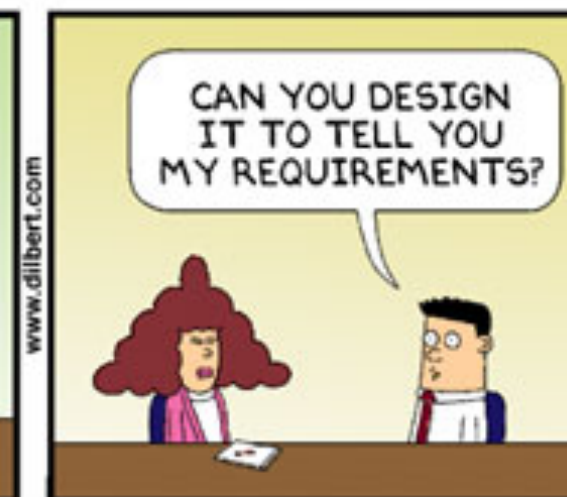


# Requirements



# DILBERT<sup>®</sup>

BY  
SCOTT ADAMS





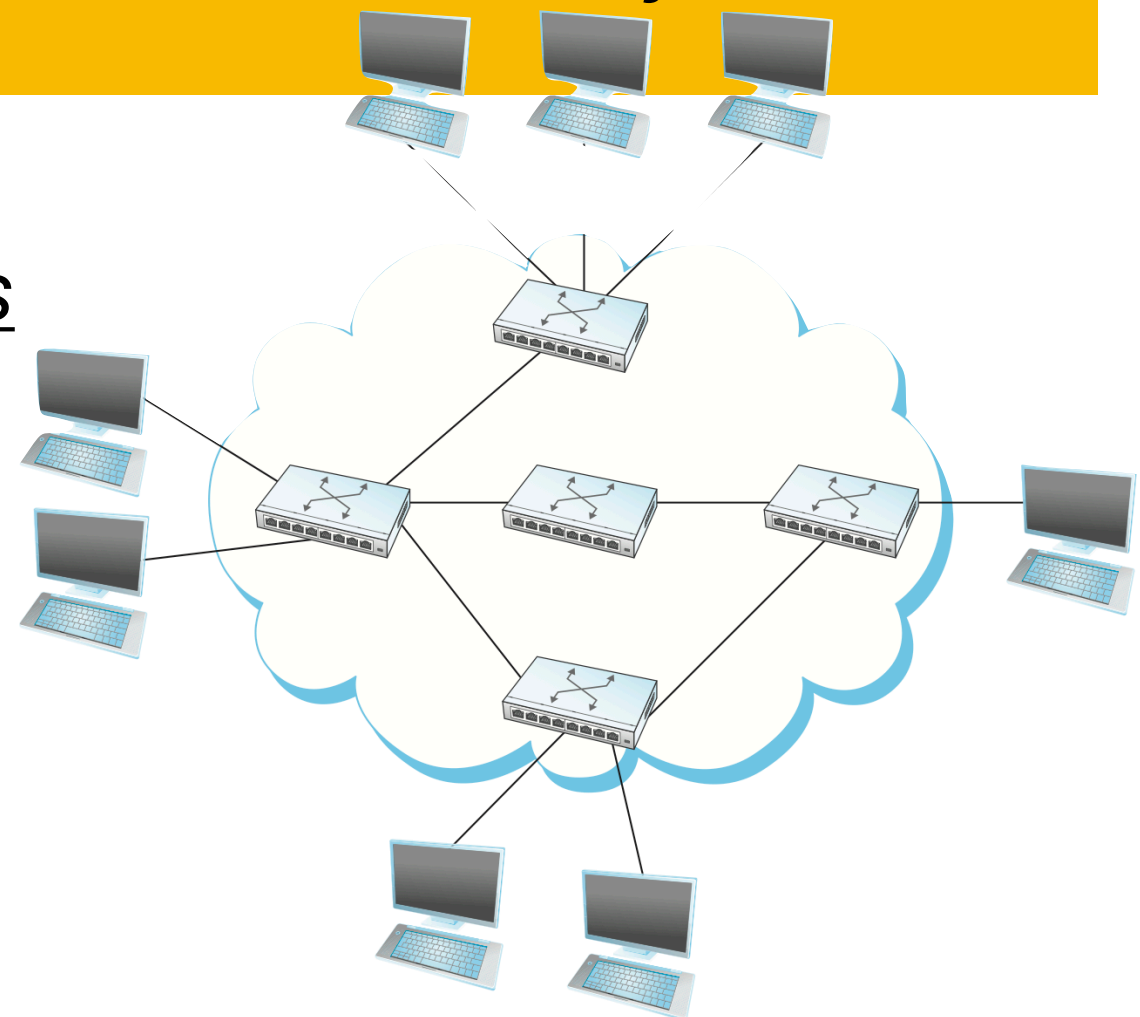
# Requirements

Scalable  
Connectivity

## Definition of Network

*“Two or more nodes connected by a physical link,  
or two or more networks connected by a node.”*

- Nodes and Links  
Hosts and Switches
- Data in packets
- Packet Switched  
Store and Forward

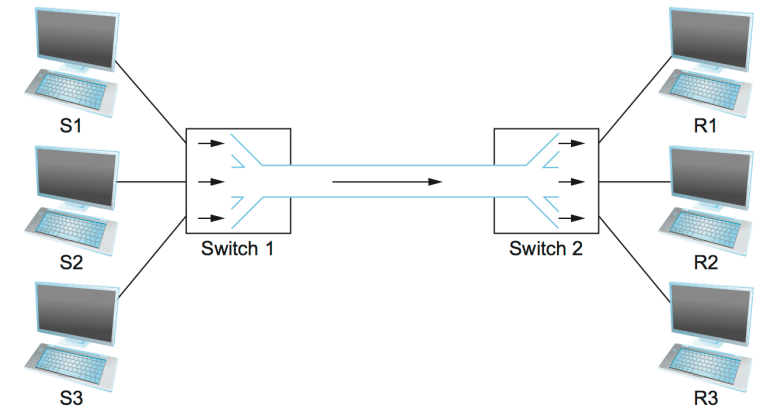


# Requirements

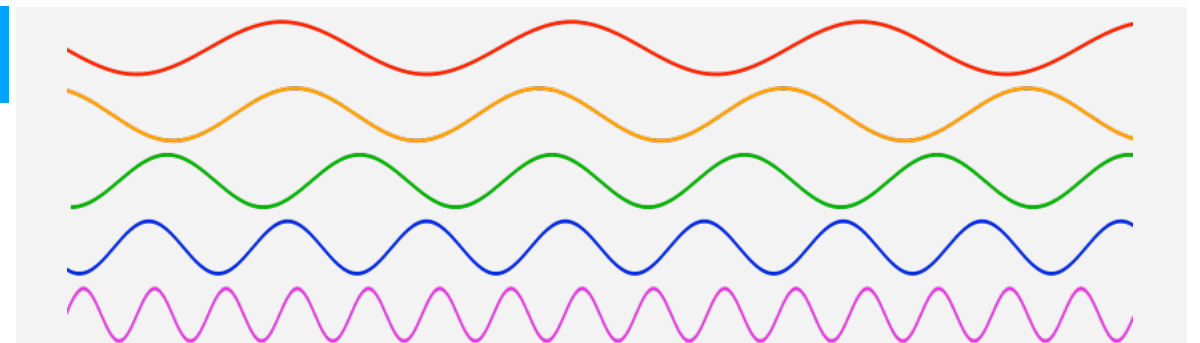
Scalable  
Connectivity

Cost  
Effective

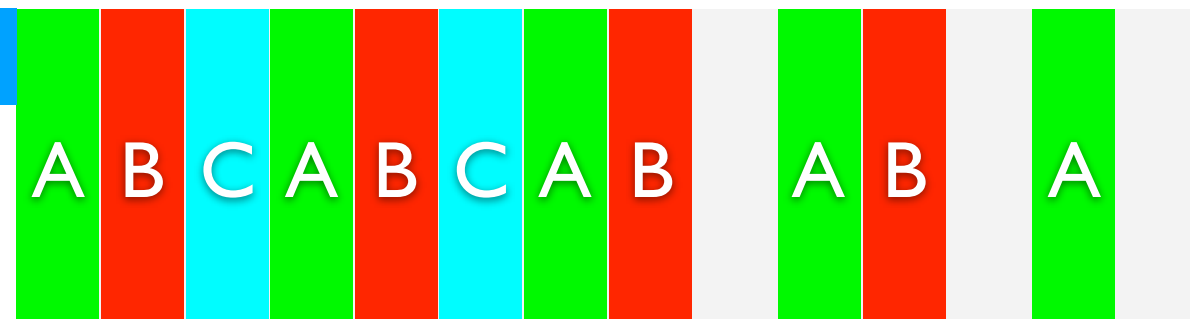
## Multiplexing data



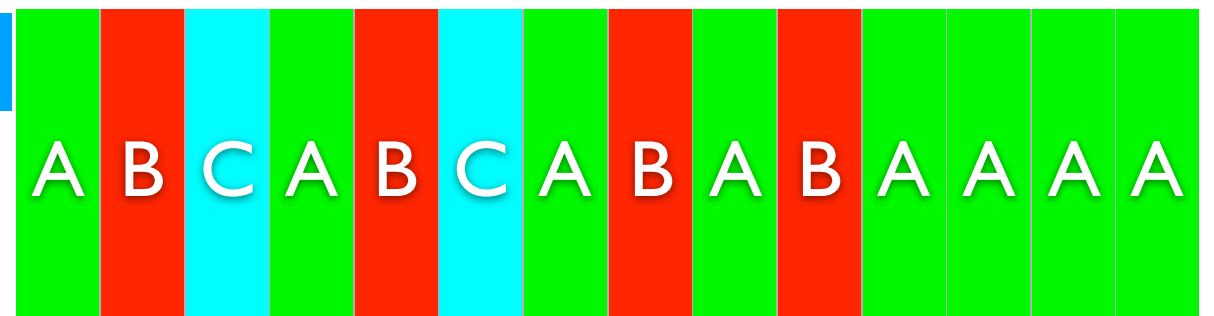
Frequency



Time Division



Packet-Switched



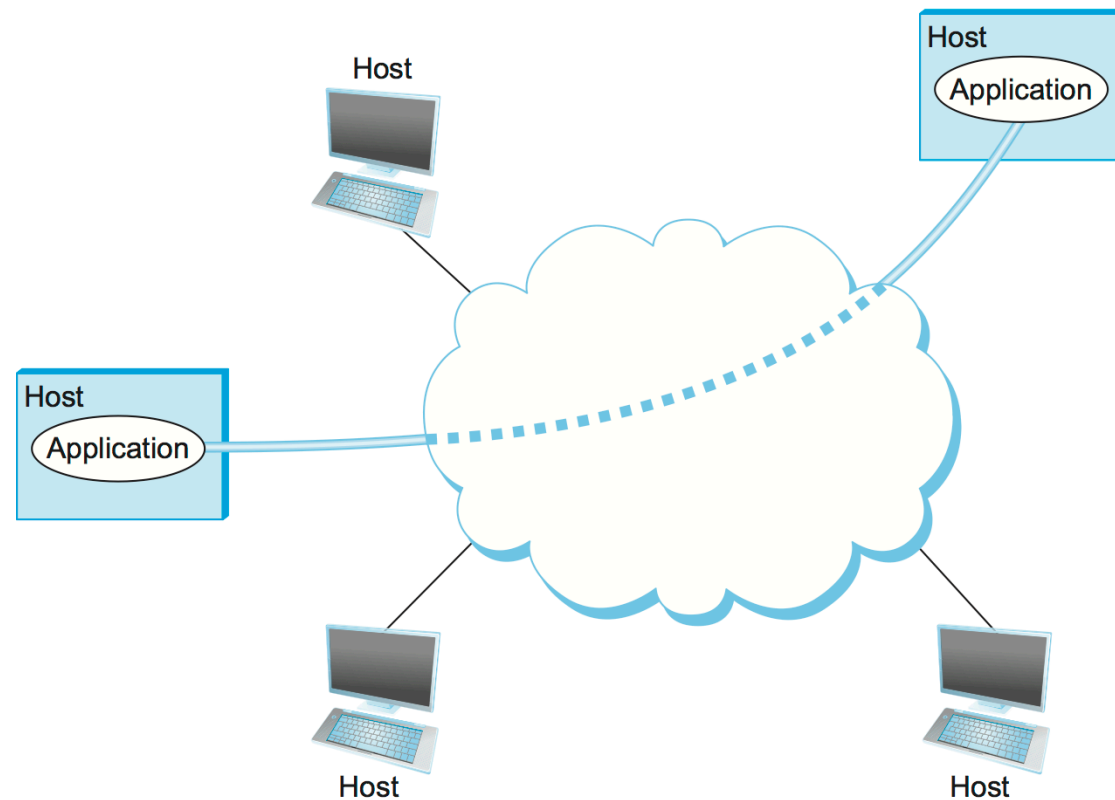


# Requirements

Scalable  
Connectivity

Cost  
Effective

Support  
Common  
Services



Define useful channels that understands the application needs and the networks ability.

file transfer — streaming — web browsing

# Requirements

Scalable  
Connectivity

Cost  
Effective

Support  
Common  
Services

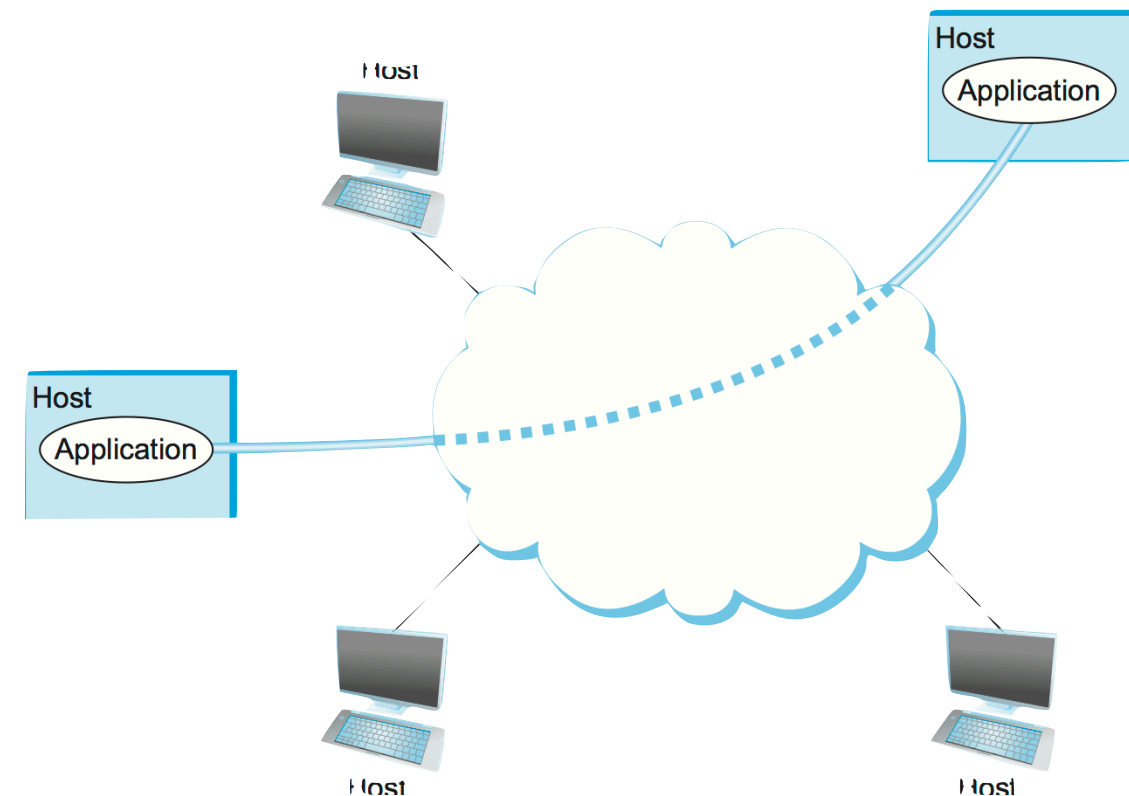
**Reliable &  
Manageable**

## Reliable

Fill in the gap between what the application expects and what the underlying technology can provide.

## Manageable

Upgrades, billing, support new applications



# Requirements

Take a couple of minutes to think about and write down

Who's requirements are we missing?

What are they?

**2:30**

# Requirements

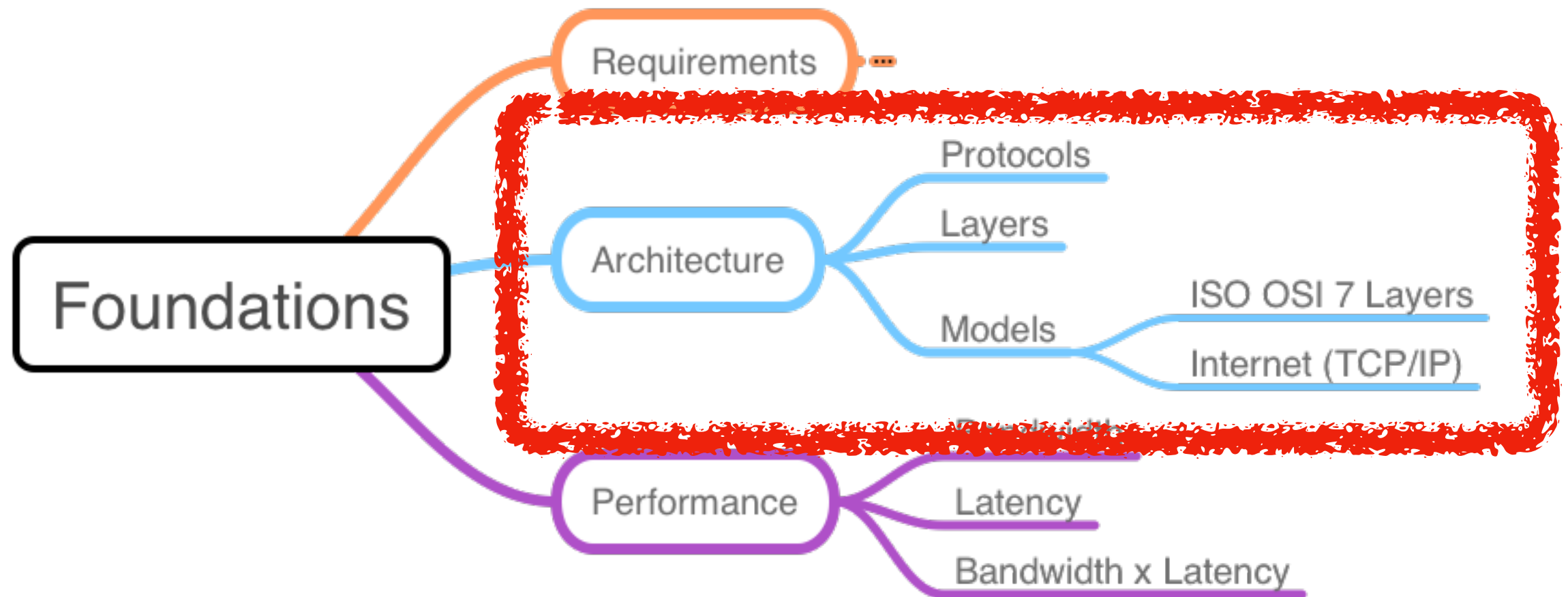
Take a couple of minutes to think about and write down

Who's requirements are we missing?

What are they?

**2:30**

# Architecture



# Architecture

## Layers & Protocols

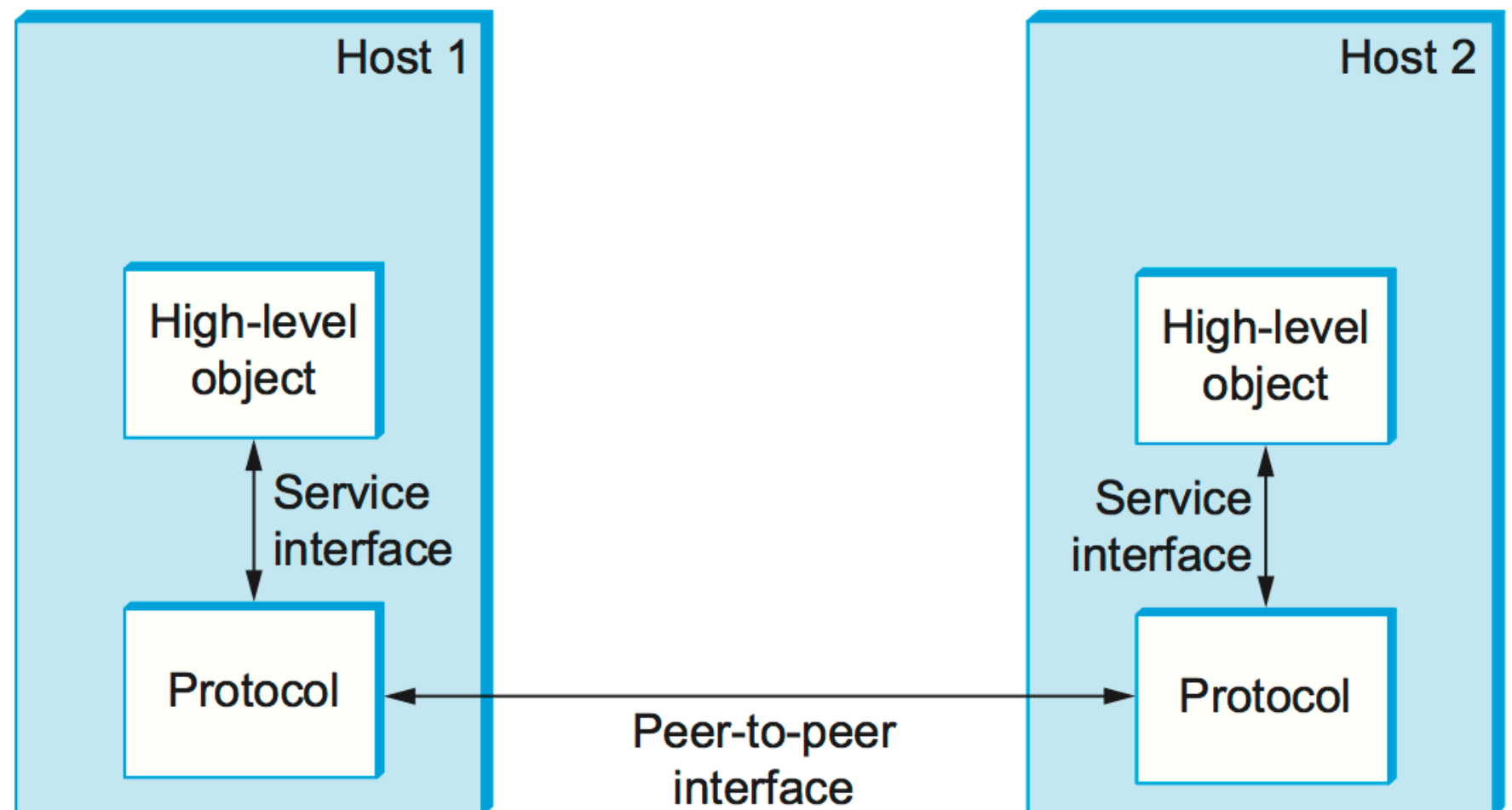
Application programs
Process-to-process channels
Host-to-host connectivity
Hardware

**Simplified Layers**

# Architecture

## Layers & Protocols

### Service and Peer Interfaces (Protocols)

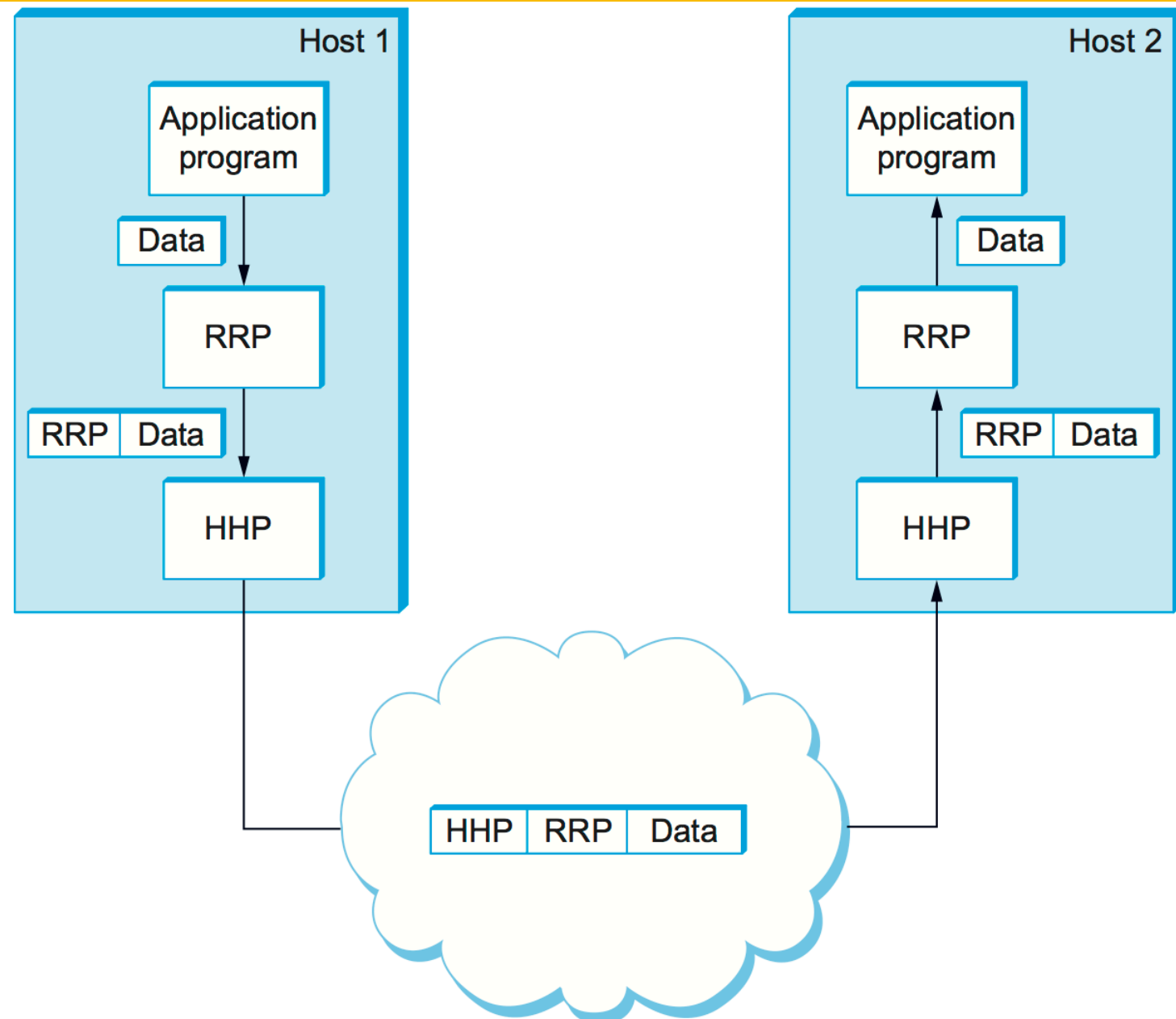


# Architecture

Layers &  
Protocols

Encapsulation

High-level encapsulated in low-level



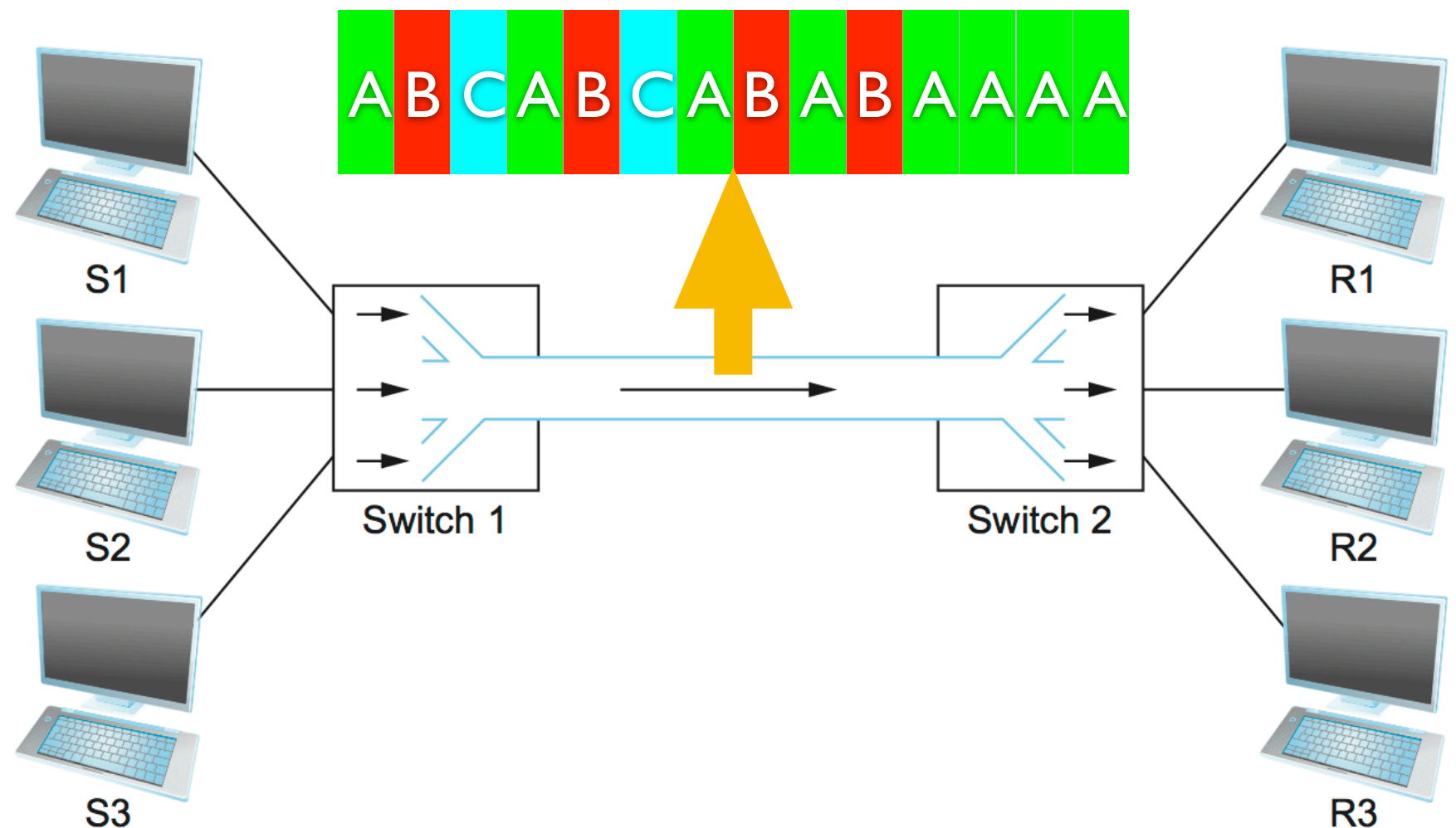


# Architecture

Layers &  
Protocols

Encapsulation

**Mux/Demux**



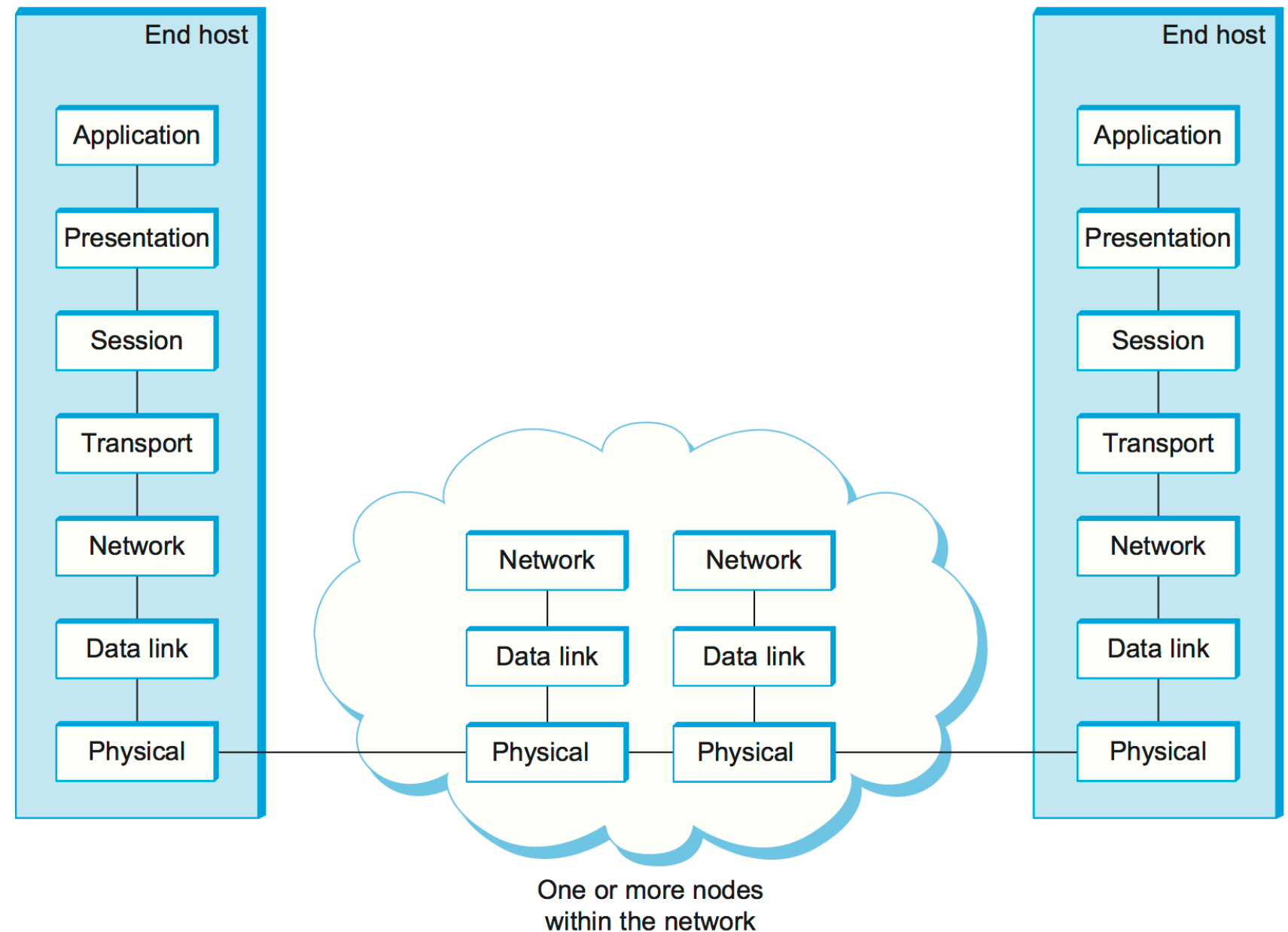
# Architecture

Layers &  
Protocols

Encapsulation

Mux/Demux

**OSI 7-Layer**



# Architecture

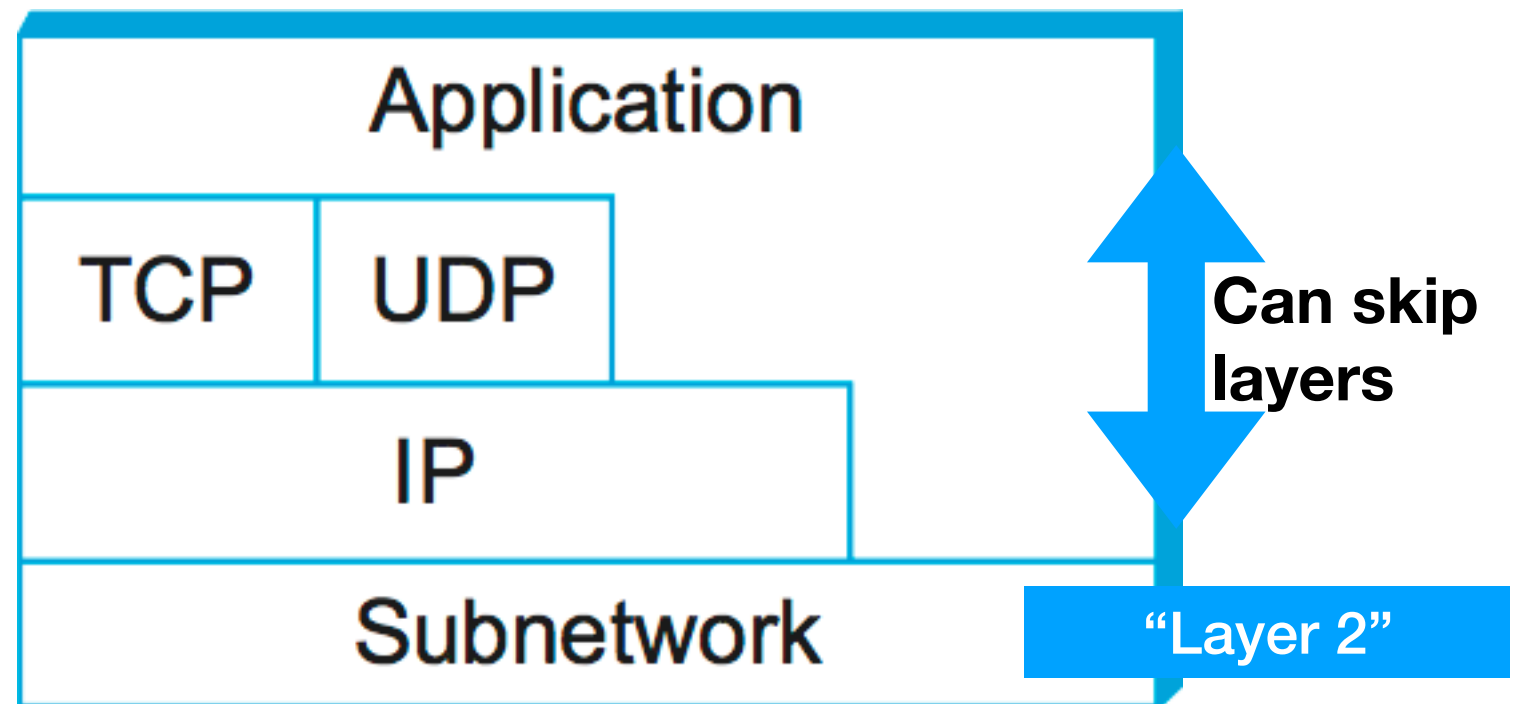
Layers &  
Protocols

Encapsulation

Mux/Demux

OSI 7-Layer

Internet



# Architecture

Layers &  
Protocols

Encapsulation

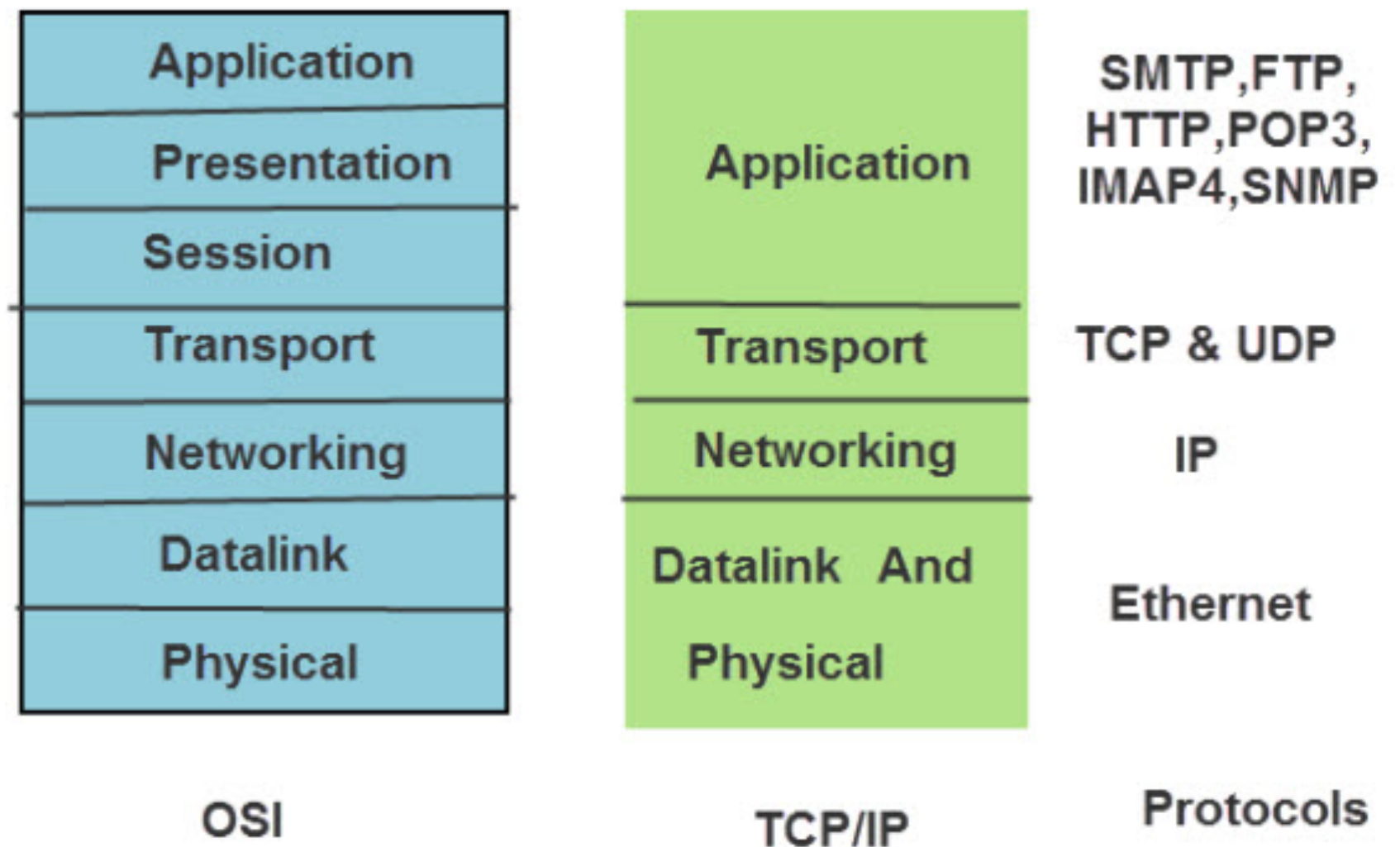
Mux/Demux

OSI 7-Layer

Internet

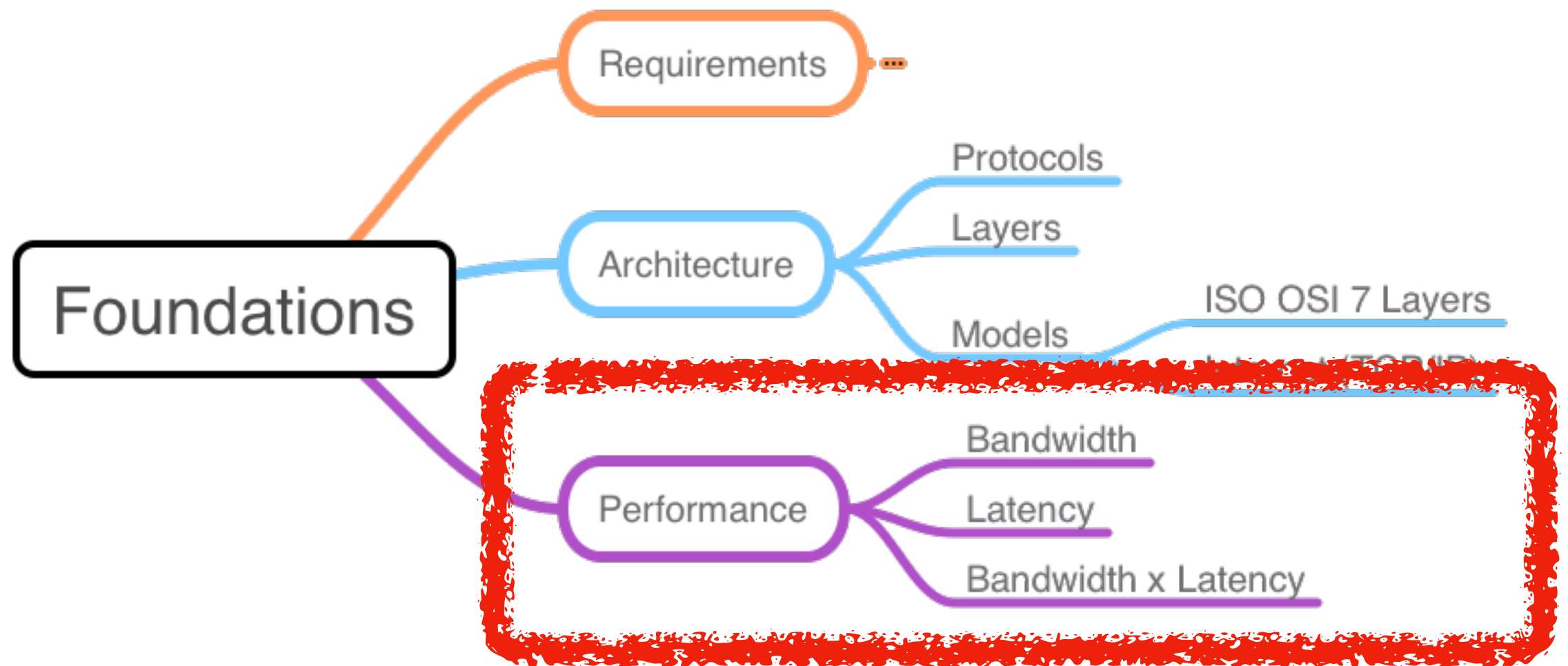
Comparison

## OSI & TCP/IP Protocol-Stacks and Protocols



— No direct mapping —

# Performance

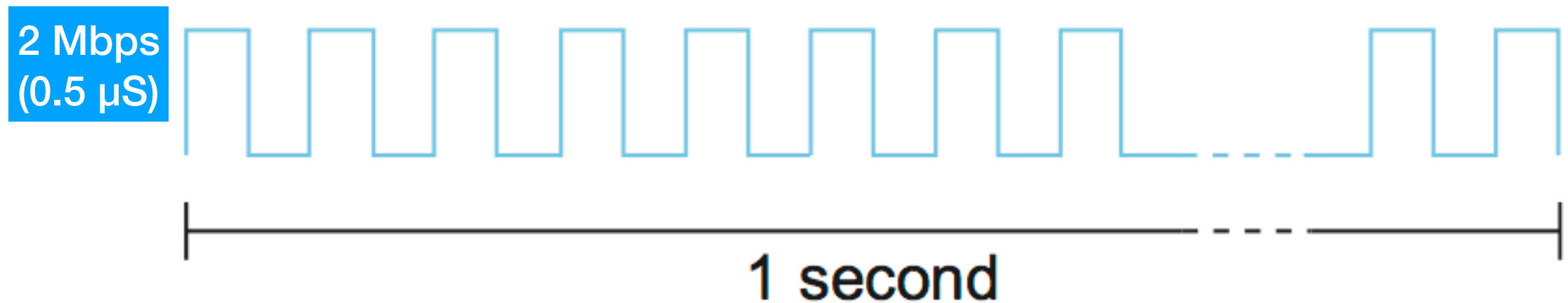
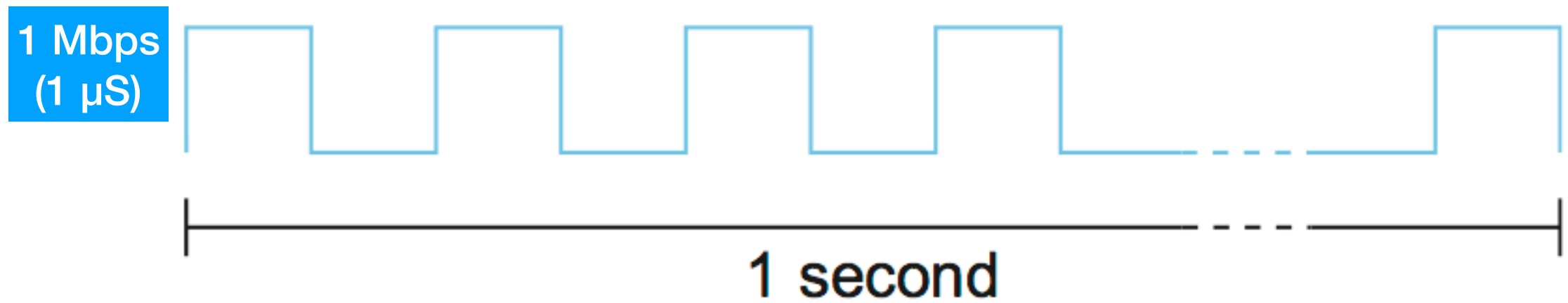


# Bandwidth

Used to denote the number of bits that can be transmitted over the network in a certain period of time, also called the '*data rate*'

Bandwidth is literally a measure of the width of a frequency band

# Bandwidth



# Bandwidth

## 10 Mbps

- 10 million bits per second
- $0.1\mu\text{S}$  for each bit

20 Mbps -  $0.05\mu\text{S}$  for each bit

100 Mbps -  $0.01\mu\text{S}$  for each bit



# Latency

How long it takes a message to travel from one node to another.

$$\text{Latency} = \text{Propagation} + \text{Transmit} + \text{Queue}$$

Where:

$$\text{Propagation} = \text{Distance} / \text{Speed of Light}$$

$$\text{Transmit} = \text{Size} / \text{Bandwidth}$$

# Speed of Light

Light travels across different mediums at different speeds

Medium	Speed of Light
Vacuum	$3 \times 10^8$ m/s
Copper	$2.3 \times 10^8$ m/s
Fiber	$2.0 \times 10^8$ m/s

# Bandwidth vs Latency

Consider a 1 Byte message and a 1 Byte response over a 10 Mbps link with a 10 ms Round Trip Time (RTT)

Does bandwidth or latency dominate the transmission time?

# Bandwidth vs Latency

Consider a 1 Byte message and a 1 Byte response over a 10 Mbps link with a 10 ms Round Trip Time (RTT)

Does bandwidth or latency dominate the transmission time?

When will bandwidth dominate?

# Delay x Bandwidth

## Amount of data “in-flight”

Consider a 10 Mbps link with 50 mS latency

How much data is “on the wire”?

$$10 \text{ Mbps} \times 50 \text{ mS}$$

$$(10 \times 10^6) \text{ bits/sec} \times (50 \times 10^{-3}) \text{ seconds}$$

$$\begin{aligned} 500 \times 10^3 &= 500 \text{ Kb} \\ &= \underline{62.5 \text{ KB}} \end{aligned}$$

# High-speed Networks

- Very large bandwidth
- Latency does **not** change
- Delay x Bandwidth goes up
- More data in-flight during RTT
- Latency begins to dominate transfer times

# Bandwidth & Latency

Importance depends on application

Application	Important
File Transfer	Bandwidth
Small Messages (HTTP, NFS)	Latency
Audio / Video Conference	Variation in Latency (jitter)

4. Calculate the total time required to transfer a 1.5-MB file in the following cases, assuming an RTT of 80 ms, a packet size of 1 KB data, and an initial  $2 \times \text{RTT}$  of “handshaking” before data is sent:
- (a) The bandwidth is 10 Mbps, and data packets can be sent continuously.
  - (b) The bandwidth is 10 Mbps, but after we finish sending each data packet we must wait one RTT before sending the next.
  - (c) The link allows infinitely fast transmit, but limits bandwidth such that only 20 packets can be sent per RTT.
  - (d) Zero transmit time as in (c), but during the first RTT we can send one packet, during the second RTT we can send two packets, during the third we can send four ( $2^{3-1}$ ), etc.  
(A justification for such an exponential increase will be given in [Chapter 6](#).)

# Exercises

Problem 4 from chapter 1 of textbook



# Exercises

1.5 MB file, 80 mS RTT, 1KB packets, handshake 2xRTT,  
10 Mbps network, continuous sending.

data size	=	$(1.5 \times 2^{20}) \times 8$	=	12,582,912 bits
bandwidth	=	$10 \times 10^6$	=	10,000,000
RTT			=	0.080 seconds
packet size	=	$2^{10} \times 8$	=	8,192 bits
handshake	=	$2 \times \text{RTT}$	=	0.16 seconds
total	=	propagation + transmit + queue(ignore)		

# Exercises

1.5 MB file, 80 mS RTT, 1KB packets, handshake 2xRTT,  
10 Mbps network, continuous sending.

data size =  $(1.5 \times 2^{20}) \times 8 = 12,582,912$  bits  
bandwidth =  $10 \times 10^6 = 10,000,000$   
RTT = 0.080 seconds  
packet size =  $2^{10} \times 8 = 8,192$  bits  
handshake =  $2 \times \text{RTT} = 0.16$  seconds  
total = propagation + transmit + queue(ignore)

total = handshake + propagation + transmit  
propagation =  $\text{RTT} / 2 = \underline{0.04 \text{ seconds}}$   
transmit = size / bandwidth =  $\underline{1.258 \text{ seconds}}$   
total =  $0.16 + 0.04 + 1.2583 = \underline{\underline{1.458 \text{ seconds}}}$

# Exercises

1.5 MB file, 80 mS RTT, 1KB packets, handshake 2xRTT,  
10 Mbps network, 1 RTT delay after each packet is sent.

data size	=	$(1.5 \times 2^{20}) \times 8$	=	12,582,912 bits
bandwidth	=	$10 \times 10^6$	=	10,000,000
RTT			=	0.080 seconds
packet size	=	$2^{10} \times 8$	=	8,192 bits
handshake	=	$2 \times \text{RTT}$	=	0.16 seconds
total	=	propagation + transmit + queue(ignore)		

# Exercises

1.5 MB file, 80 mS RTT, 1KB packets, handshake 2xRTT,  
10 Mbps network, 1 RTT delay after each packet is sent.

data size =  $(1.5 \times 2^{20}) \times 8 = 12,582,912$  bits  
bandwidth =  $10 \times 10^6 = 10,000,000$   
RTT = 0.080 seconds  
packet size =  $2^{10} \times 8 = 8,192$  bits  
handshake =  $2 \times \text{RTT} = 0.16$  seconds  
total = propagation + transmit + queue(ignore)

propagation =  $\text{RTT} / 2 = \underline{0.04 \text{ seconds}}$   
transmit =  $\text{size} / \text{bandwidth} = \underline{1.258 \text{ seconds}}$   
packets =  $\text{size} / \text{packet size} = \underline{1,536 \text{ packets}}$   
packet delay =  $(\text{packets} - 1) \times \text{RTT} = 122.8 \text{ seconds}$   
total =  $0.16 + 0.04 + (1.258 + 122.8) = \underline{\underline{124.258 \text{ s}}}$

# Exercises

1.5 MB file, 80 mS RTT, 1KB packets, handshake 2xRTT,  
infinite transmit network, can send 20 packets per RTT

data size =  $(1.5 \times 2^{20}) \times 8 = 12,582,912$  bits

RTT = 0.080 seconds

packet size =  $2^{10} \times 8 = 8,192$  bits

handshake =  $2 \times \text{RTT} = 0.16$  seconds

total = propagation + transmit + queue(ignore)

# Exercises

1.5 MB file, 80 mS RTT, 1KB packets, handshake 2xRTT,  
infinite transmit network, can send 20 packets per RTT

data size      =  $(1.5 \times 2^{20}) \times 8$  = 12,582,912 bits  
RTT                      = 0.080 seconds  
packet size =  $2^{10} \times 8$                       = 8,192 bits  
handshake =  $2 \times \text{RTT}$                       = 0.16 seconds  
total              = propagation + transmit + queue(ignore)

total              = handshake + (chunks x RTT)  
packets      = size / packet size              = 1,536 packets  
chunks      = ceil(packets / 20)              = 77 chunks  
total              =  $0.16 + (77 \times 0.080)$               = **6.32 seconds**

# Exercises

1.5 MB file, 80 mS RTT, 1KB packets, handshake 2xRTT,  
infinite transmit network, can send 20 packets per RTT

data size =  $(1.5 \times 2^{20}) \times 8 = 12,582,912$  bits  
RTT = 0.080 seconds  
packet size =  $2^{10} \times 8 = 8,192$  bits  
handshake =  $2 \times \text{RTT} = 0.16$  seconds  
total = propagation + transmit + queue(ignore)

total = handshake + (chunks x RTT)  
packets = size / packet size = 1,536 packets  
chunks = ceil(packets / 20) = 77 chunks  
total =  $0.16 + (77 \times 0.080) = \underline{6.32 \text{ seconds}}$   
total\* =  $0.16 + (77 \times 0.080) - \underline{0.04} = \underline{6.28 \text{ seconds}}$

Don't need to wait  
for last response  
(- prop delay)

# Exercises

1.5 MB file, 80 mS RTT, 1KB packets, handshake 2xRTT,  
infinite transmit network, send  $2^{(n-1)}$  packets each RTT.  
(1 packet 1st RTT, 2 second RTT, 4 third RTT, ...)

data size =  $(1.5 \times 2^{20}) \times 8 = 12,582,912$  bits

RTT = 0.080 seconds

packet size =  $2^{10} \times 8 = 8,192$  bits

handshake =  $2 \times \text{RTT} = 0.16$  seconds

total = propagation + transmit + queue(ignore)



# Exercises

1.5 MB file, 80 mS RTT, 1KB packets, handshake 2xRTT,  
infinite transmit network, send  $2^{(n-1)}$  packets each RTT.  
(1 packet 1st RTT, 2 second RTT, 4 third RTT, ...)

data size      =  $(1.5 \times 2^{20}) \times 8$  = 12,582,912 bits  
RTT                      = 0.080 seconds  
packet size =  $2^{10} \times 8$                       = 8,192 bits  
handshake =  $2 \times \text{RTT}$                       = 0.16 seconds  
total              = propagation + transmit + queue(ignore)

total              = handshake + (chunks x RTT) - prop\_delay  
packets      = size / packet size              = 1,536 packets  
chunks      = ceil(log<sub>2</sub>(packets))              = 11 chunks

# Exercises

1.5 MB file, 80 mS RTT, 1KB packets, handshake 2xRTT,  
infinite transmit network, send  $2^{(n-1)}$  packets each RTT.  
(1 packet 1st RTT, 2 second RTT, 4 third RTT, ...)

data size =  $(1.5 \times 2^{20}) \times 8 = 12,582,912$  bits  
RTT = 0.080 seconds  
packet size =  $2^{10} \times 8 = 8,192$  bits  
handshake =  $2 \times \text{RTT} = 0.16$  seconds  
total = propagation + transmit + queue(ignore)

total = handshake + (chunks x RTT) - prop\_delay  
packets = size / packet size = 1,536 packets  
chunks =  $\text{ceil}(\log_2(\text{packets})) = \underline{11 \text{ chunks}}$   
  
total\* =  $0.16 + (11 \times 0.080) - \underline{0.04} = \underline{\mathbf{1 \text{ second}}}$

3. Calculate the total time required to transfer a 1000-KB file in the following cases, assuming an RTT of 50 ms, a packet size of 1 KB data, and an initial  $2 \times \text{RTT}$  of “handshaking” before data is sent:
- (a) The bandwidth is 1.5 Mbps, and data packets can be sent continuously.
  - (b) The bandwidth is 1.5 Mbps, but after we finish sending each data packet we must wait one RTT before sending the next.
  - (c) The bandwidth is “infinite,” meaning that we take transmit time to be zero, and up to 20 packets can be sent per RTT.
  - (d) The bandwidth is infinite, and during the first RTT we can send one packet ( $2^{1-1}$ ), during the second RTT we can send two packets ( $2^{2-1}$ ), during the third we can send four ( $2^{3-1}$ ), and so on. (A justification for such an exponential increase will be given in [Chapter 6](#).)

# Exercise

Problem 3 from chapter 1 of textbook

## # Exercise 1.3

1,000KB file,

50ms RTT,

1KB packet size,

2 RTT handshake at start

$\text{size} = (1,000 * 2^{10}) * 8 \Rightarrow 8,192,000$

$\text{rtt} = 0.050 \Rightarrow 0.05$

$\text{packet\_size} = 2^{10} * 8 \Rightarrow 8,192$

$\text{handshake} = 2 * \text{rtt} \Rightarrow 0.1$

### # Exercise 1.3

1,000KB file, RTT 50ms, 2 RTT handshake at start

$$\text{size} = (1,000 * 2^{10}) * 8 \Rightarrow 8,192,000$$

$$\text{rtt} = 0.050 \Rightarrow 0.05$$

$$\text{packet\_size} = 2^{10} * 8 \Rightarrow 8,192$$

$$\text{handshake} = 2 * \text{rtt} \Rightarrow 0.1$$

a) 1.5Mbps, continuous

$$\text{bandwidth} = 1.5 * 10^6 \Rightarrow 1,500,000$$

$$\text{propagation\_delay} = \text{rtt} / 2 \Rightarrow 0.025$$

$$\text{transmit\_time} = \text{size} / \text{bandwidth} \Rightarrow 5.4613$$

$$\text{handshake} + \text{propagation\_delay} + \text{transmit\_time} \Rightarrow 5.5863$$

### # Exercise 1.3

1,000KB file, RTT 50ms, 2 RTT handshake at start

$$\text{size} = (1,000 * 2^{10}) * 8 \Rightarrow 8,192,000$$

$$\text{rtt} = 0.050 \Rightarrow 0.05$$

$$\text{packet\_size} = 2^{10} * 8 \Rightarrow 8,192$$

$$\text{handshake} = 2 * \text{rtt} \Rightarrow 0.1$$

b) 1.5Mbps, one RTT after each packet

$$\text{bandwidth} = 1.5 * 10^6 \Rightarrow 1,500,000$$

$$\text{propagation\_delay} = \text{rtt} / 2 \Rightarrow 0.025$$

$$\text{transmit\_time} = \text{size} / \text{bandwidth} \Rightarrow 5.4613$$

$$\text{packet\_count} = \text{size} / \text{packet\_size} \Rightarrow 1,000$$

$$\begin{aligned} \text{t\_time} &= \text{transmit\_time} + ((\text{packet\_count} - 1) * \text{rtt}) \\ &\Rightarrow 55.4113 \end{aligned}$$

$$\text{handshake} + \text{propagation\_delay} + \text{t\_time} \Rightarrow 55.5363$$

### # Exercise 1.3

1,000KB file, RTT 50ms, 2 RTT handshake at start

$$\text{size} = (1,000 * 2^{10}) * 8 \Rightarrow 8,192,000$$

$$\text{rtt} = 0.050 \Rightarrow 0.05$$

$$\text{packet\_size} = 2^{10} * 8 \Rightarrow 8,192$$

$$\text{handshake} = 2 * \text{rtt} \Rightarrow 0.1$$

c) infinite bandwidth, 20 packets per RTT

$$\text{packet\_count} = \text{size} / \text{packet\_size} \Rightarrow 1,000$$

$$\text{packet\_chunks} = \text{ceil}(\text{packet\_count} / 20) \Rightarrow 50$$

$$\text{handshake} + (\text{packet\_chunks} * \text{rtt}) \Rightarrow 2.6$$

$$\begin{aligned} &\text{handshake} + (\text{packet\_chunks} * \text{rtt}) \\ &\quad - \text{propagation\_delay} \Rightarrow 2.575 \end{aligned}$$

### # Exercise 1.3

1,000KB file, RTT 50ms, 2 RTT handshake at start

$\text{size} = (1,000 * 2^{10}) * 8 \Rightarrow 8,192,000$

$\text{rtt} = 0.050 \Rightarrow 0.05$

$\text{packet\_size} = 2^{10} * 8 \Rightarrow 8,192$

$\text{handshake} = 2 * \text{rtt} \Rightarrow 0.1$

d) exponential packet counts, 1, 2, 4, 8, ...

$\text{packet\_count} = \text{size} / \text{packet\_size} \Rightarrow 1,000$

$\text{packet\_chunks} = \text{ceil}(\log(\text{packet\_count}, 2)) \Rightarrow 10$

Don't need to wait for response on last packet to calculate arrival times.

$\text{handshake} + (\text{packet\_chunks} * \text{rtt})$   
 $- \text{propagation\_delay} \Rightarrow 0.575$



**fin**

Foundations