communication networks laboratory

Computer Networks

Sándor Laki

ELTE-Ericsson Communication Networks Laboratory

ELTE FI – Department Of Information Systems

lakis@elte.hu

http://lakis.web.elte.hu

Eötvös Loránd University

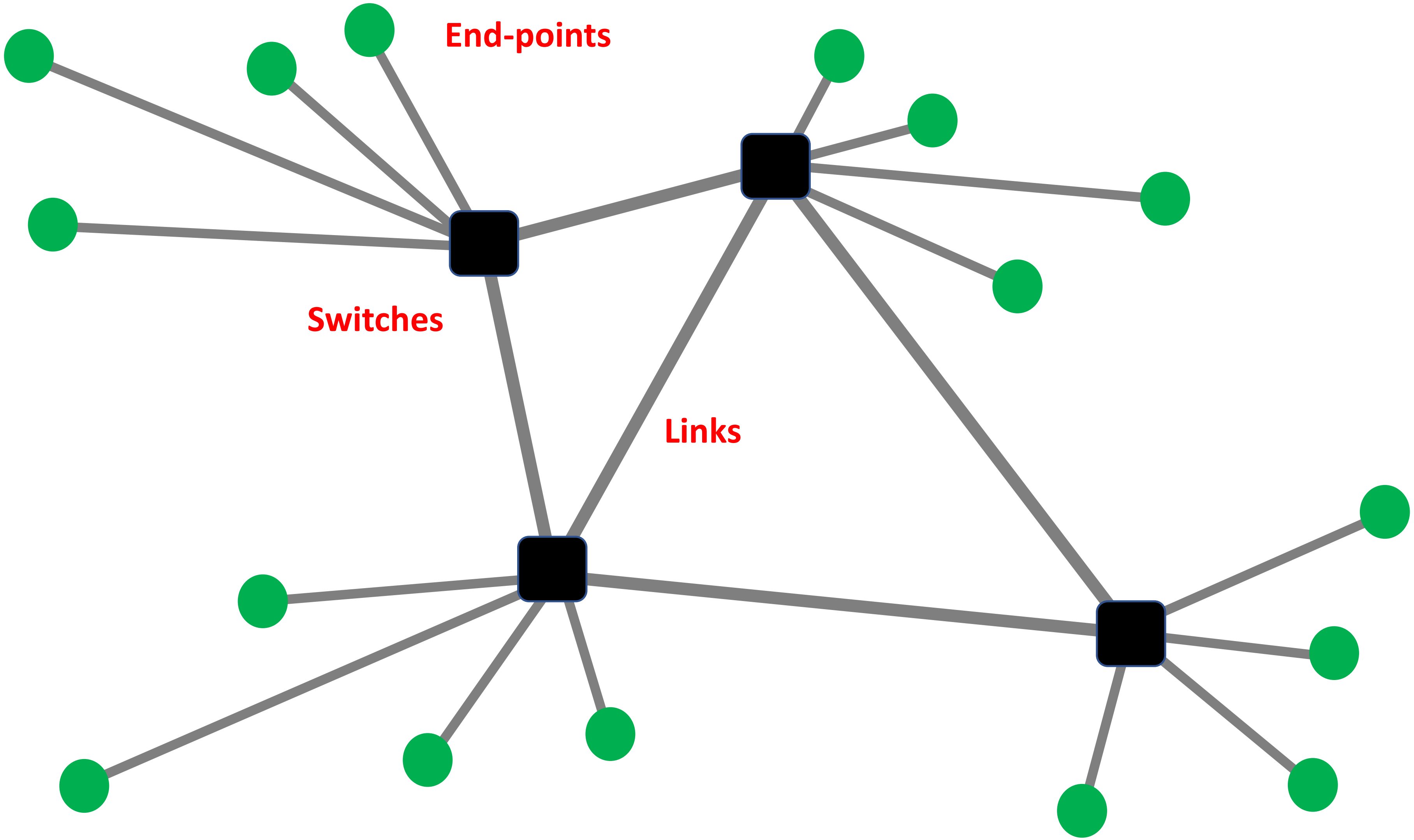
Based on the slides of Laurent Vanbever.
Further inspiration: Scott Shenker & Jennifer Rexford & Phillipa Gill

Last week on
Computer Networks

Overview

What is a network made of?

Three main components



Overview

How to share network resources?

Resource handling
Two different approaches for sharing

Reservation

Reserve the needed bandwidth in advance

Flow-level multiplexing

On-demand

Send data when needed

Packet-level multiplexing

Pros & Cons

Pros

Predictable performance

Simple and fast switching
once circuit established

Cons

Low efficiency

- Bursty traffic
- Short flows

Complexity of circuit establ./teard.
Increased delay

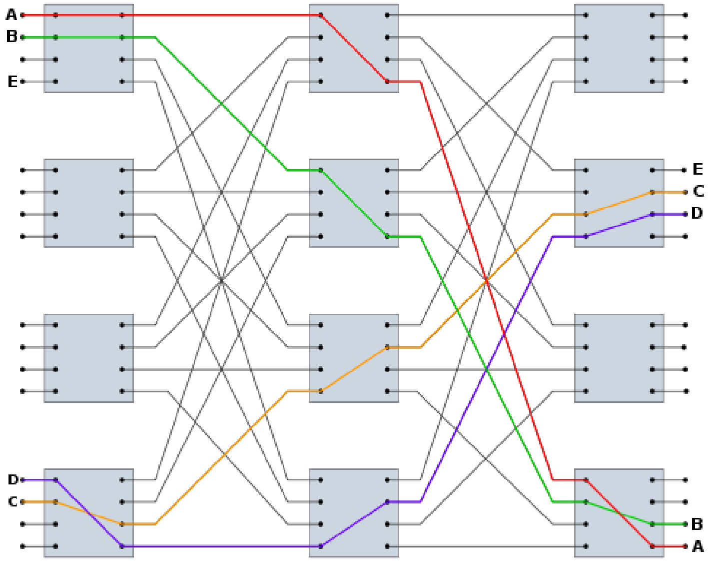
New circuit is needed in case of failures

Implementation

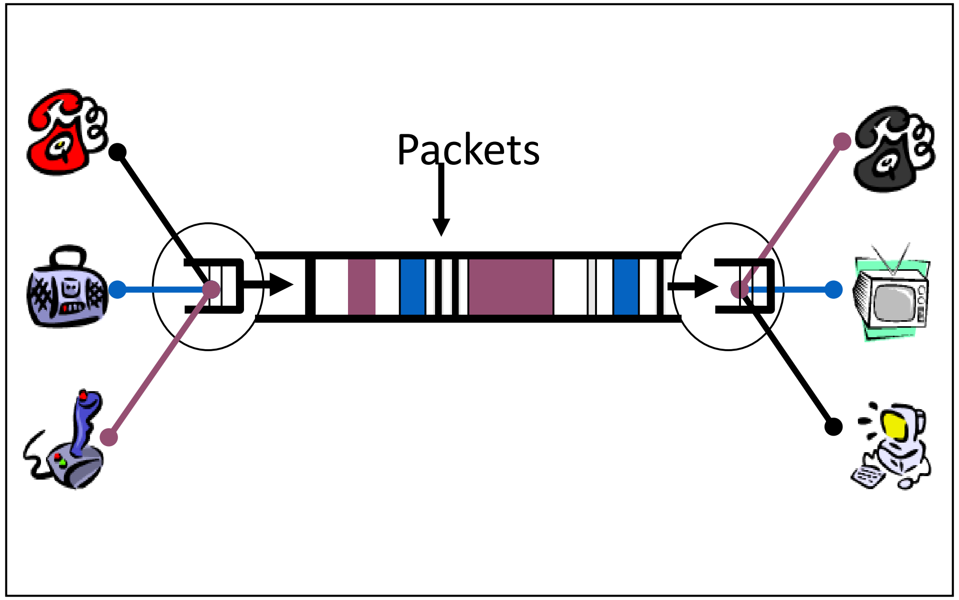
Reservation

On-demand

Circuit-switching
e.g. landline phone networks

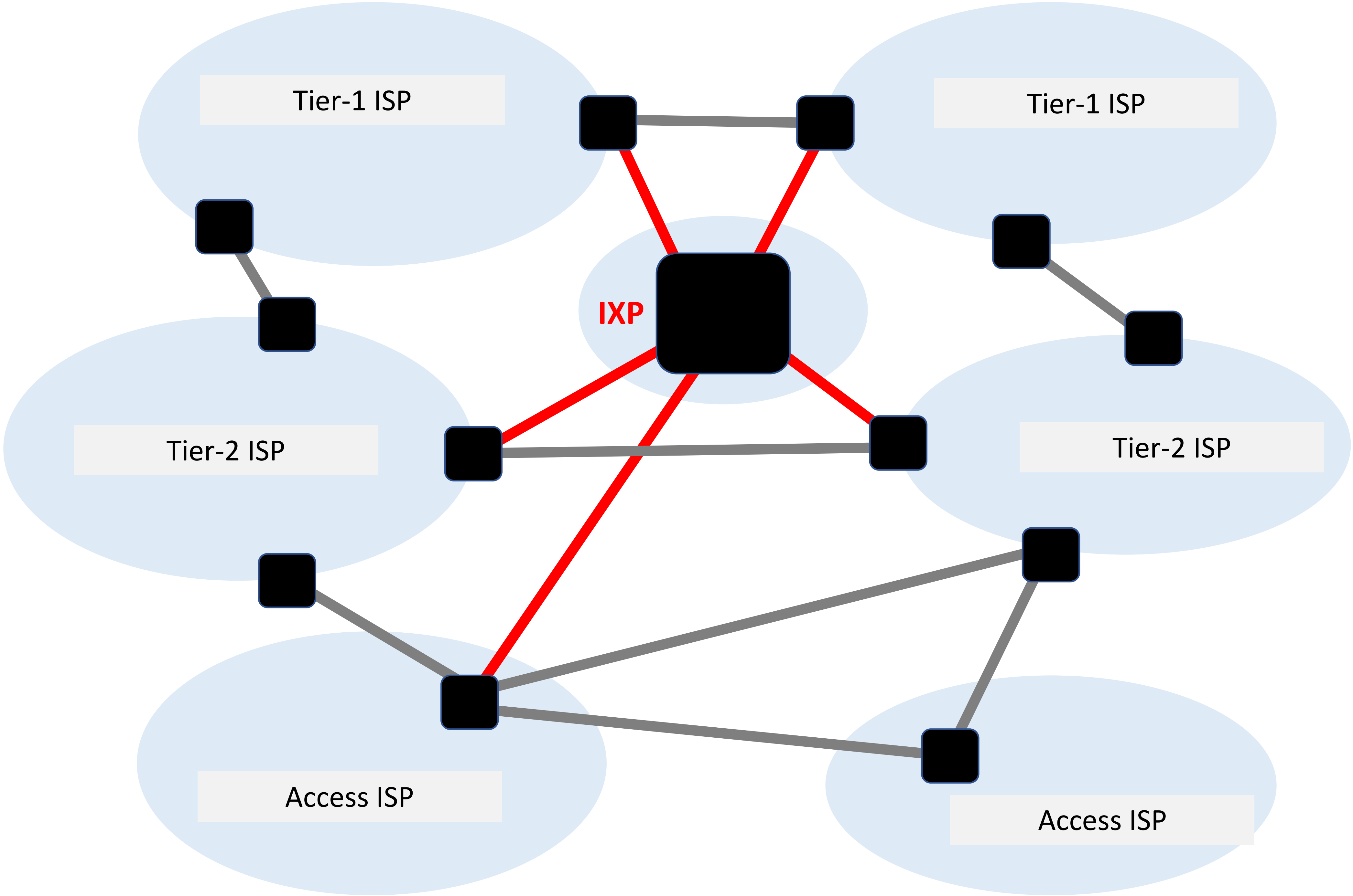


Packet-switching
e.g. Internet



Overview

How to organize the network?


Tier-1 ISP
Tier-1 ISP
IXP
Tier-2 ISP
Tier-2 ISP
Access ISP
Access ISP

This week

How does communication happen?

How do we characterize it?

Briefly…

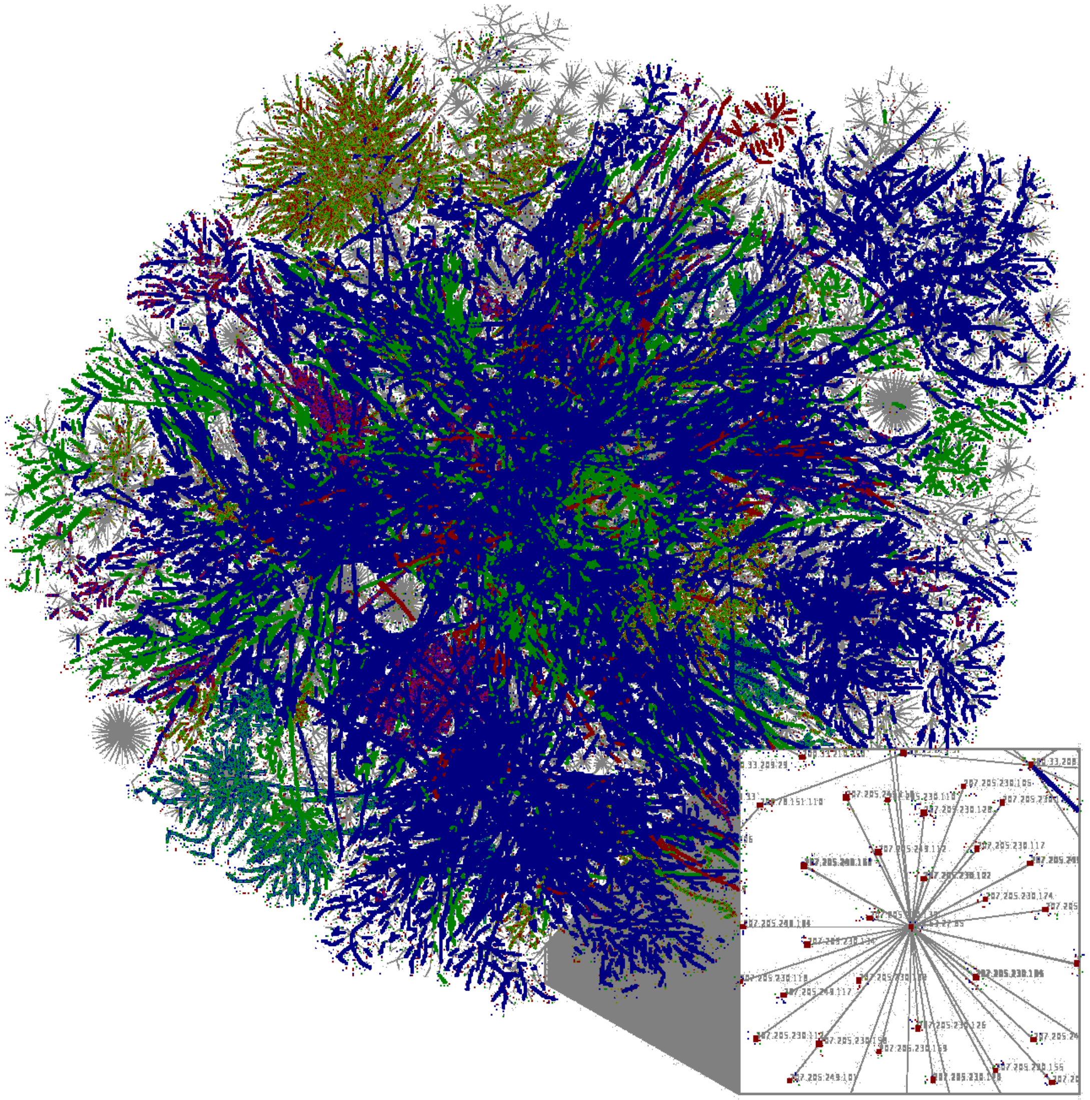
The Internet should allow

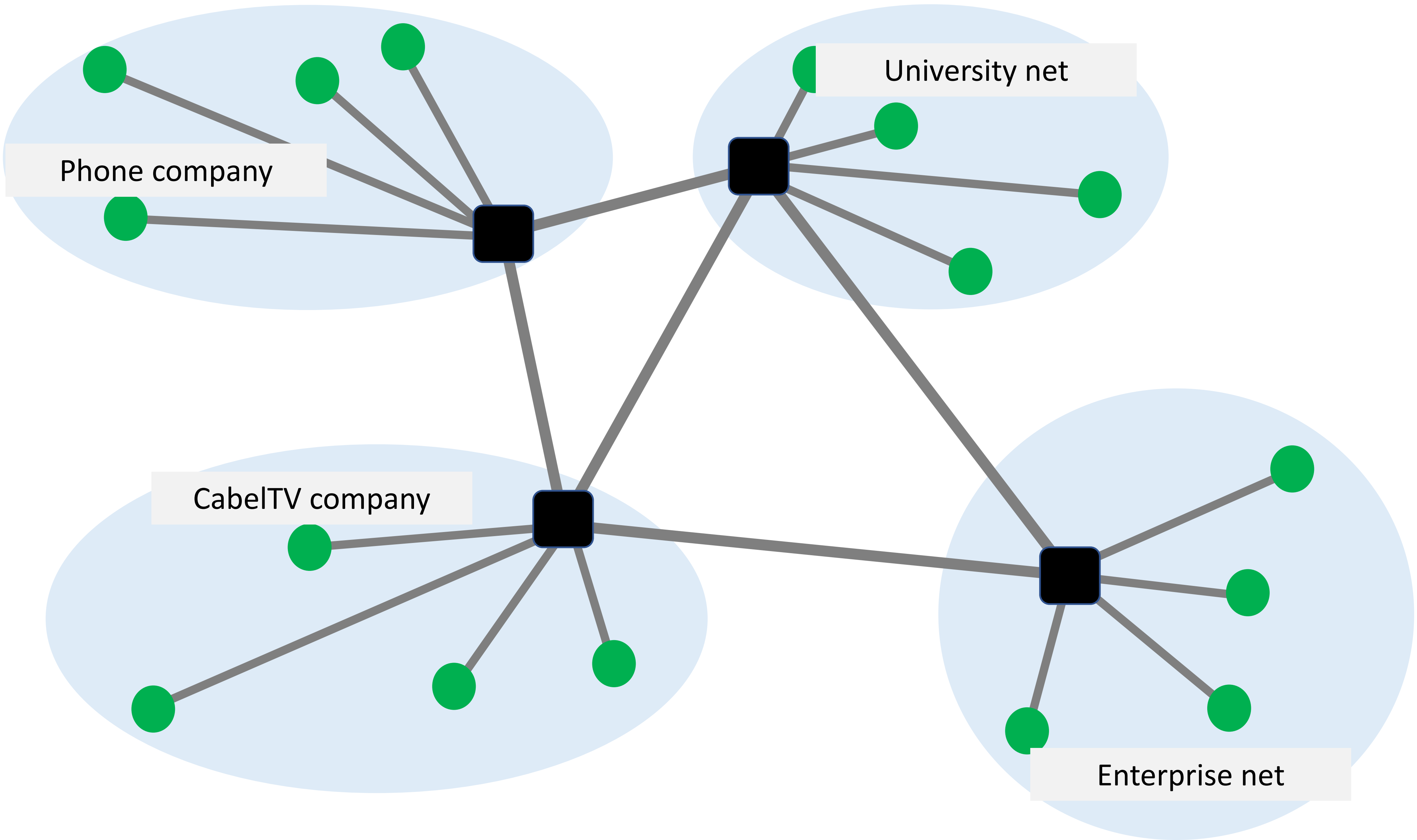
processes on different hosts

to exchange data

everything else is just commentary…

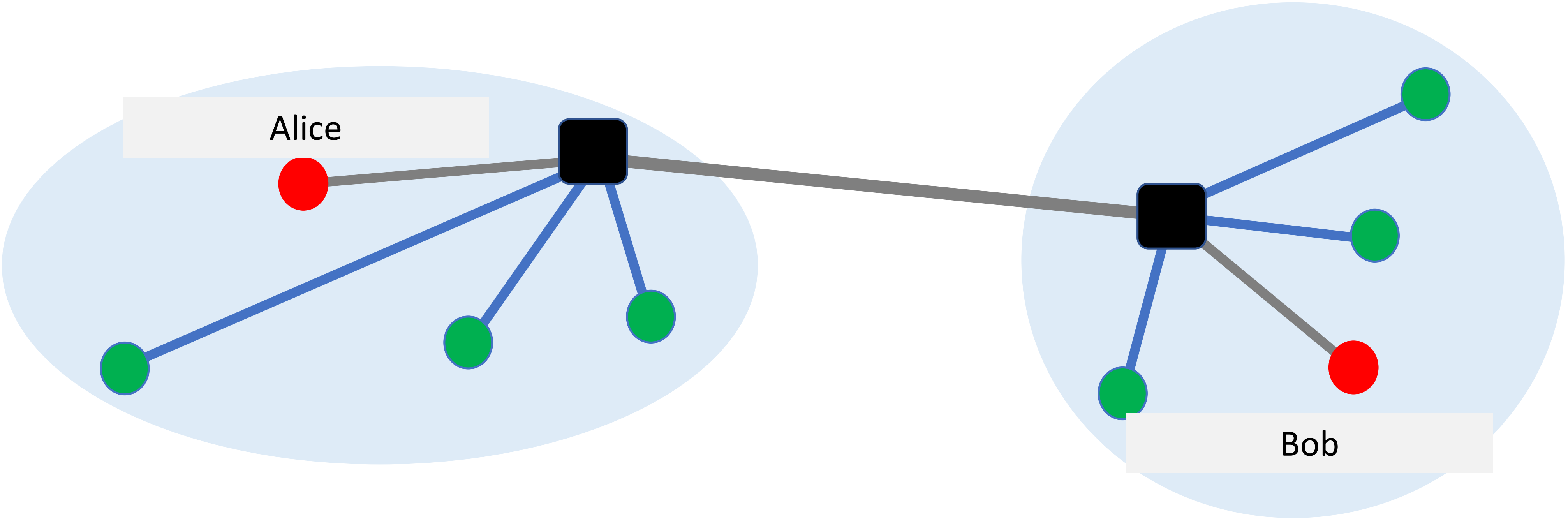
Ok, but how to do that in a complex system like the Internet?





Phone company
University net
CabelTV company
Enterprise net

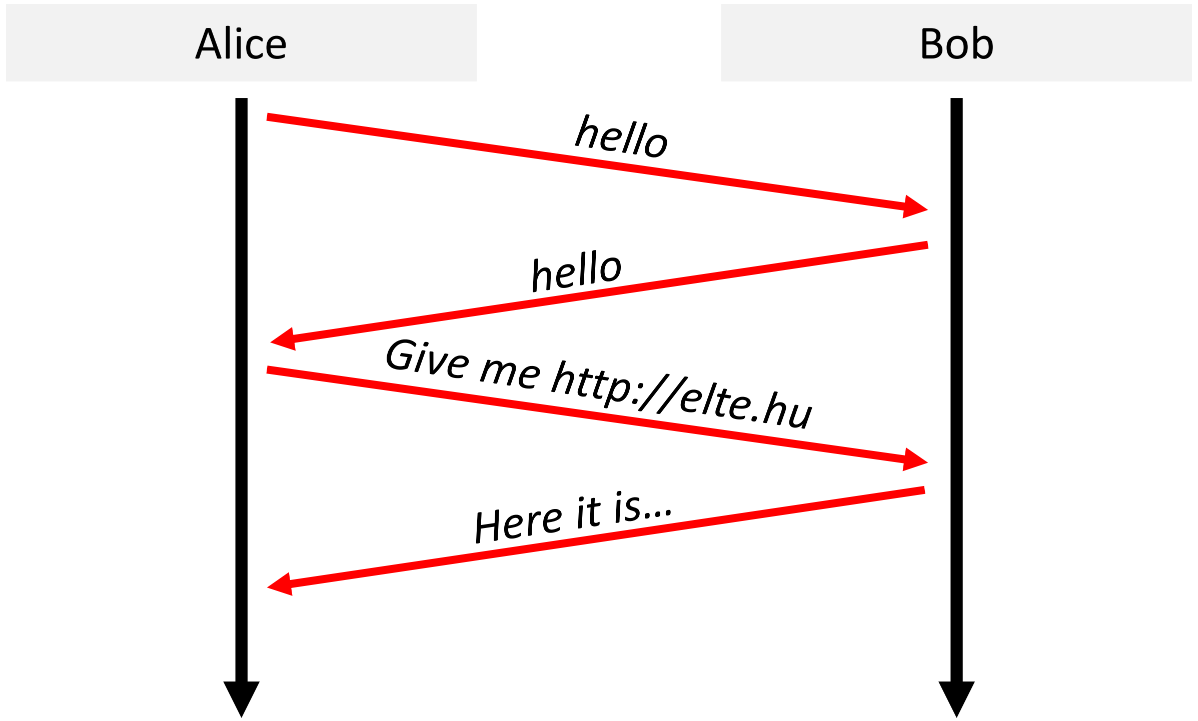
To exchange data, Alice and Bob use
a set of network protocols



A protocol is like a conversational convention

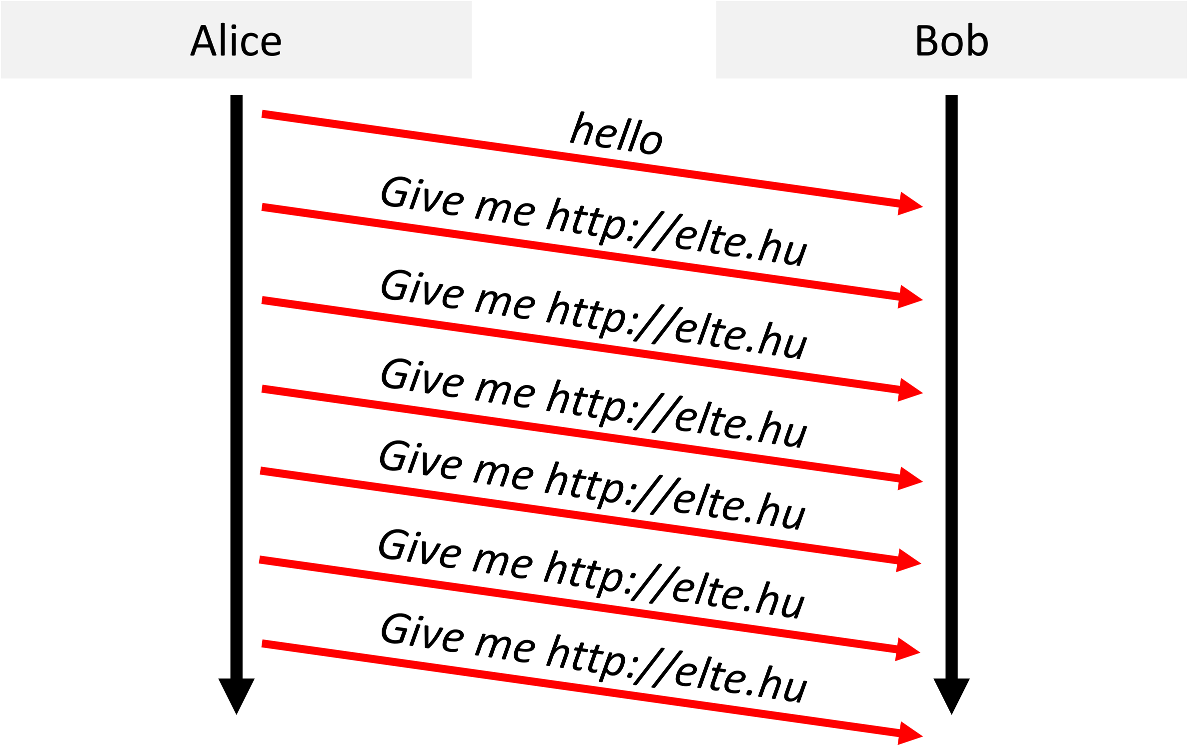
The protocol defines the order and rules the parties should follow

Who should talk next and how to respond…

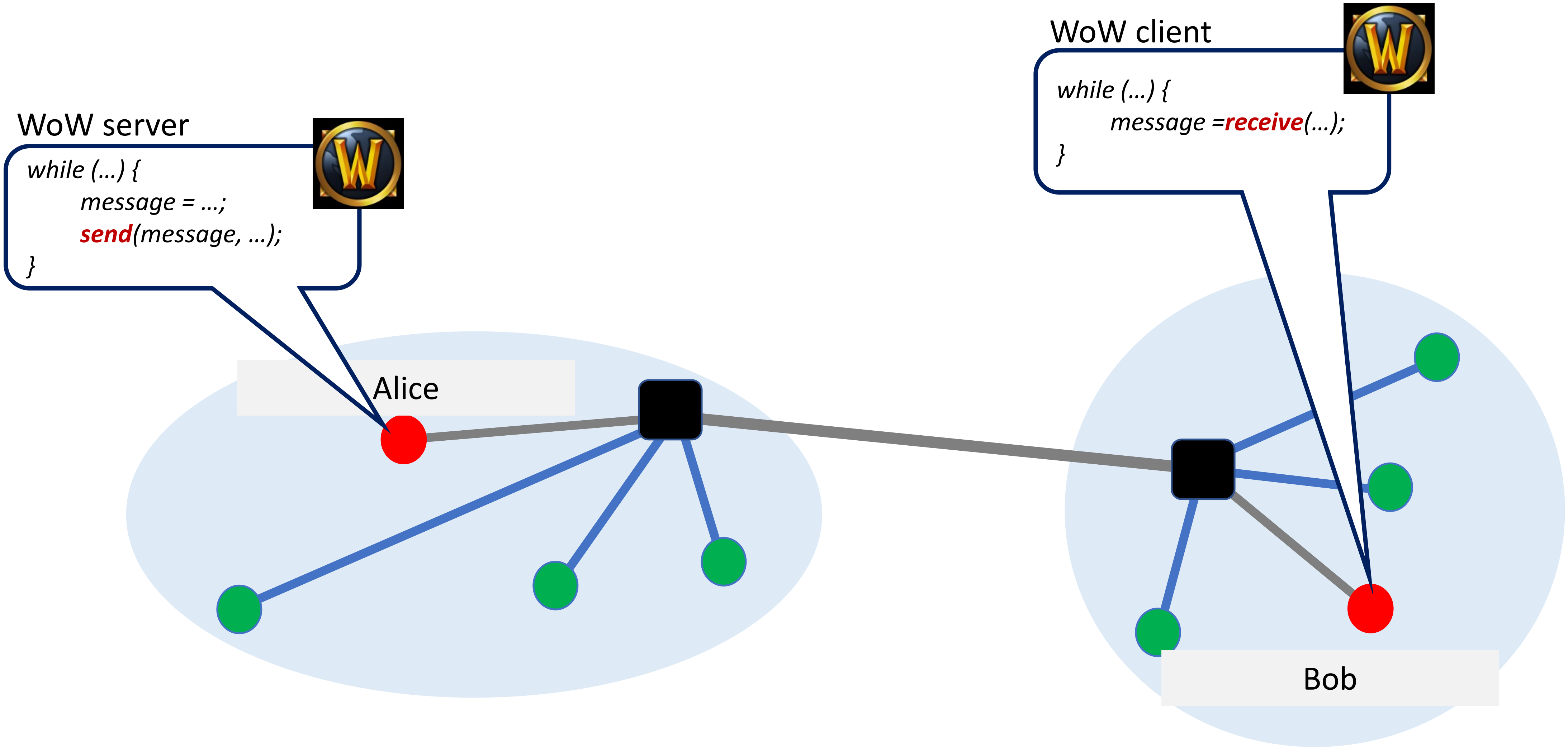


There are other kind of implementations…

Gimme, gimme, gimme
a web site after Midnight

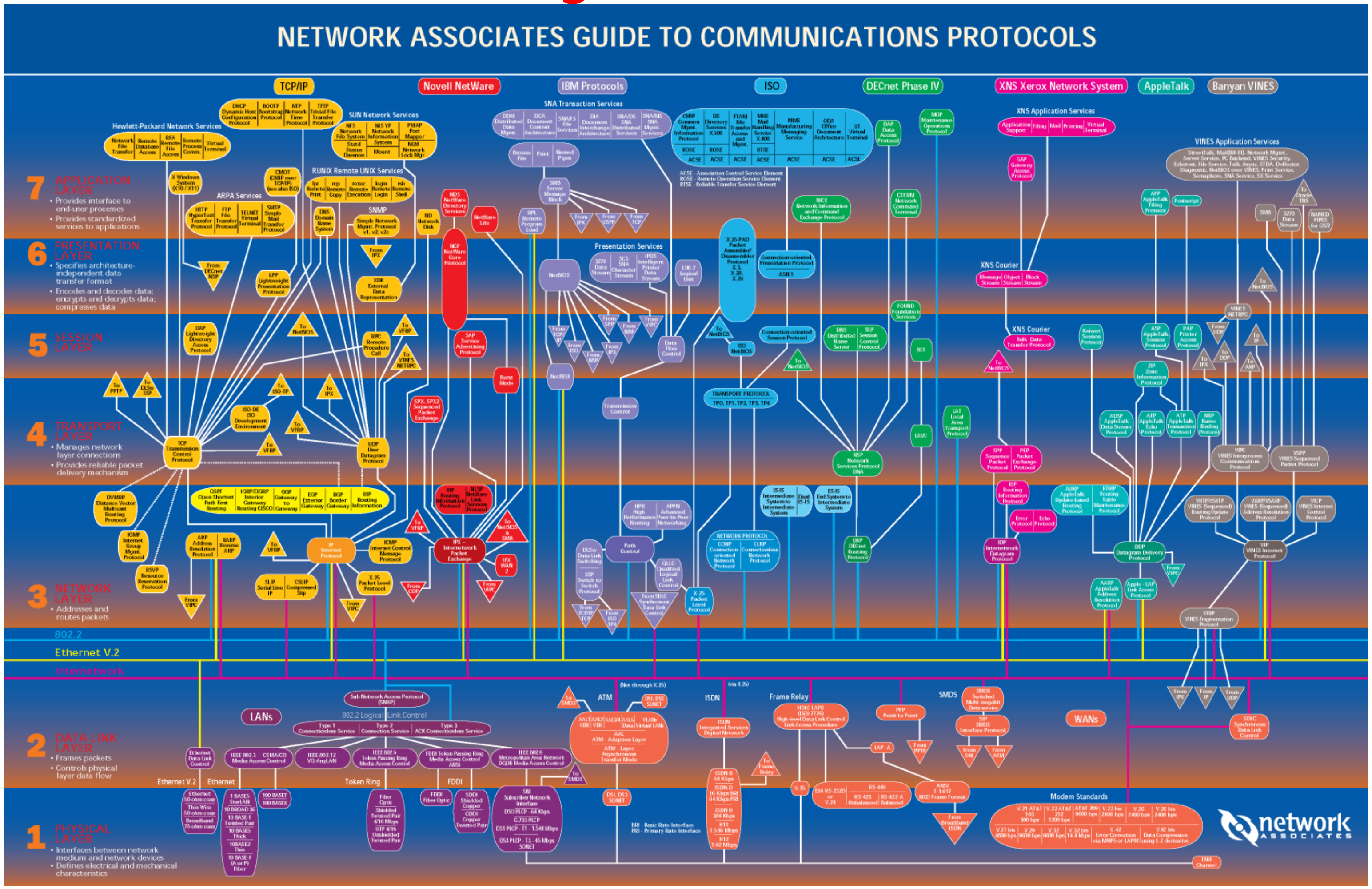


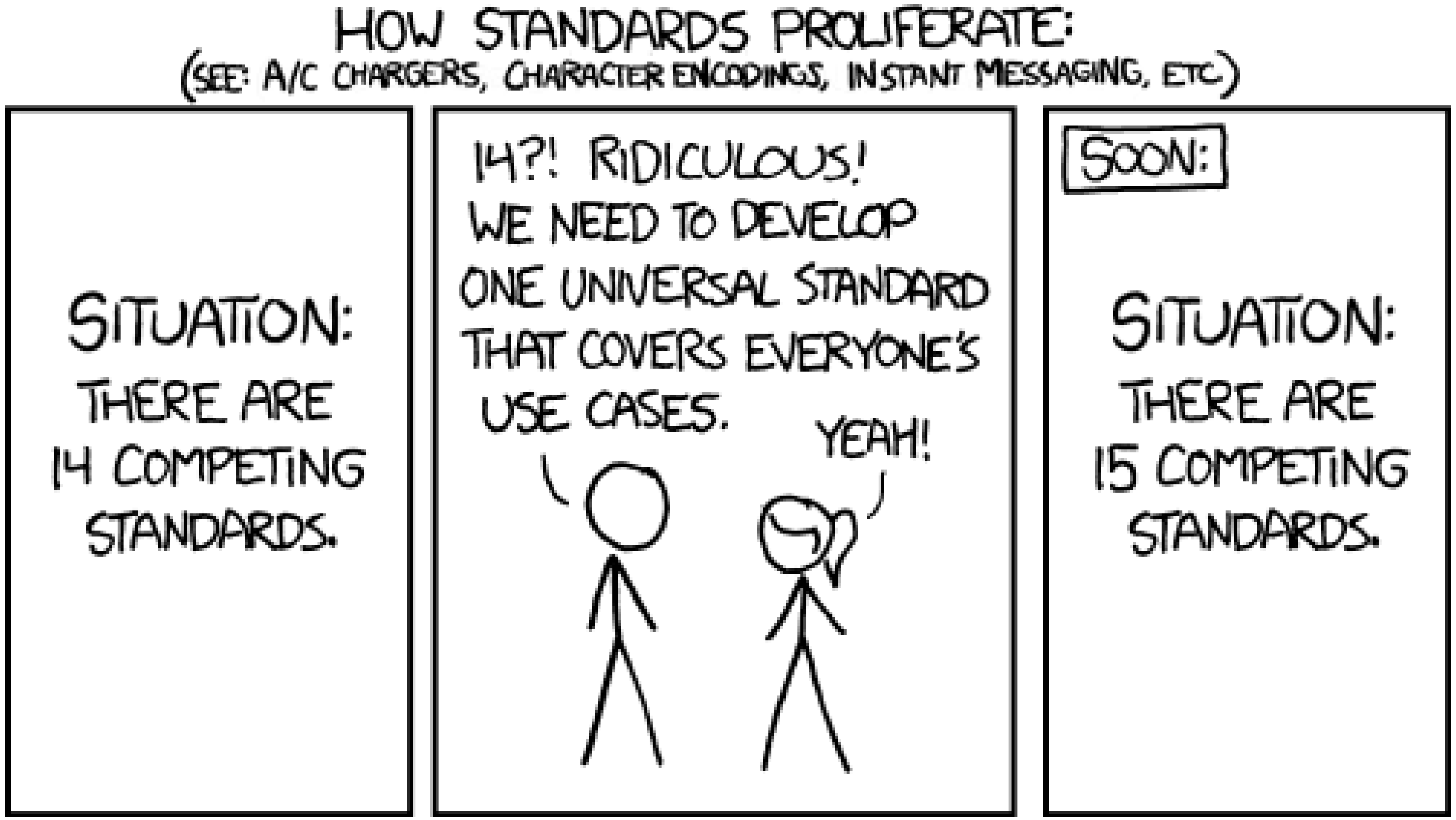
Each protocol is governed by a specific API



In practice, many existing protocols…

How does the Internet ***organize this****???*




HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC.)
SITUATION:
THERE ARE 14 COMPETING STANDARDS.
14?! RIDICULOUS! WE NEED TO DEVELOP ONE UNIVERSAL STANDARD THAT COVERS EVERYONE'S USE CASES.
YEAH!
SOON:
SITUATION:
THERE ARE 15 COMPETING STANDARDS.

Modularity is a key component of any good system

Problem **can't build large systems out of spaghetti code**

hard (if not, impossible) to understand, debug, update

need to bound the scope of changes

evolve the system without rewriting it from scratch

Solution **Modularity is how we do it**

…and understand the system at a higher-level

„Modularity,
based on abstraction,
is **the way** things get done”

Barbara Liskov, MIT

To provide structure to the design of network protocols, network designers organize protocols in layers

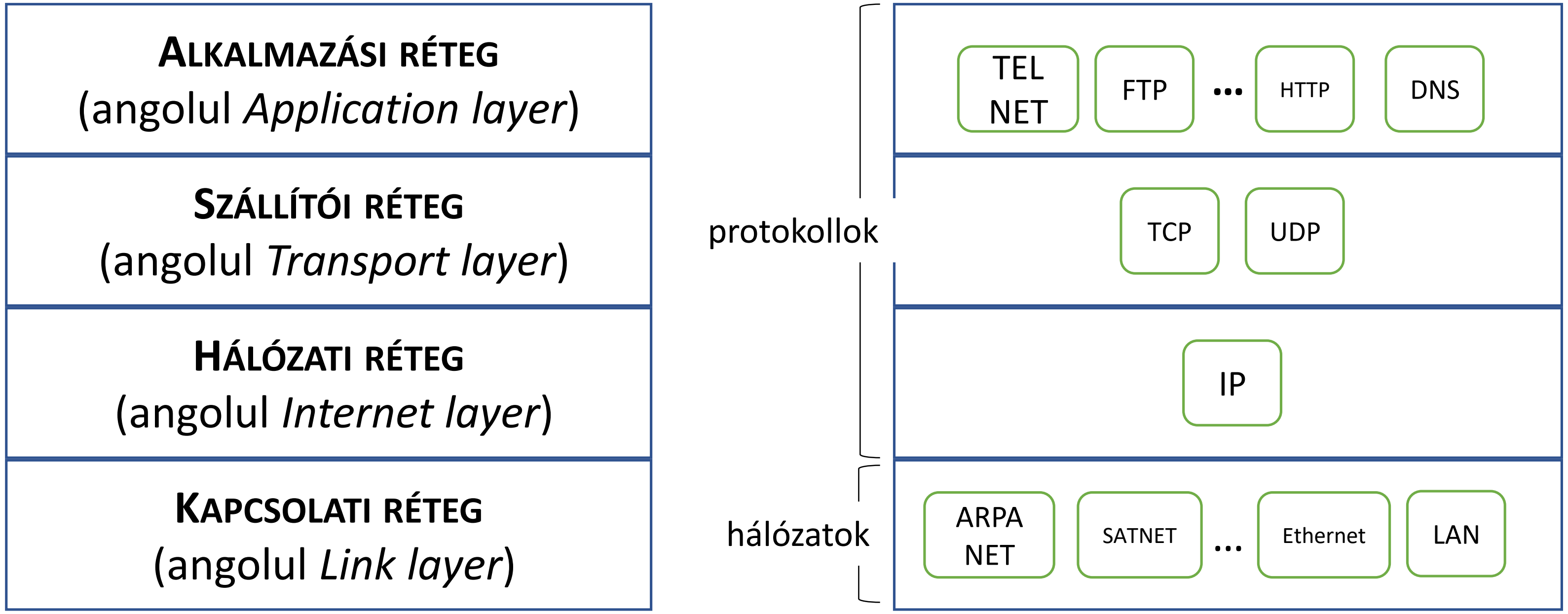
and the network hardware/software that implement them

Hálózatok modelljei

- Internet rétegmodelljei
  - TCP/IP modell: 4 réteget különböztet meg. 1982 márciusában az amerikai hadászati célú számítógépes hálózatok standardja lett. 1985-től népszerűsítették kereskedelmi felhasználásra. (*Interop*)
  - Hibrid TCP/IP modell: 5 réteget különböztet meg (*Tanenbaum, Stallings, Kurose, Forouzan*)

- Nyílt rendszerek hálózatának standard modellje
  - ***O**pen **S**ystem **I**nterconnection Reference Model*: Röviden OSI referencia modell, amely egy 7-rétegű standard, koncepcionális modellt definiál kommunikációs hálózatok belső funkcionalitásaihoz. (*ISO/IEC 7498-1*)

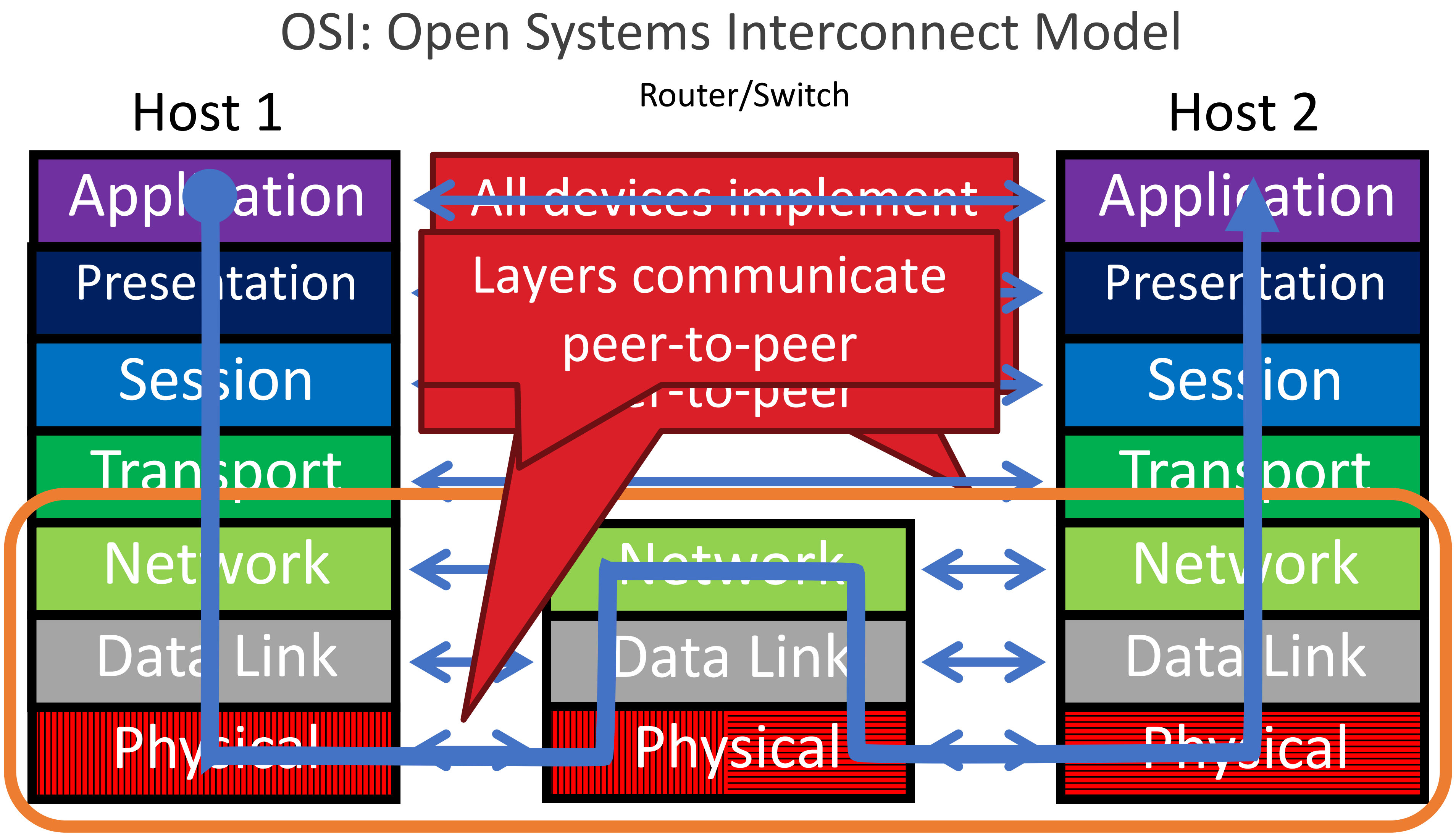
TCP/IP modell (RFC 1122)



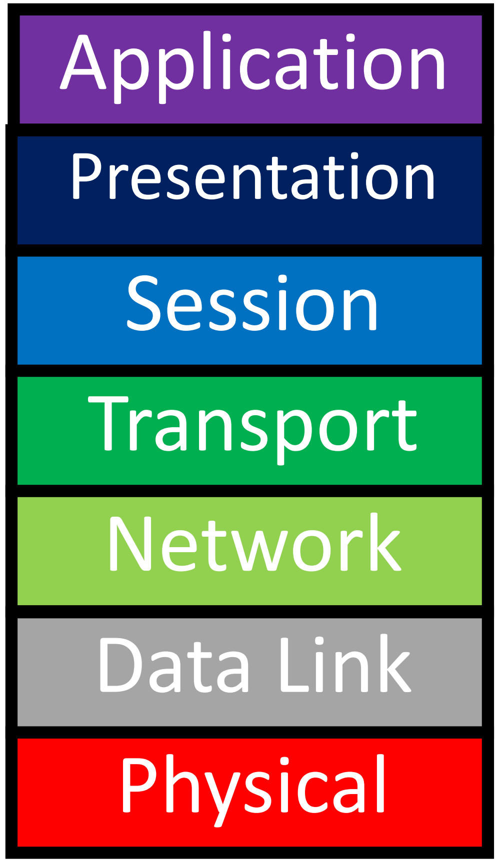
TCP/IP modell rétegei („bottom-up”)

- **Kapcsolati réteg / Host-to-network or Link layer**
  - nem specifikált
  - a LAN-tól függ
- **Internet réteg / Internet or Network layer**
  - speciális csomagformátum
  - útvonal meghatározás (routing)
  - csomag továbbítás (angolul *packet forwarding*)
- **Szállítói réteg / Transport layer**
  - **T**ransport **C**ontrol **P**rotocol
    - megbízható, kétirányú bájt-folyam átviteli szolgáltatás
    - szegmentálás, folyamfelügyelet, multiplexálás
  - **U**ser **D**atagram **P**rotocol
    - nem megbízható átviteli szolgáltatás
    - nincs folyamfelügyelet
- **Alkalmazási réteg / Application layer**
  - Szolgáltatások nyújtása: Telnet, FTP, SMTP, HTTP, NNTP, DNS, SSH, etc.

The ISO OSI Model

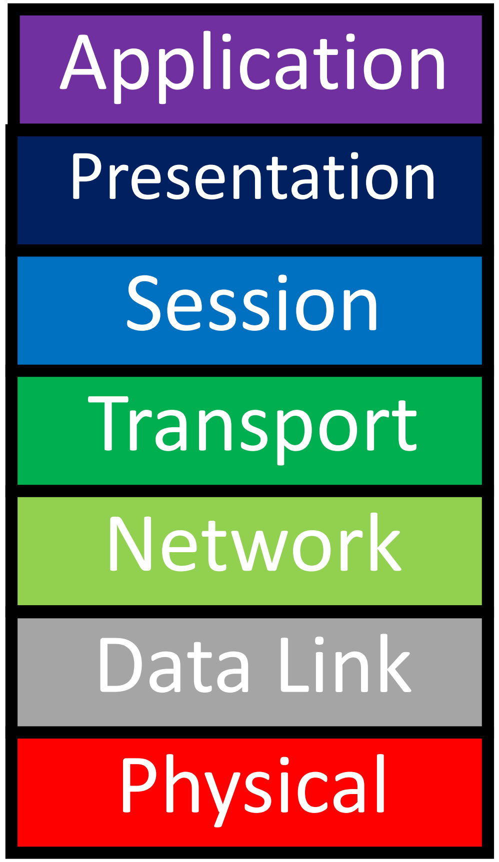


Layer Features



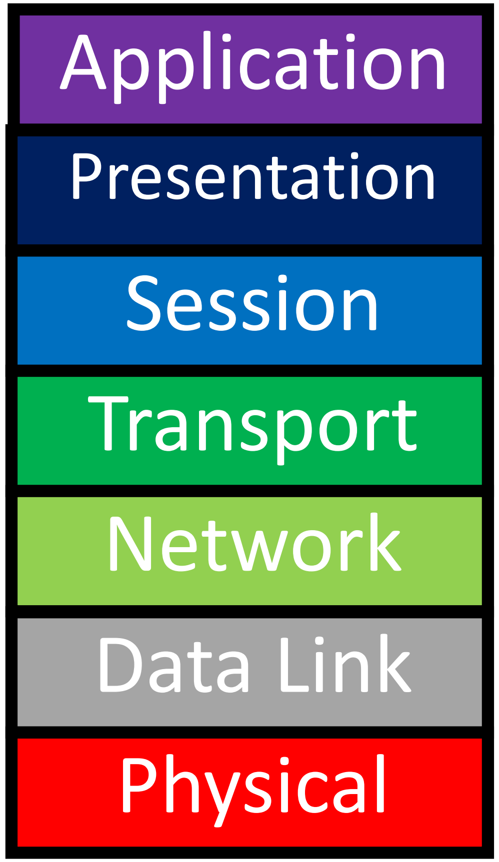
- Service
  - What does this layer do?
- Interface
  - How do you access this layer?
- Protocol
  - How is this layer implemented?

Physical Layer



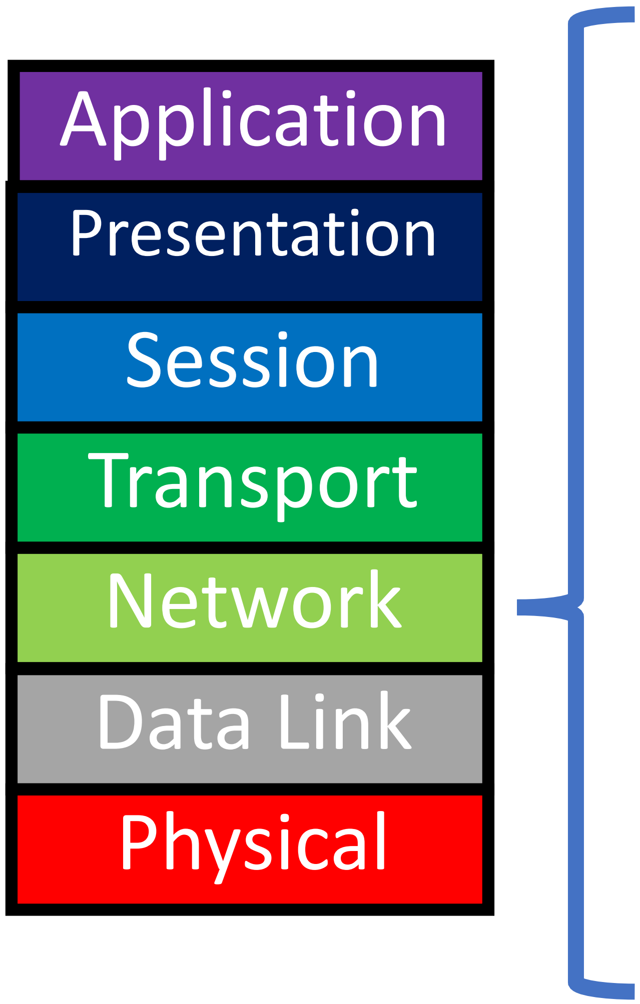
- Service
  - Move information between two systems connected by a physical link
- Interface
  - Specifies how to send one bit
- Protocol
  - Encoding scheme for one bit
  - Voltage levels
  - Timing of signals
- Examples: coaxial cable, fiber optics, radio frequency transmitters

Data Link Layer



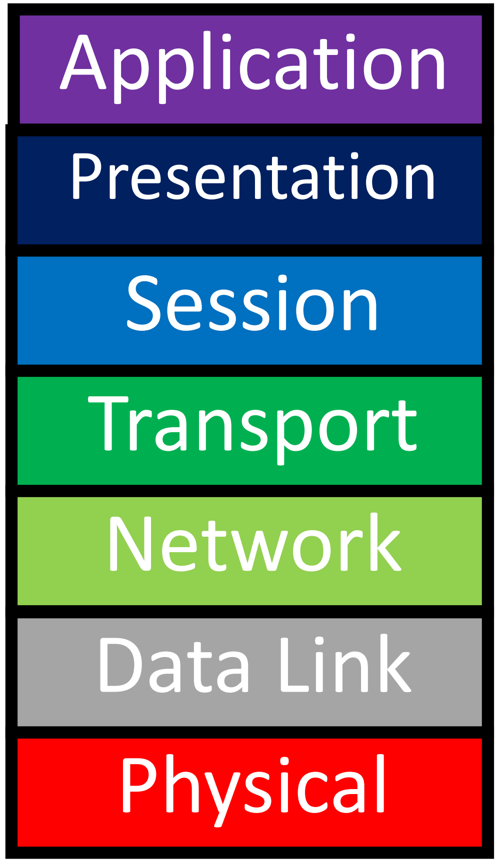
- Service
  - Data framing: boundaries between packets
  - Media access control (MAC)
  - Per-hop reliability and flow-control
- Interface
  - Send one packet between two hosts connected to the same media
- Protocol
  - Physical addressing (e.g. MAC address)
- Examples: Ethernet, Wifi, DOCSIS

Network Layer



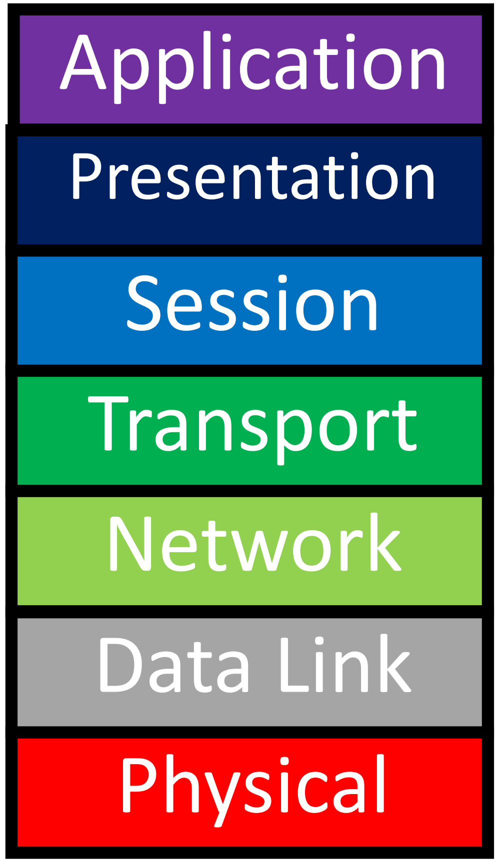
- Service
  - Deliver packets across the network
  - Handle fragmentation/reassembly
  - Packet scheduling
  - Buffer management
- Interface
  - Send one packet to a specific destination
- Protocol
  - Define globally unique addresses
  - Maintain routing tables
- Example: Internet Protocol (IP), IPv6

Transport Layer



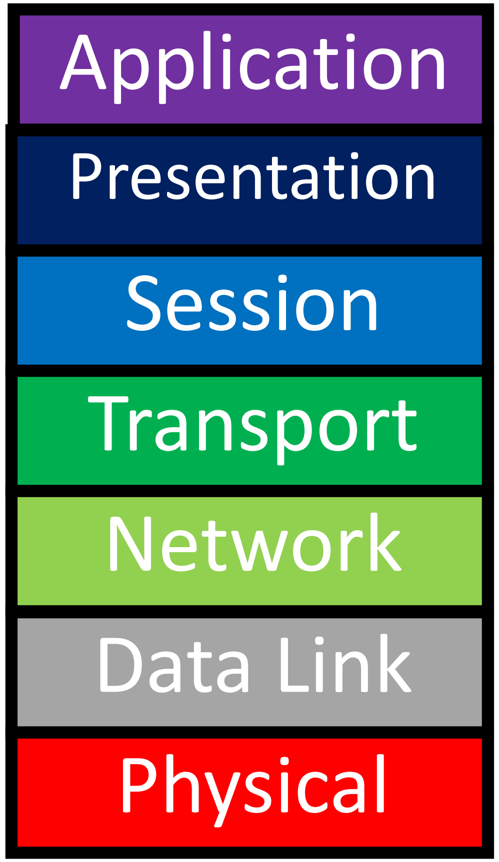
- Service
  - Multiplexing/demultiplexing
  - Congestion control
  - Reliable, in-order delivery
- Interface
  - Send message to a destination
- Protocol
  - Port numbers
  - Reliability/error correction
  - Flow-control information
- Examples: UDP, TCP

Session Layer



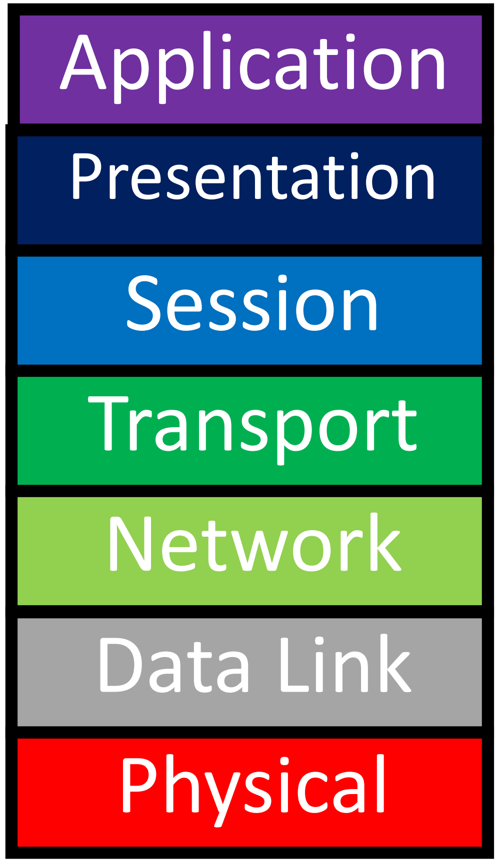
- Service
  - Access management
  - Synchronization
- Interface
  - It depends…
- Protocol
  - Token management
  - Insert checkpoints
- Examples: none

Presentation Layer



- Service
  - Convert data between different representations
  - E.g. big endian to little endian
  - E.g. Ascii to Unicode
- Interface
  - It depends…
- Protocol
  - Define data formats
  - Apply transformation rules
- Examples: none

Application Layer



- Service
  - Whatever you want :)
- Interface
  - Whatever you want :D
- Protocol
  - Whatever you want ;)
- Examples: turn on your smartphone and look at the list of apps

Hybrid model – 5 layers

Each layer provides a service to the layer above

	layer	service provided
L5	**Application**	*high level network access*
L4	**Transport**	*end-to-end delivery (reliable or not)*
L3	**Network**	*global best-effort delivery*
L2	**Link**	*local best-effort delivery*
L1	**Physical**	*physical transfer of bits*

Each layer provides a service to the layer above by using the services of the layer directly below it

Applications

...built on...

Reliable (or unreliable) transport

...built on...

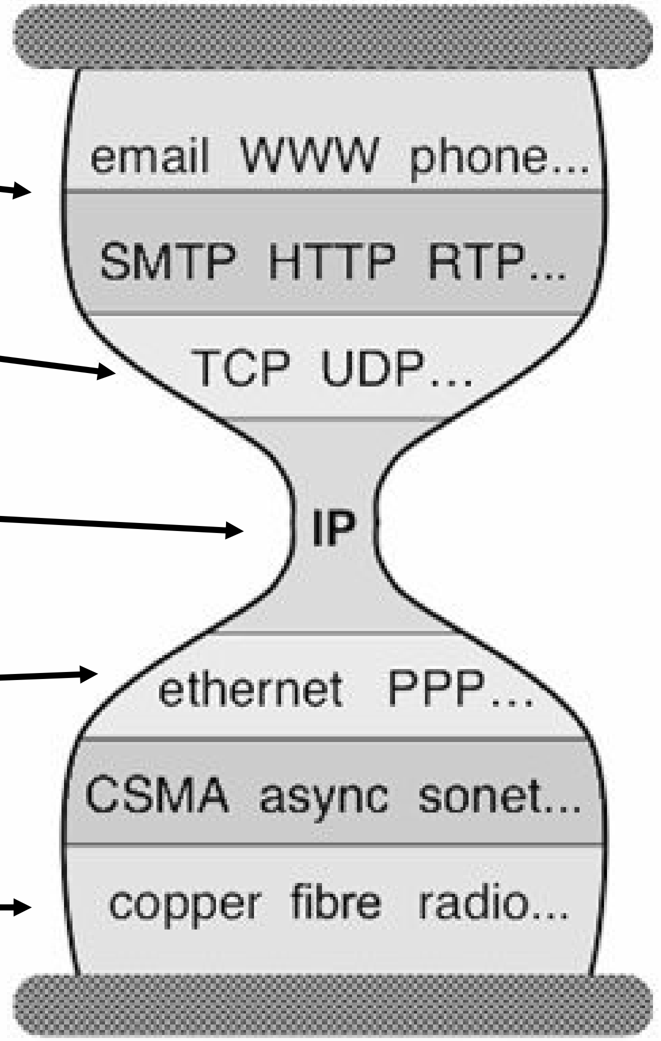
Best-effort global packet delivery

...built on...

Best-effort local packet delivery

...built on...

Physical transfer of bits



Each layer has a unit of data (aka protocol data unit)

	layer	role (PDU)
L5	**Application**	exchanges messages between processes
L4	**Transport**	transports segments between end-systems
L3	**Network**	moves packets around the network
L2	**Link**	moves frames across a link
L1	**Physical**	moves bits across a physical medium

Each layer (except for L3) is implemented with different protocols

	layer	protocols
L5	**Application**	HTTP, SMTP, FTP, SIP, …
L4	**Transport**	TCP, UDP, SCTP
L3	**Network**	IP
L2	**Link**	Ethernet, Wifi, ADSL, WiMAX, LTE, …
L1	**Physical**	Twisted pair, fiber, coaxial cable, …

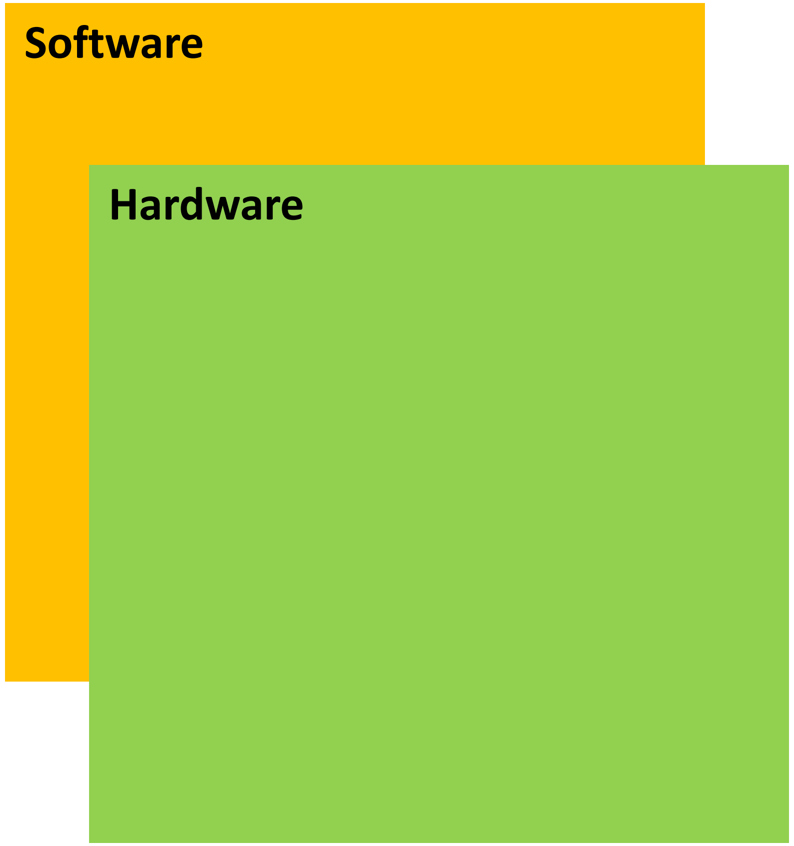
The Internet Protocol (IP) is the **glue** acting as a unifying network layer

	layer	protocols
L5	**Application**	HTTP, SMTP, FTP, SIP, …
L4	**Transport**	TCP, UDP, SCTP
L3	**Network**	IP
L2	**Link**	Ethernet, Wifi, ADSL, WiMAX, LTE, …
L1	**Physical**	Twisted pair, fiber, coaxial cable, …

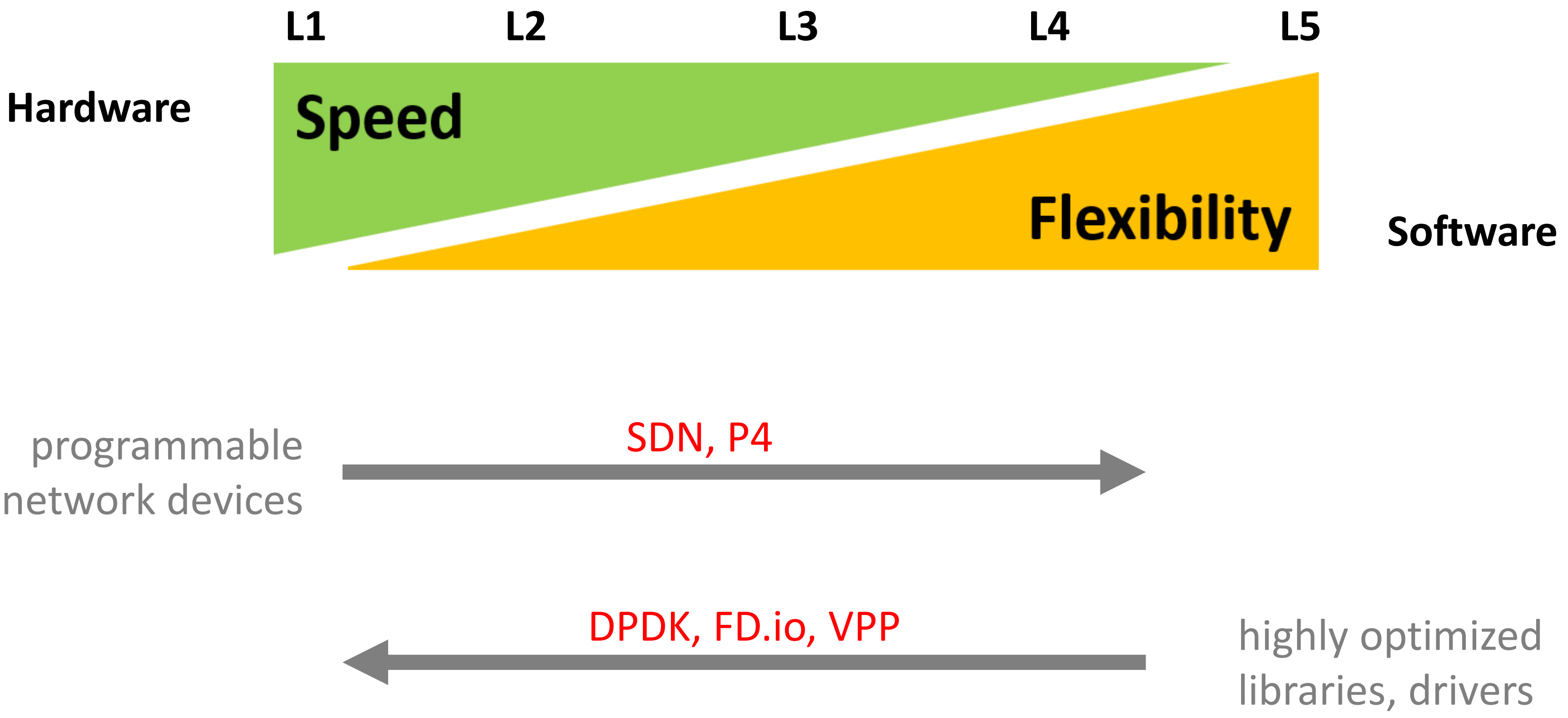
Each layer is implemented with different protocols and technologies

	layer
L5	**Application**
L4	**Transport**
L3	**Network**
L2	**Link**
L1	**Physical**

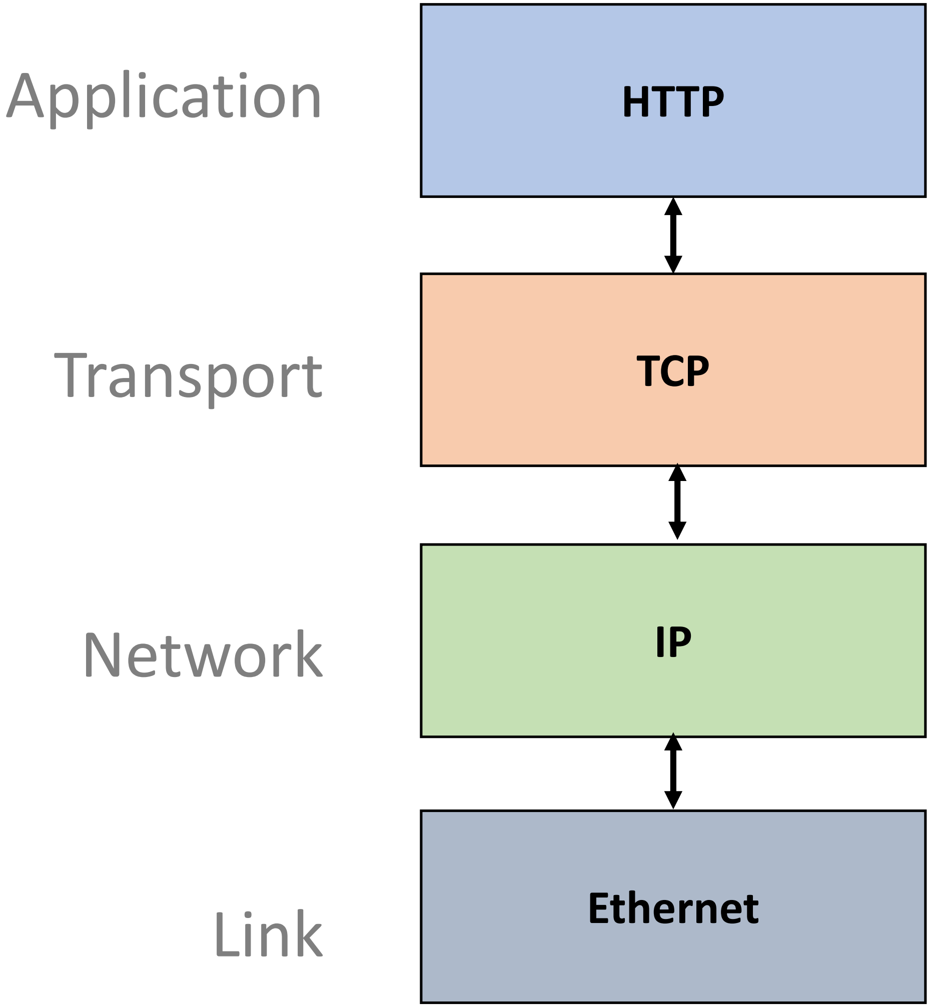
technology

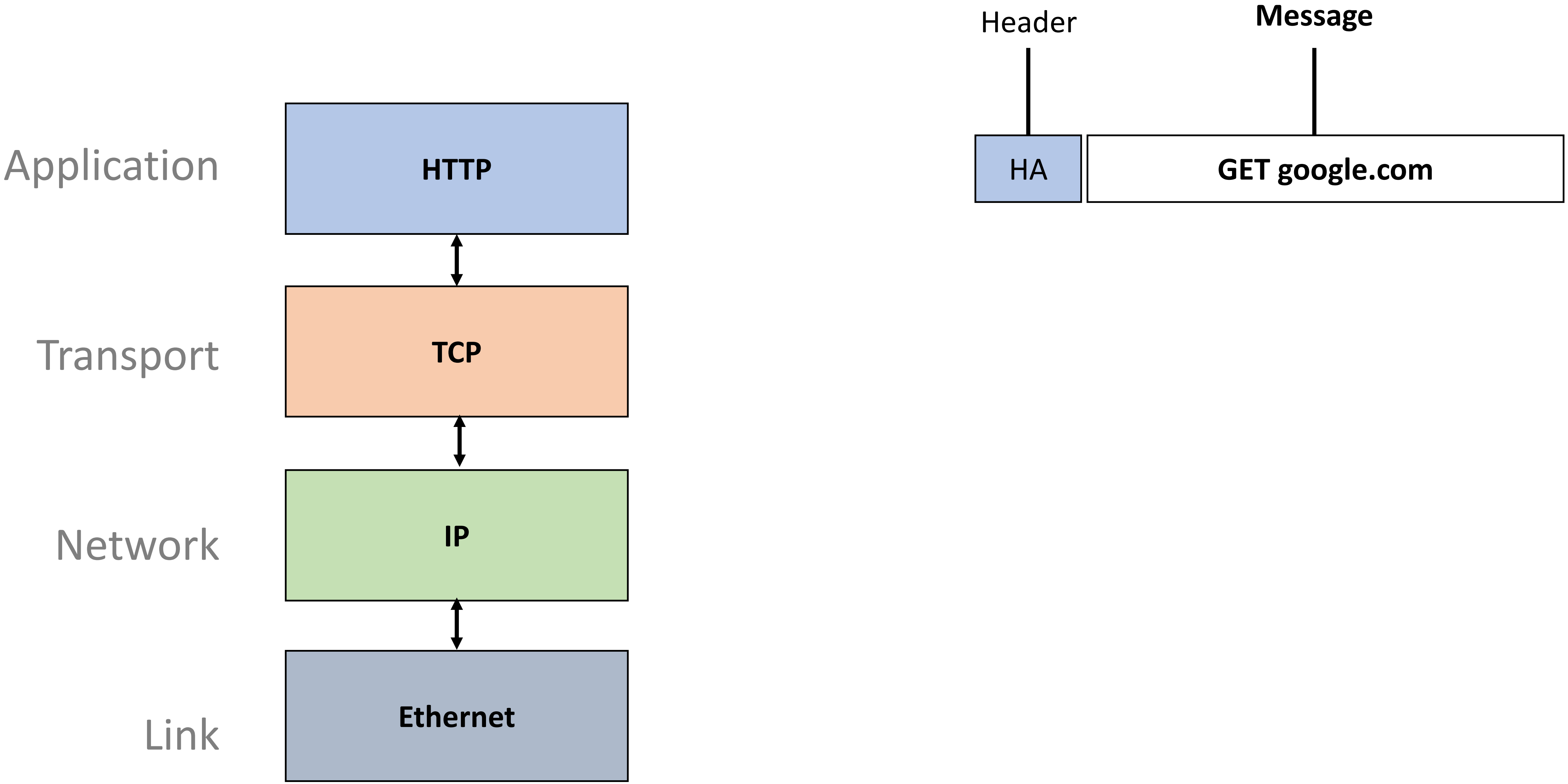


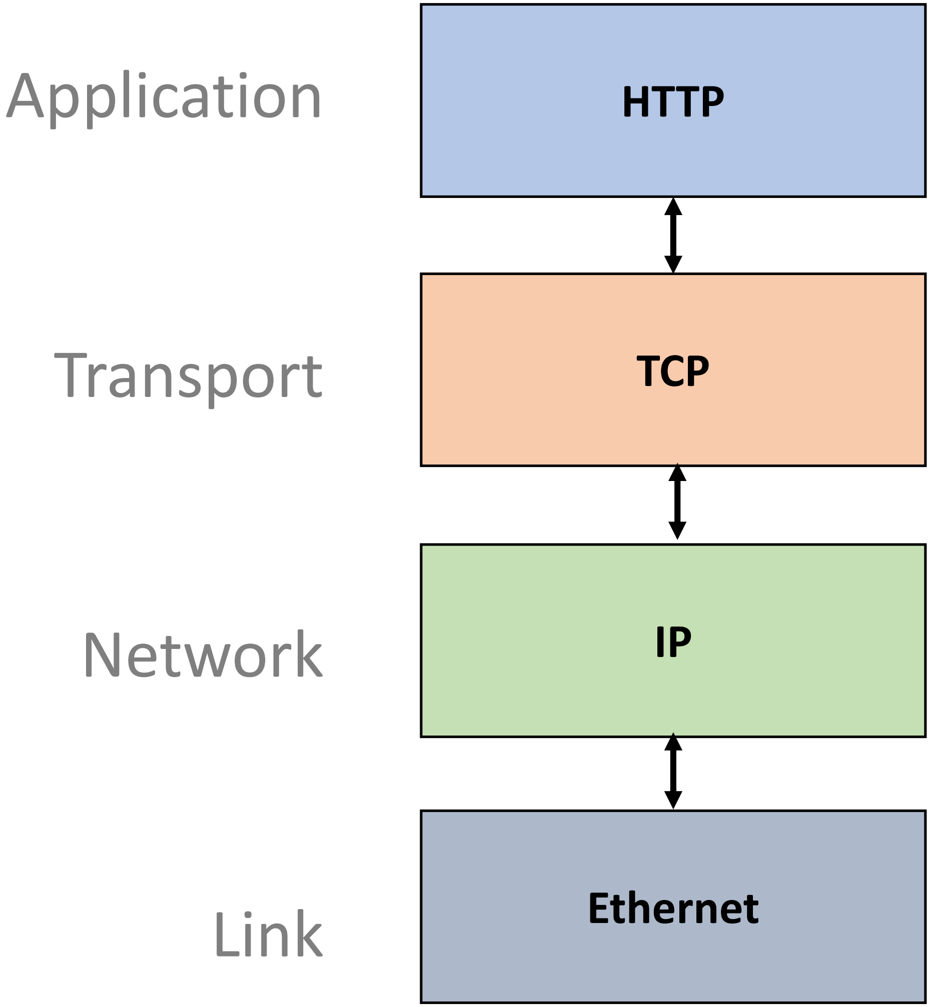
Software and hardware advancements

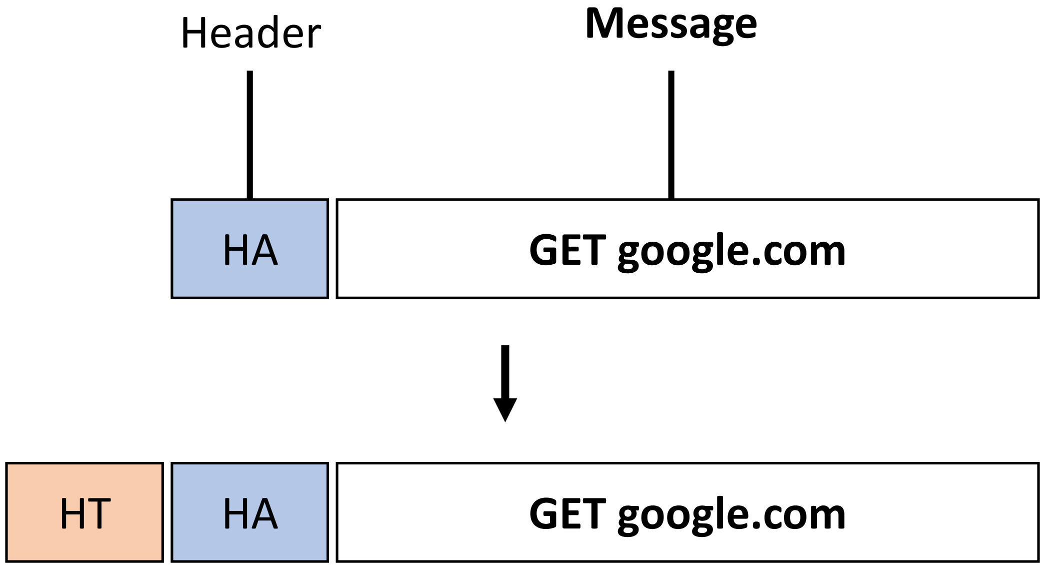


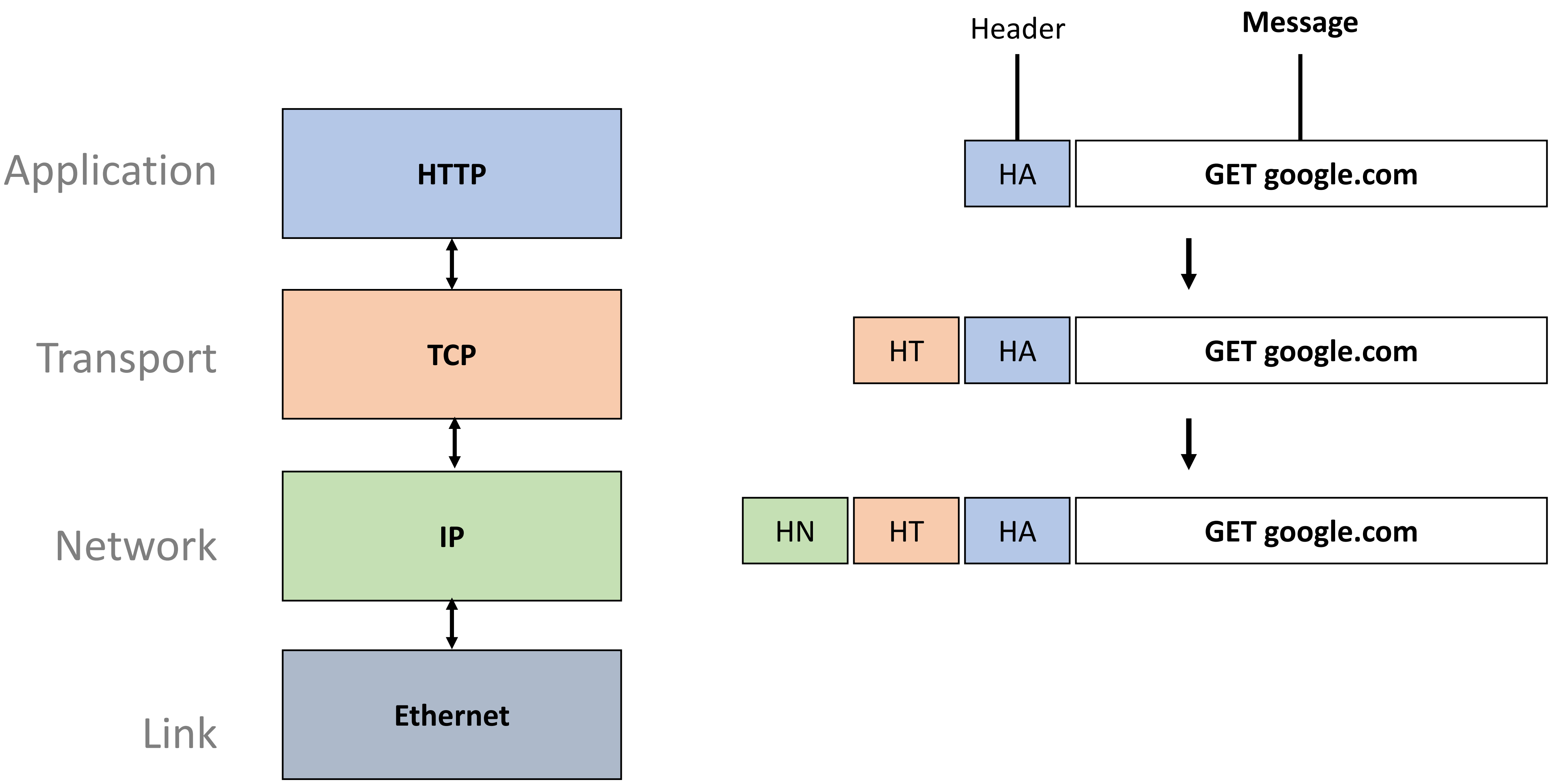
Each layer takes messages from the layer above, and *encapsulates* with its own header and/or trailer

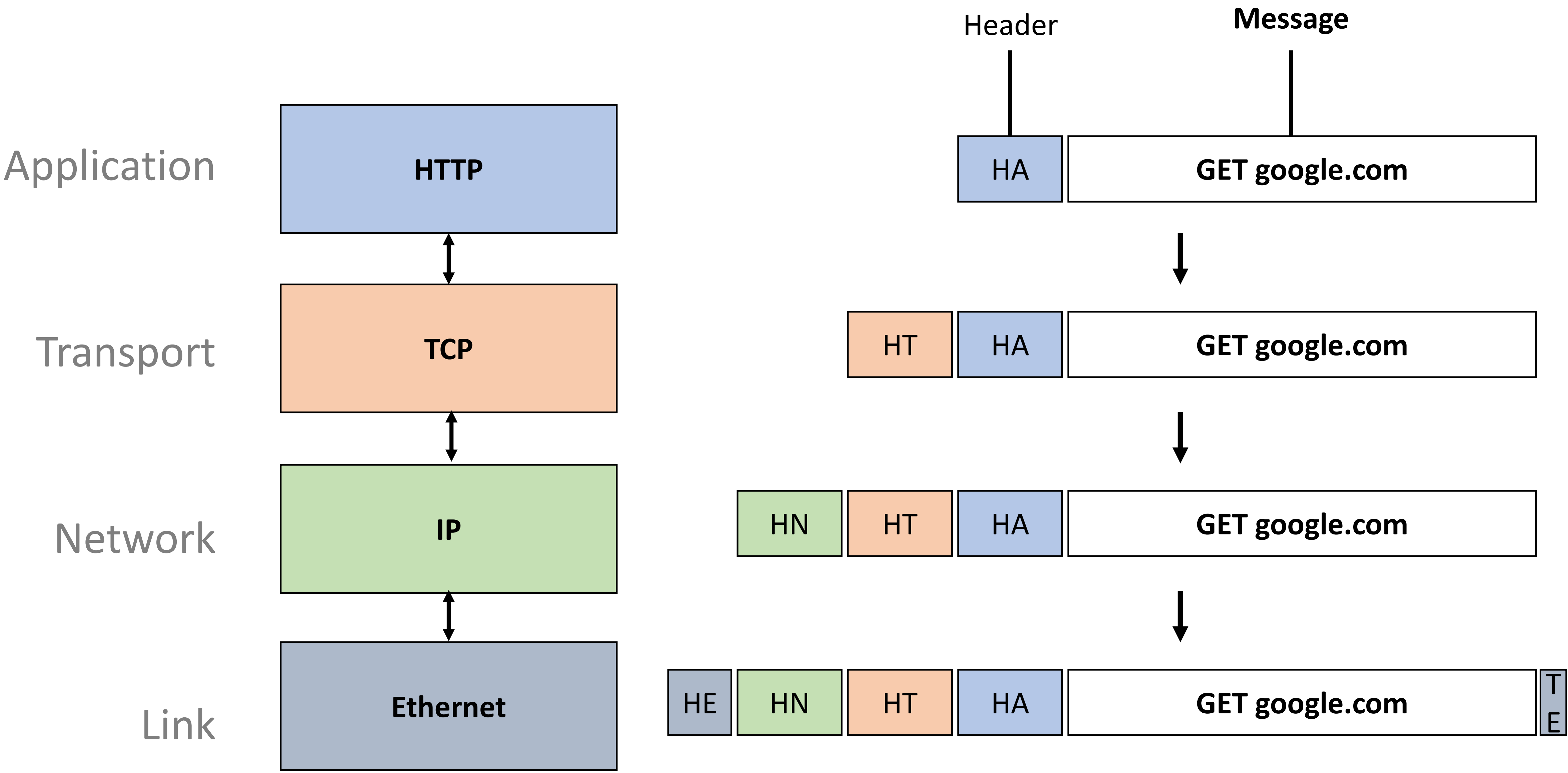



Application
Transport
Network
Link
HTTP
TCP
IP
Ethernet
Header
Message
HA
GET google.com

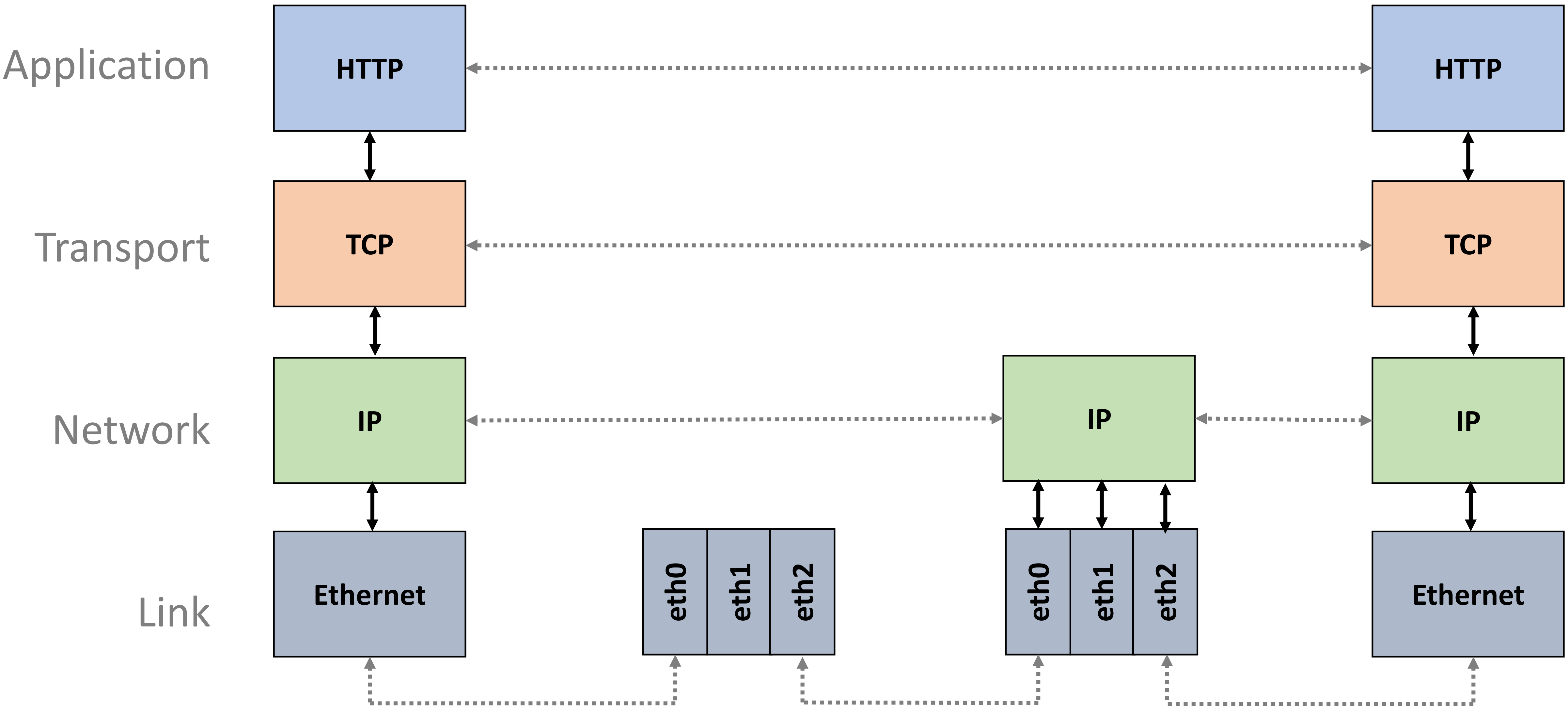

Application
HTTP
Transport
TCP
Network
IP
Link
Ethernet


Header
Message
HA
GET google.com
HT
HA
GET google.com

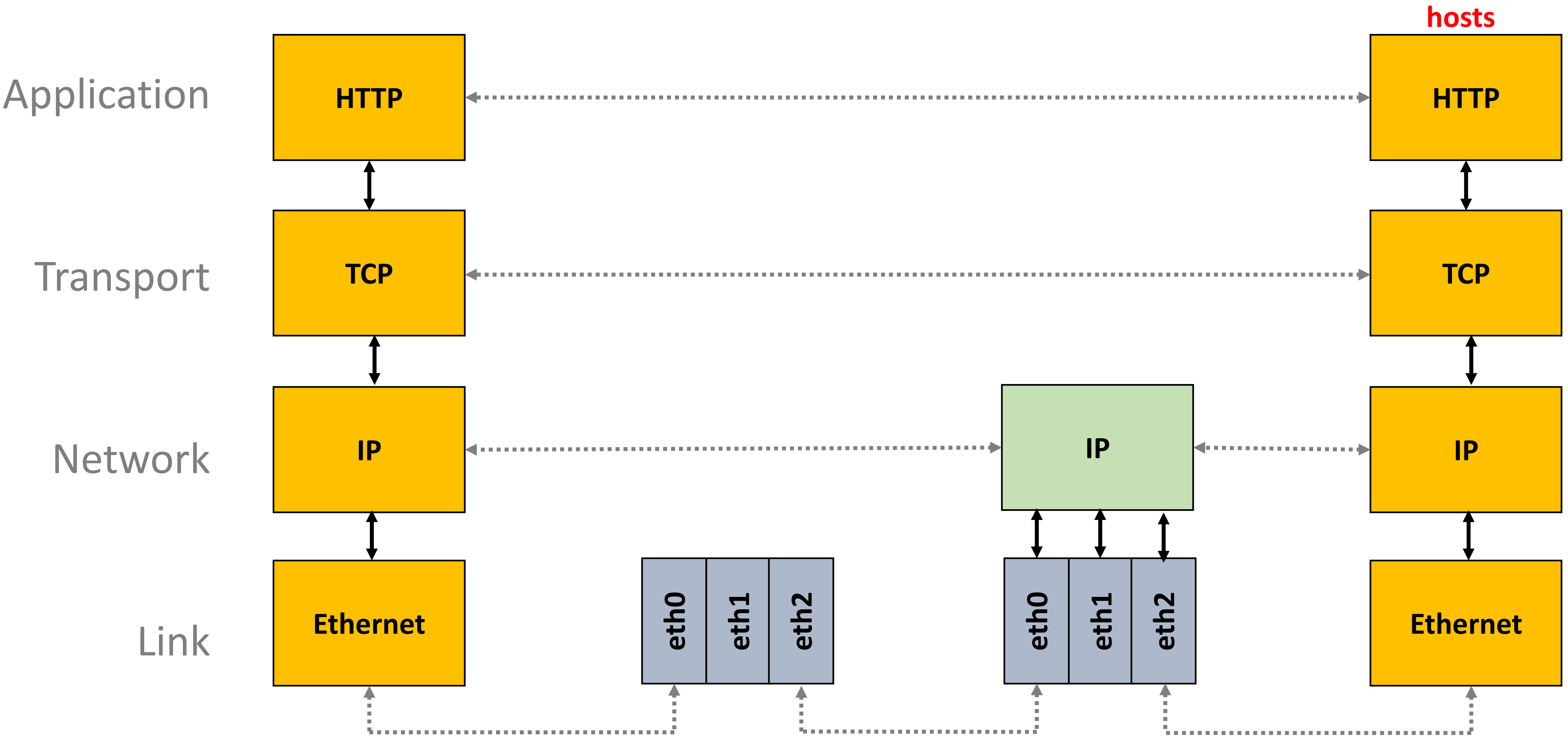

Application
Transport
Network
Link
HTTP
TCP
IP
Ethernet
Header
Message
HA
GET google.com
HT
HA
GET google.com
HN
HT
HA
GET google.com


Application
Transport
Network
Link
HTTP
TCP
IP
Ethernet
Header
Message
HA
GET google.com
HT
HA
GET google.com
HN
HT
HA
GET google.com
HE
HN
HT
HA
GET google.com
TE

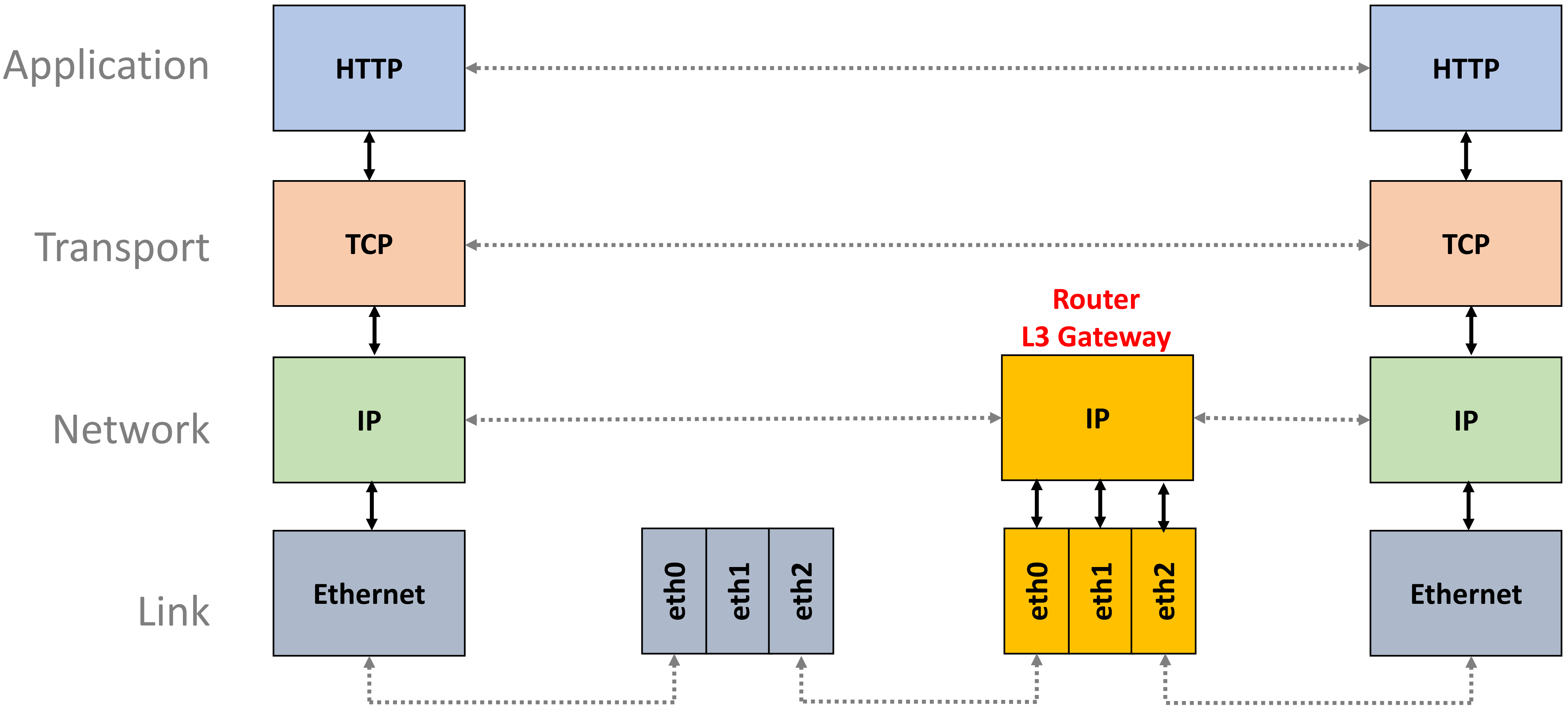
In practice, layers are **distributed** on every network device



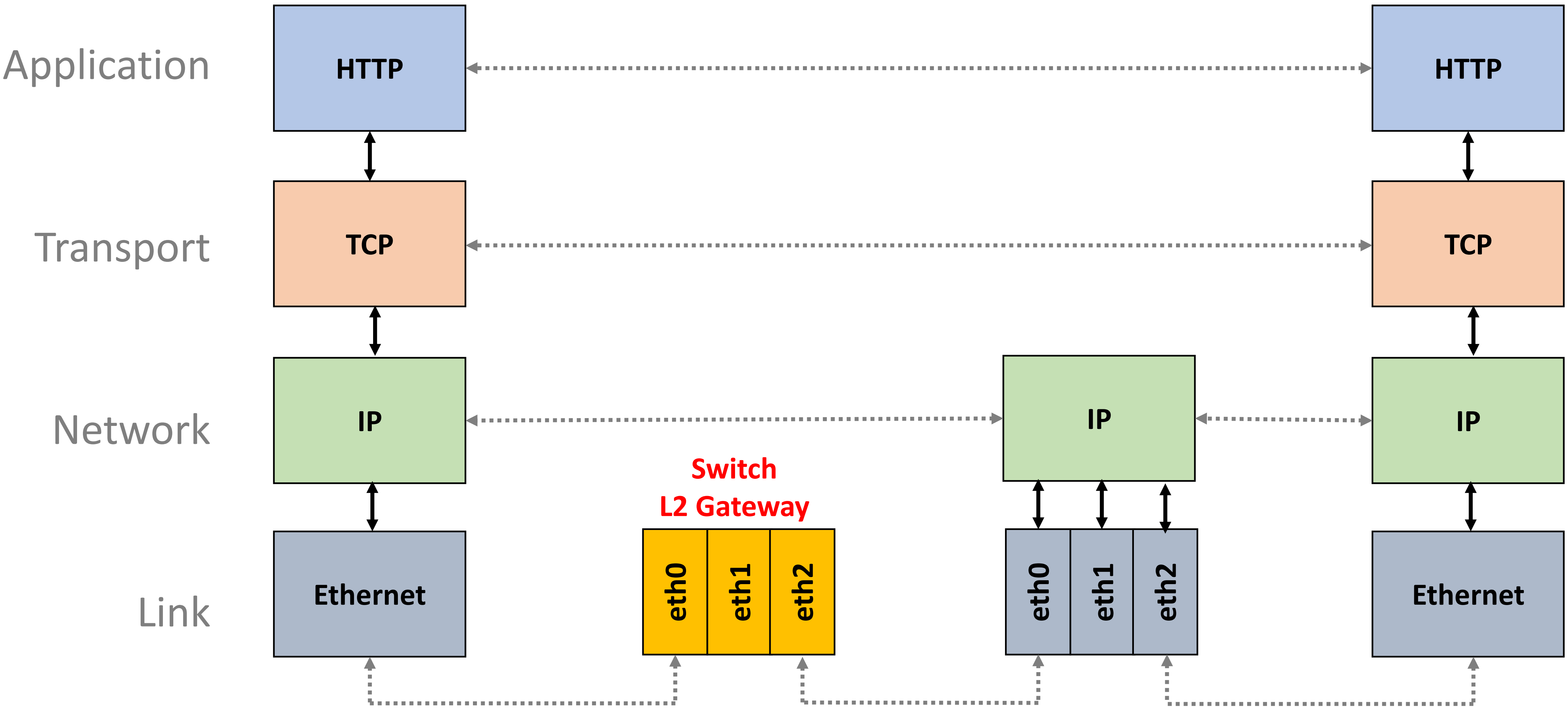
Since when bits arrive they must make it to the application, all the layers exist on a host



Routers act as L3 gateway
as such they implement L2 and L3



Switches act as L2 gateway as such they only implement L2



Overview

How do we characterize the network?

A network *connection* is characterized by its delay, loss rate and throughput

How long does it take for a packet to reach the destination

What fraction of packets sent to a destination are dropped?

At what rate is the destination receiving data from the source?

Delay

Each packet suffers from several types of delays at *each node* along the path

	transmission delay	due to **link properties**
+	**propagation** delay	
+	**processing** delay	due to **traffic mix & switch internals**
+	**queueing** delay	
=	**total** delay	

Each packet suffers from several types of delays at *each node* along the path

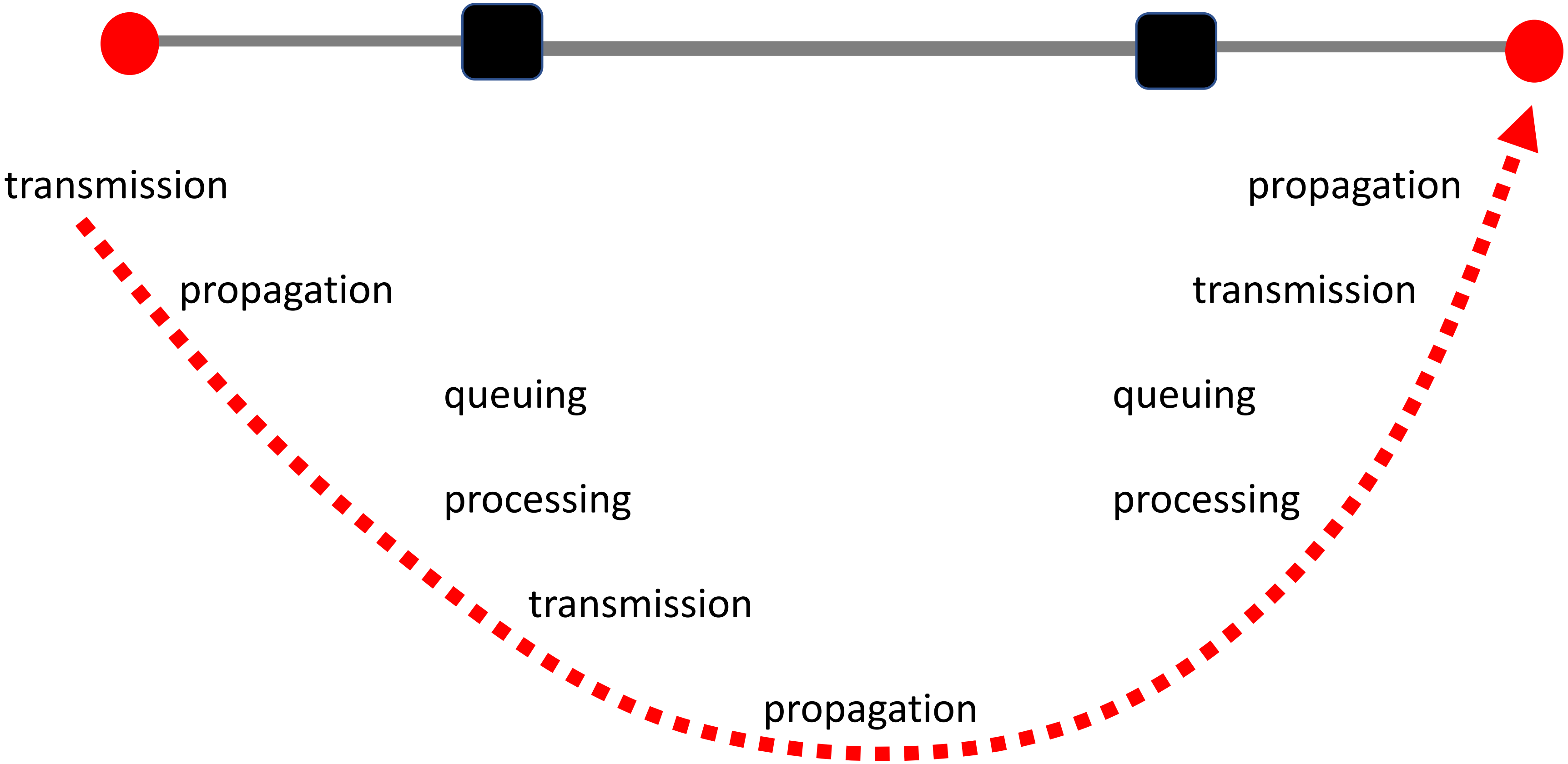
transmission delay

\+ **propagation** delay

~~+ **processing** delay~~ tend to be tiny

\+ **queueing** delay

= **total** delay


transmission
propagation
queuing
processing
transmission
propagation
queuing
processing
transmission
propagation

The transmission delay is the amount of time required to push all of the bits onto the link

$$\text{Transmission delay } \color{red}{[\text{sec}]} \quad = \quad \frac{\text{packet size}}{\text{link bandwidth}} \quad \begin{matrix}\color{red}{[\#\text{bits}]} \\ \color{red}{[\#\text{bits/sec}]}\end{matrix}$$

$$\color{red}{\text{Example}} \quad = \quad \frac{1000 \text{ bits}}{100 \text{ Mpbs}} \quad = \quad 10\ \mu\text{sec}$$

The propagation delay is the amount of time required for a bit to travel to the end of the link

$$\text{Propagation delay} = \frac{\text{link length}}{\text{signal propagation speed}} \quad \frac{[m]}{[m/sec]}$$

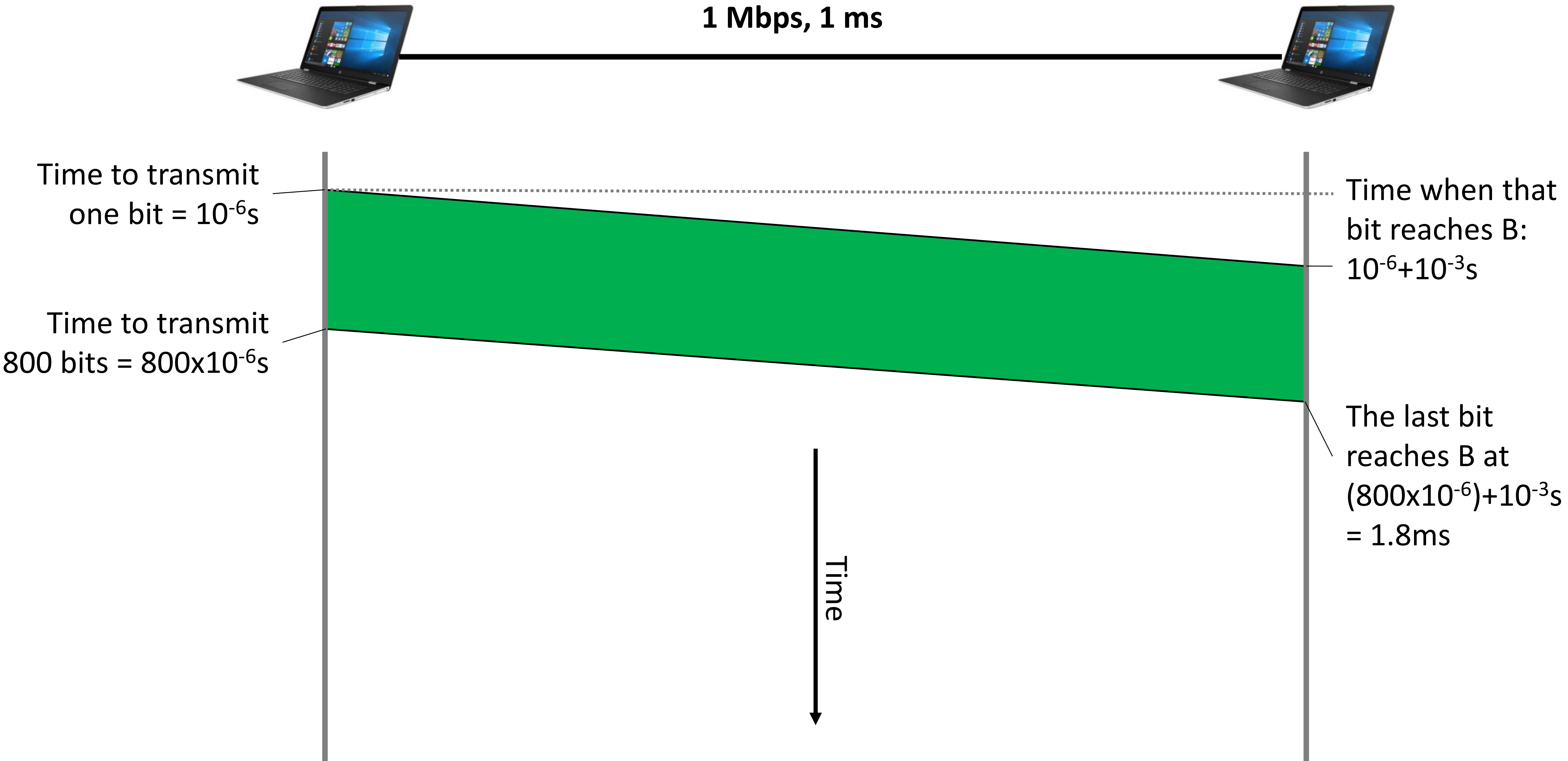
[sec]

Example

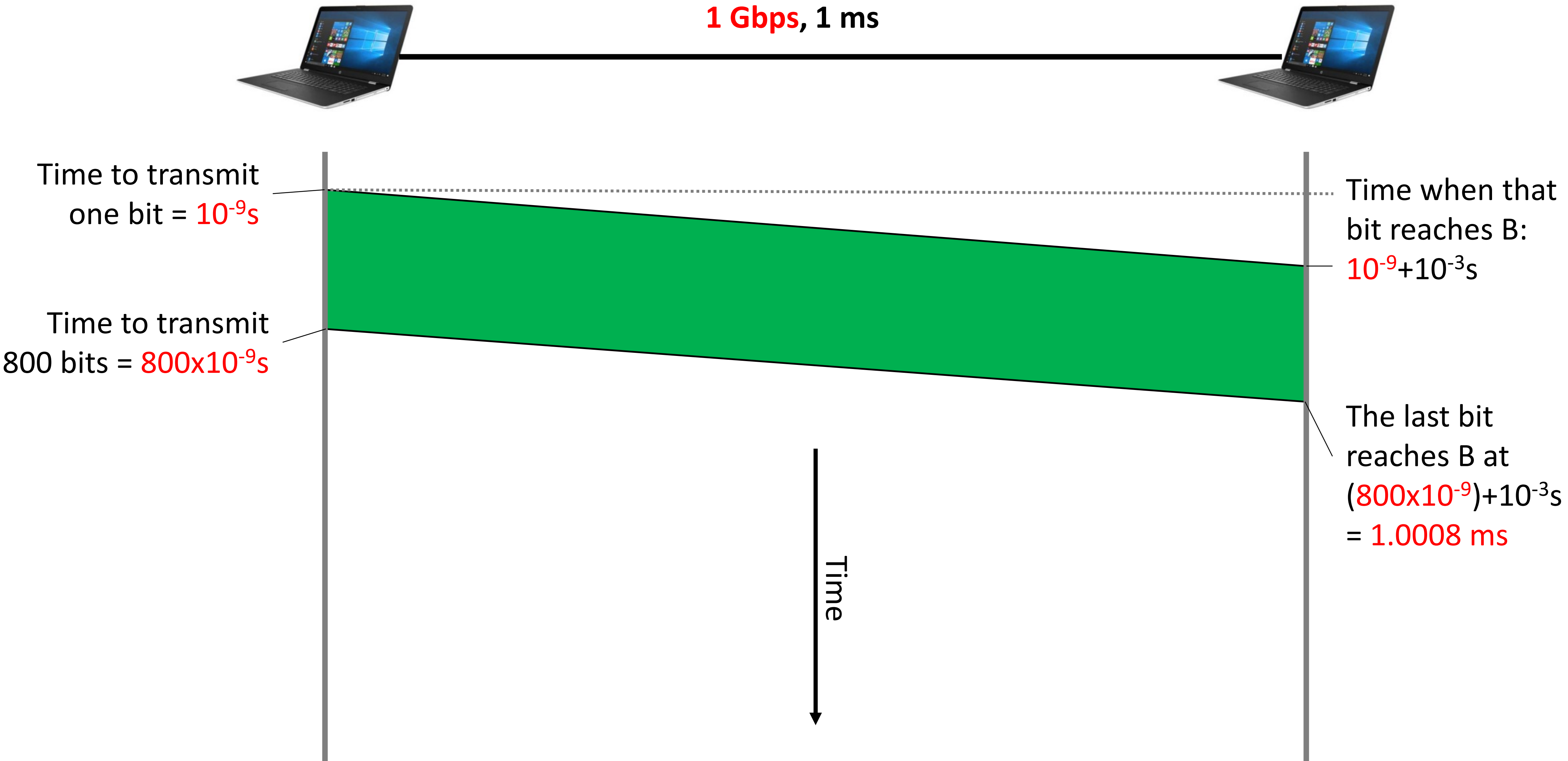
$$= \frac{30000\ m}{2 \times 10^8\ m/sec} = 150\ \mu sec$$

(speed of light in fiber)

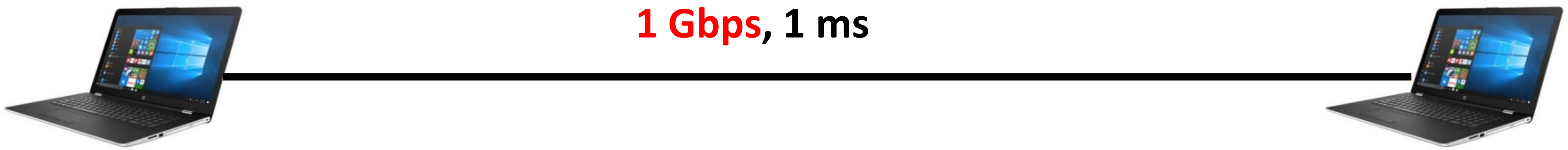
How long does it take to exchange 100 Bytes packet?

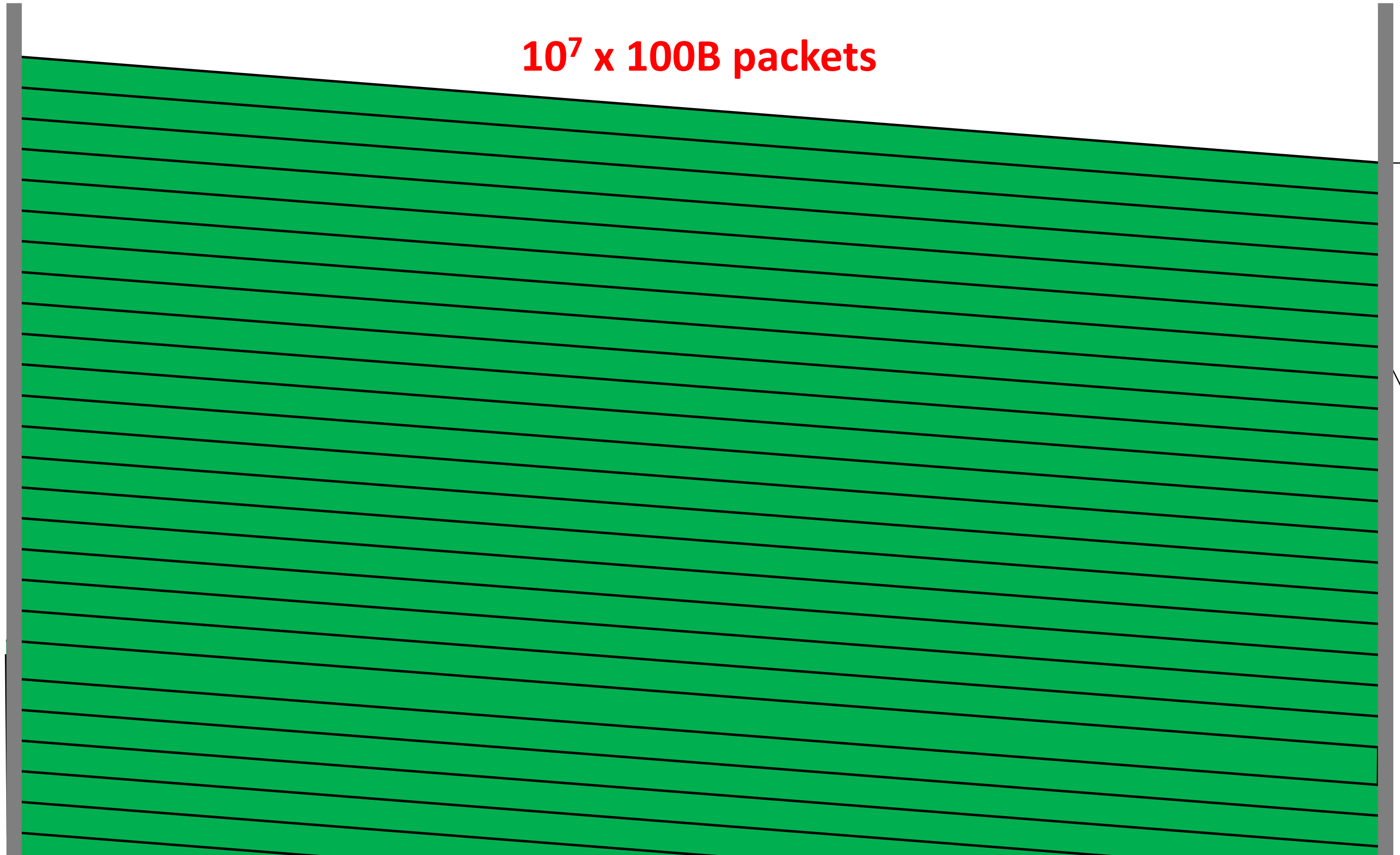


If we have a 1 Gbps link, the total time decreases to 1.0008 ms



If we now exchange a 1GB file split in 100B packets





Time when that bit reaches B: $10^{-9}+10^{-3}$s

The last bit reaches B at $(10^7 \times 800 \times 10^{-9})+10^{-3}$s = 8001 ms

Different transmission characteristics imply different tradeoffs in terms of which delay dominates

10^7x100B pkts	1Gbps link	***transmission delay dominates***
1x100B pkt	1Gbps link	***propagation delay dominates***
1x100B pkt	1Mbps link	***both matter***

In the Internet, we cannot know in advance which one matter!

The queuing delay is the amount of time a packet **waits** (in a buffer) to be transmitted on a link

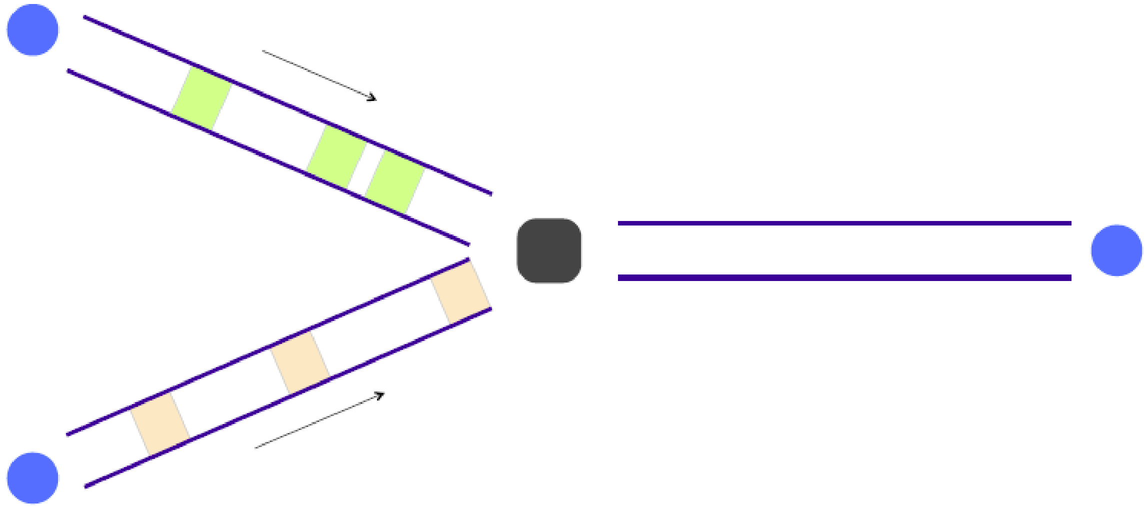
Queuing delay is the hardest to evaluate

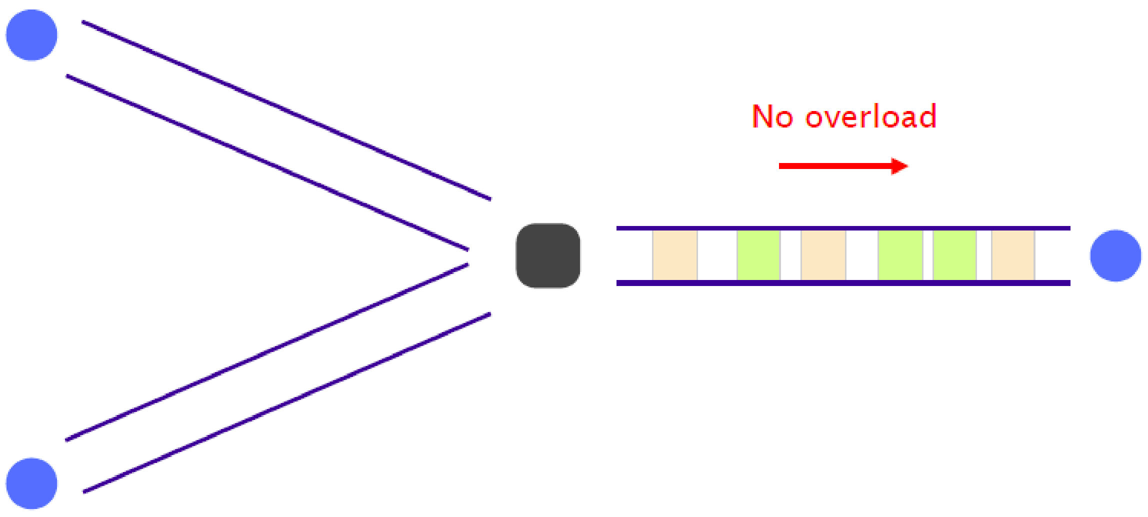
as it varies from packet to packet

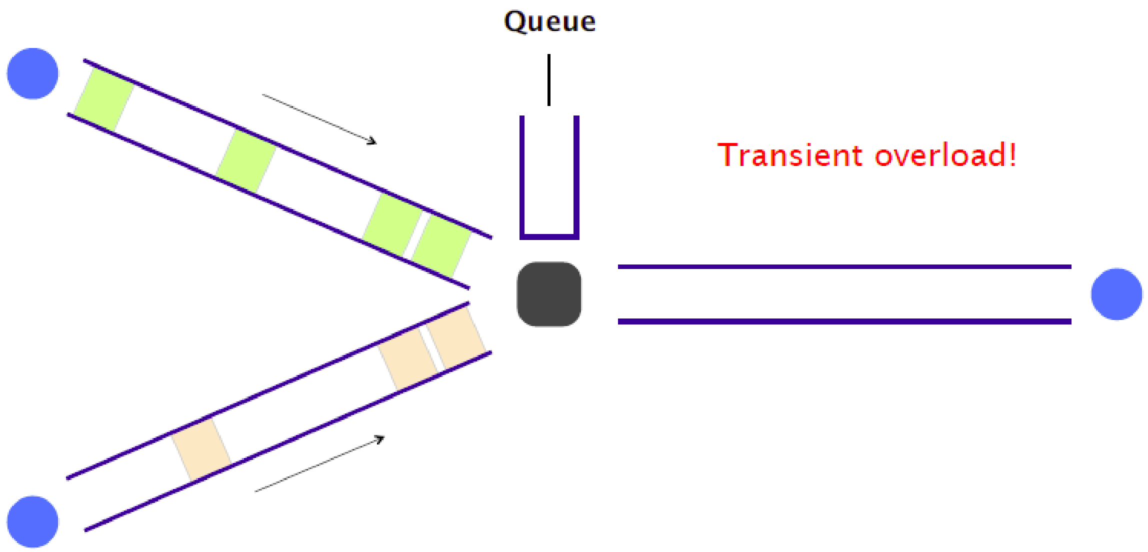
It is characterized with statistical measures

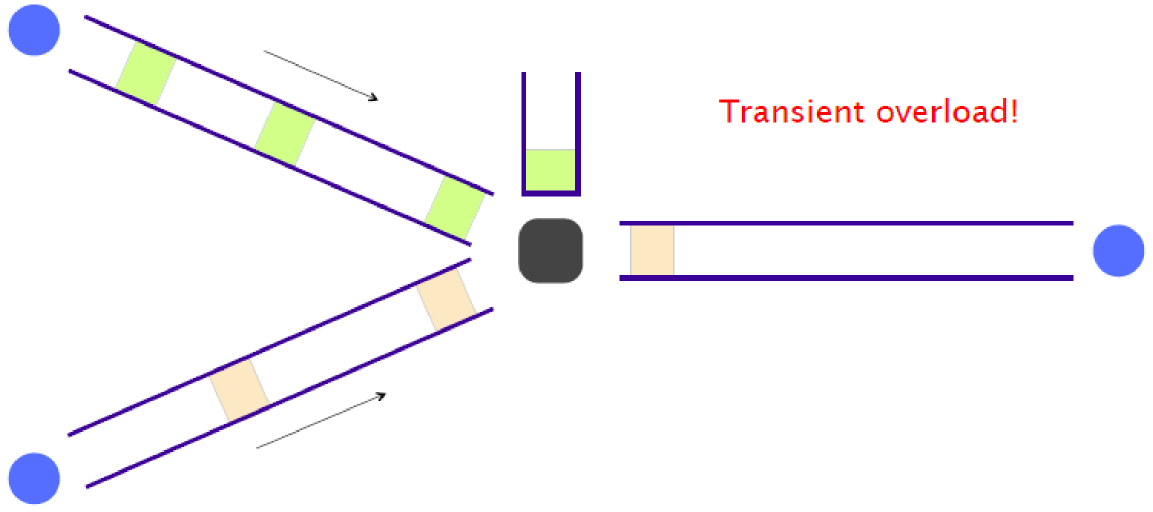
e.g., average delay & variance, probability of exceeding *x*

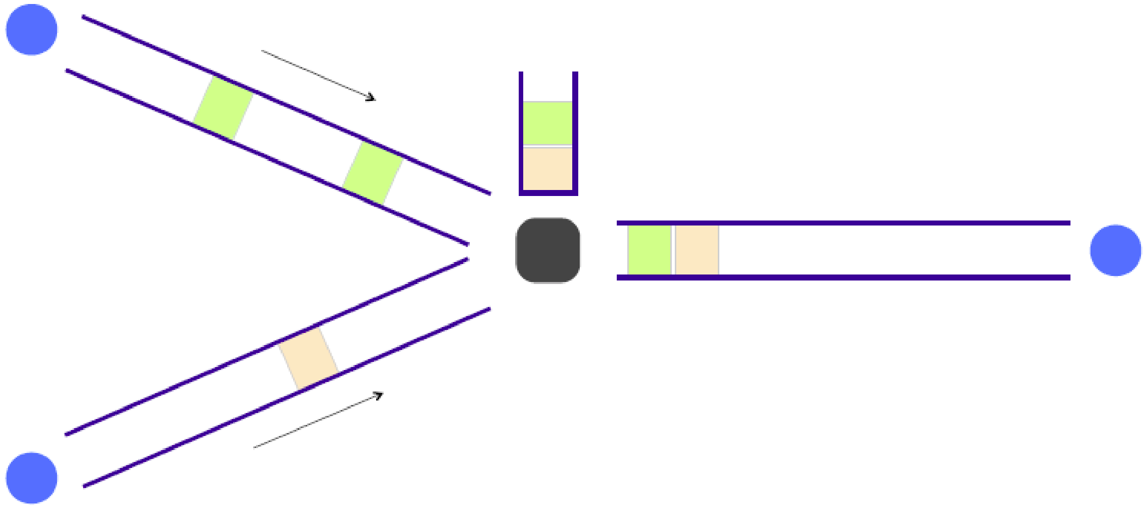
Queuing delay depends on the traffic pattern

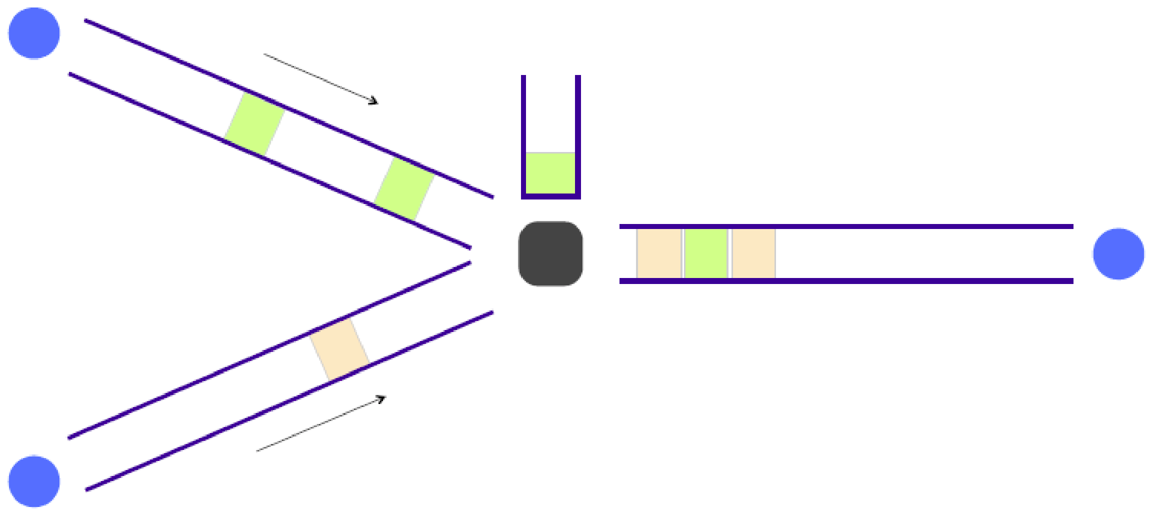


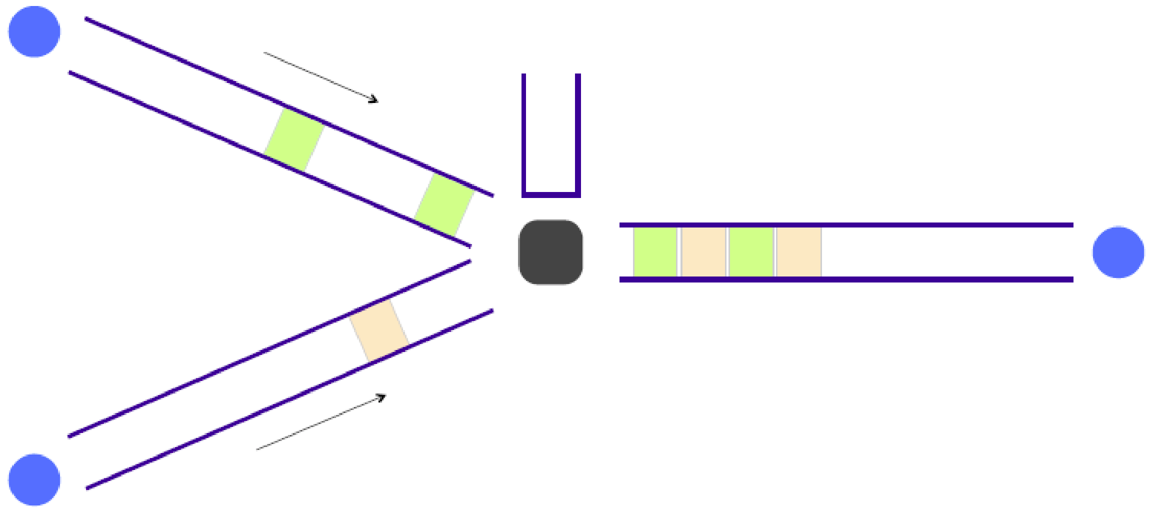

No overload


Queue
Transient overload!

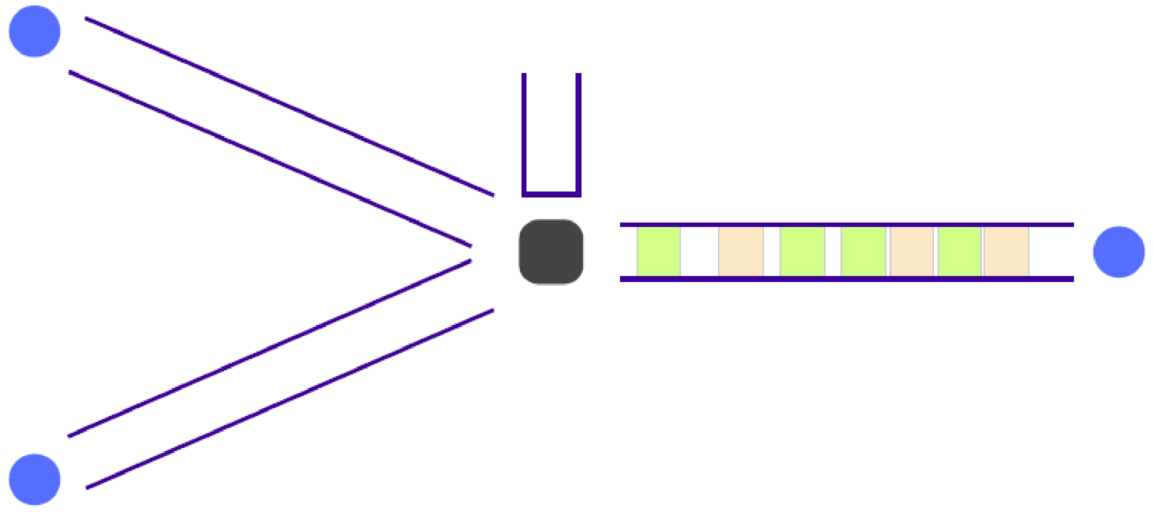

Transient overload!







Queues absorb transient bursts, but introduce queueing delays



The time a packet has to sit in a buffer before being processed depends on the traffic pattern

Queueing delay depends on:

arrival rate at the queue

transmission rate of the outgoing link

traffic **burstiness**

• **average packet arrival rate**	a	[packet/sec]
• **transmission rate of outgoing link**	R	[bit/sec]
• **fixed packets length**	L	[bit]
• **average bits arrival rate**	La	[bit/sec]
• **traffic intensity**	La/R	

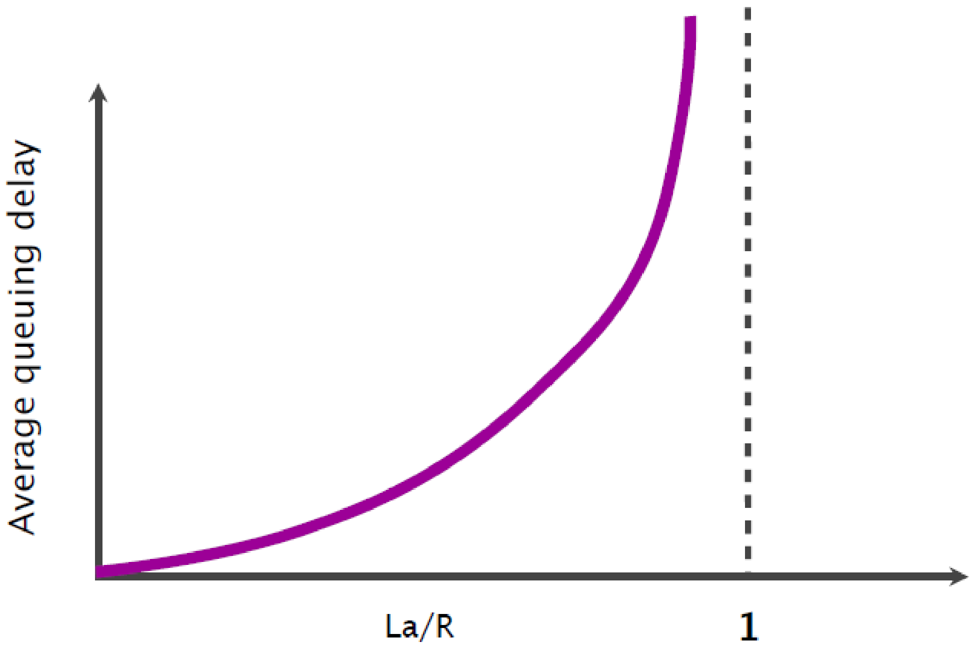
When the traffic intensity is >1, the queue will increase without bound, and so does the queuing delay

Golden rule

Design your queuing system,

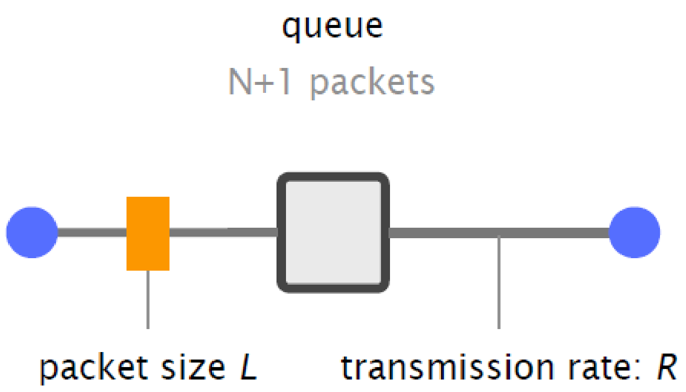
so that it operates far from that point

When the traffic intensity is <=1, queueing delay depends on the burst size



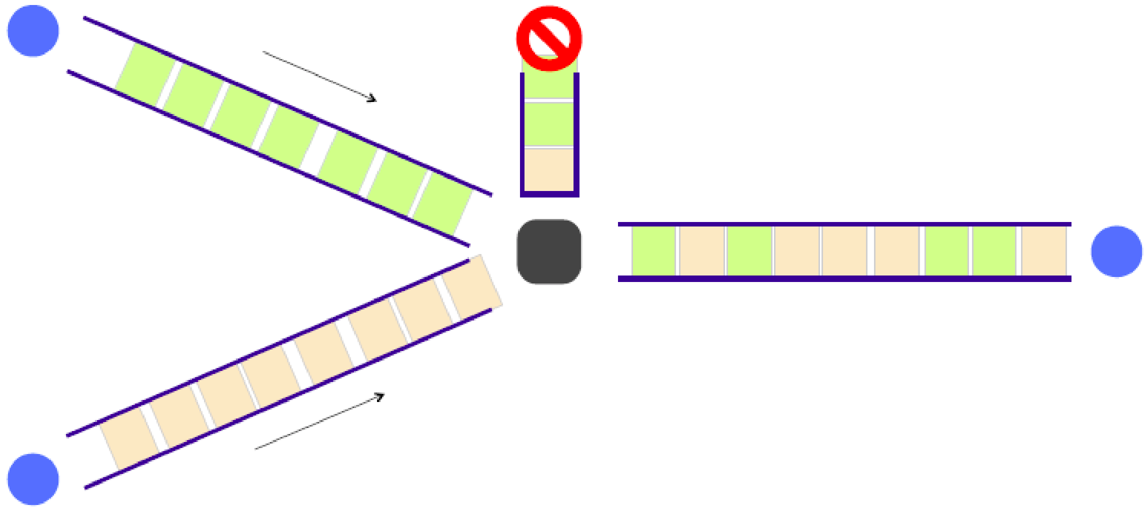
Loss

In practice, queues are not infinite.
There is an upper bound on queuing delay.



queuing delay upper bound: N*L/R

If the queue is persistently overloaded, it will eventually drop packets (loss)

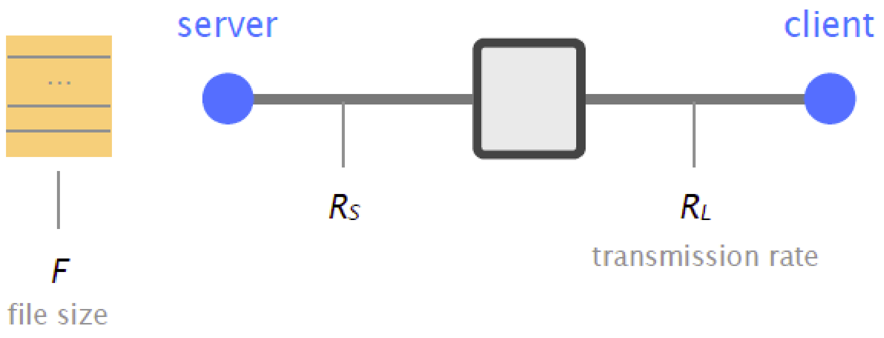


Throughput

The throughput is the instantaneous rate at which a host receives data

$$\underset{[\#\text{bits/sec}]}{\text{Average throughput}} = \frac{\text{data size} \quad [\#\text{bits}]}{\text{transfer time} \quad [\text{sec}]}$$

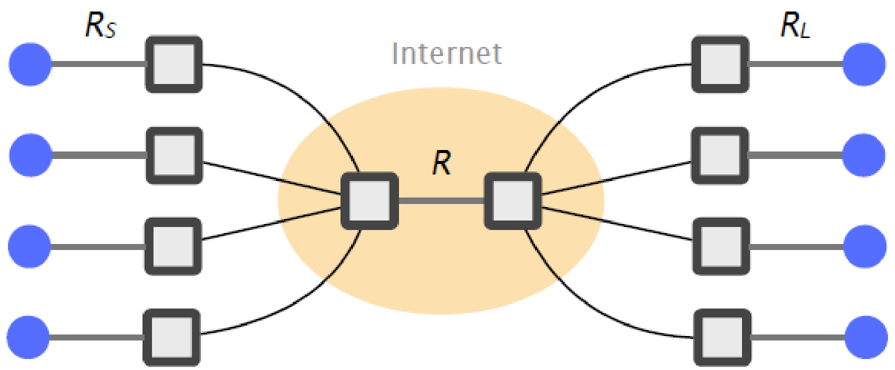
To compute throughput, one has to consider the bottleneck link



Average throughput $\min(R_S, R_L)$

= transmission rate of the bottleneck link

To compute throughput, one has to consider the bottleneck link… and the intervening traffic

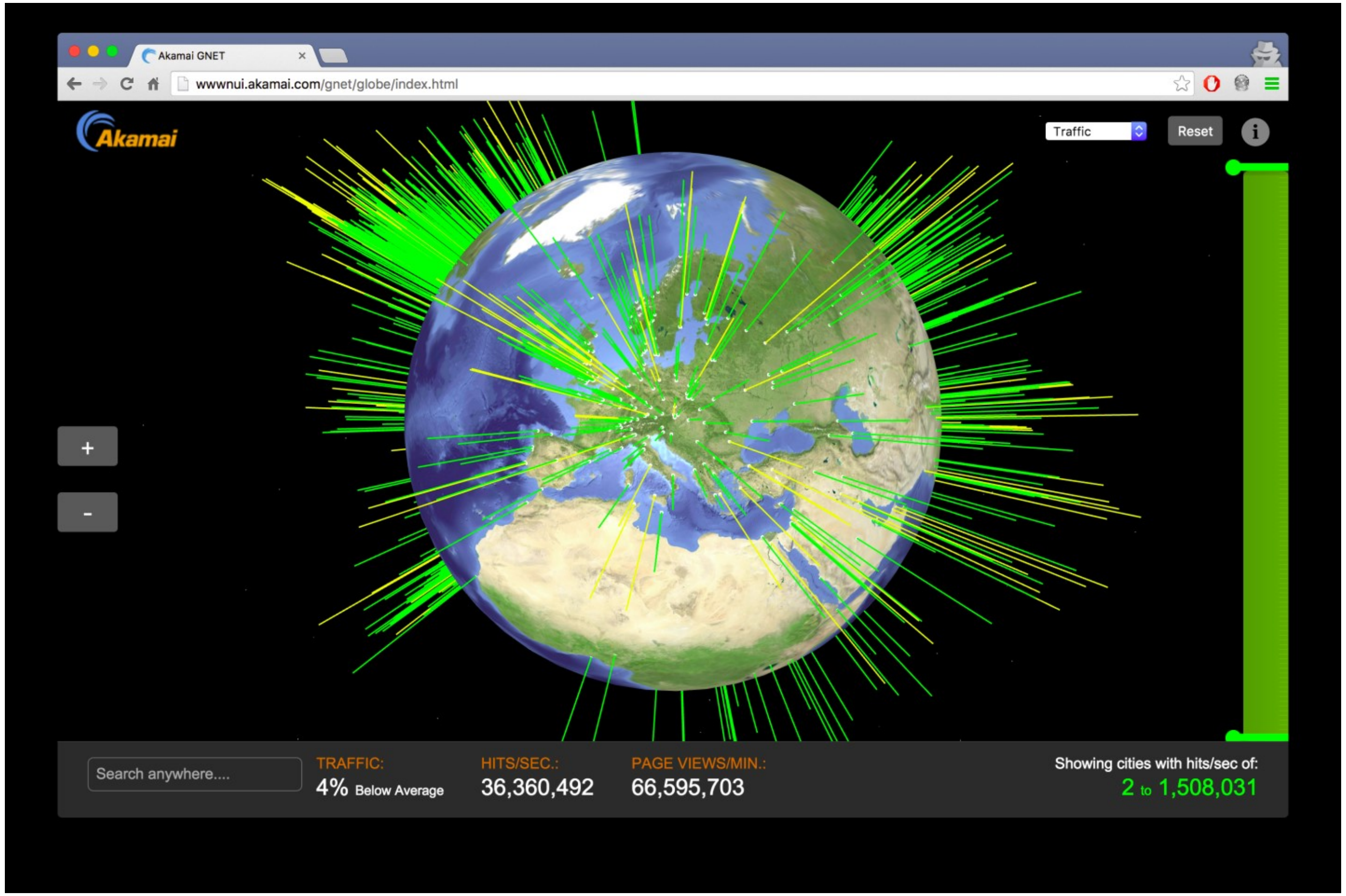


if $4*\min(R_S,R_L) > R$ the bottleneck is now in the core, providing each download R/4 of throughput

As technology improves, throughput increase & delays are getting lower except for propagation

(speed of light)

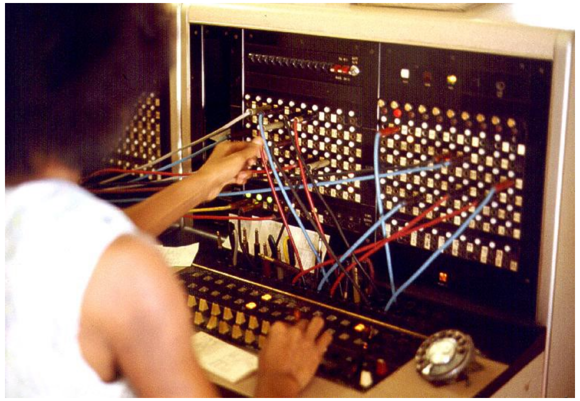
Because of propagation delays,
Content Delivery Networks move content closer to you



* http://wwwnui.akamai.com/gnet/globe/index.html

Brief history of Internet

The Internet history starts in the late 50's…



Phone networks = the communication network

fully circuit-switched

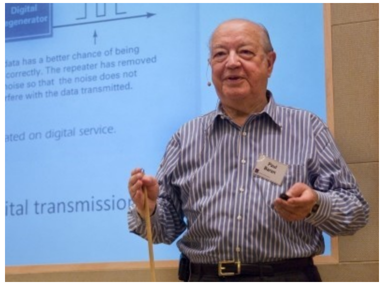
People wanted to use networks for other things

defense, computers, etc.

circuit-switching does not fit to these new requirements…

inefficient for bursty loads and not resilient

Three main questions



Paul Baran RAND

How can we design a more resilient network?

… led to the invention of packet switching

Leonard Kleinrock
UCLA

How can we design a more efficient network?

… also led to the invention of packet switching

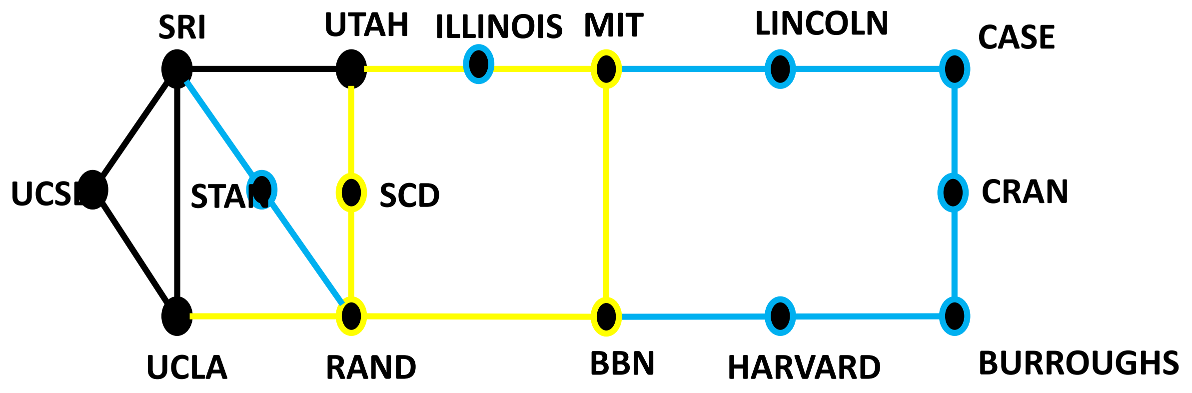
Vint Cerf & Bob Kahn
DARPA

How can we connect all these networks together?

… the invention of Internet as we know it

The 60’s was all about packet switching…

Advanced Research Projects Agency NETwork (ARPANET)



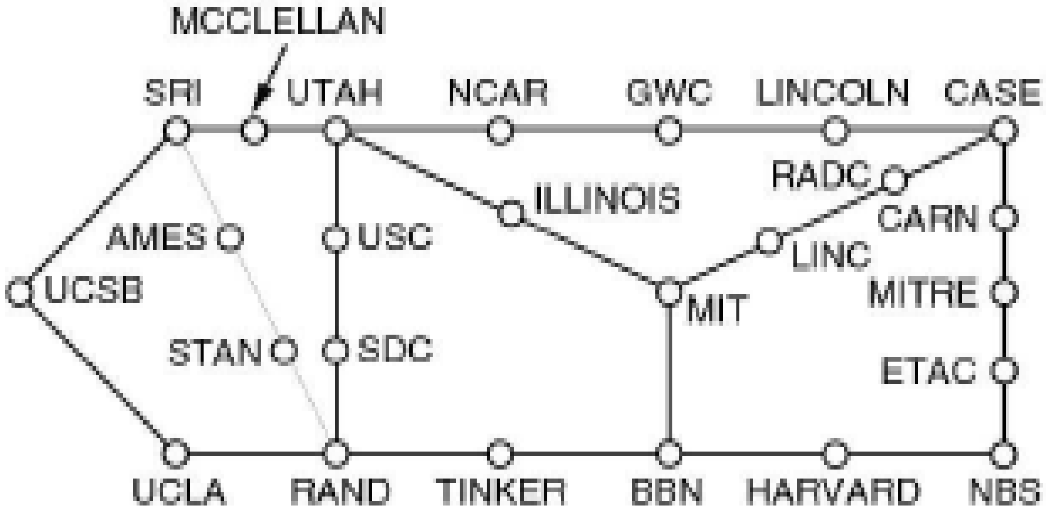
1969 december

1970 july

1971 march

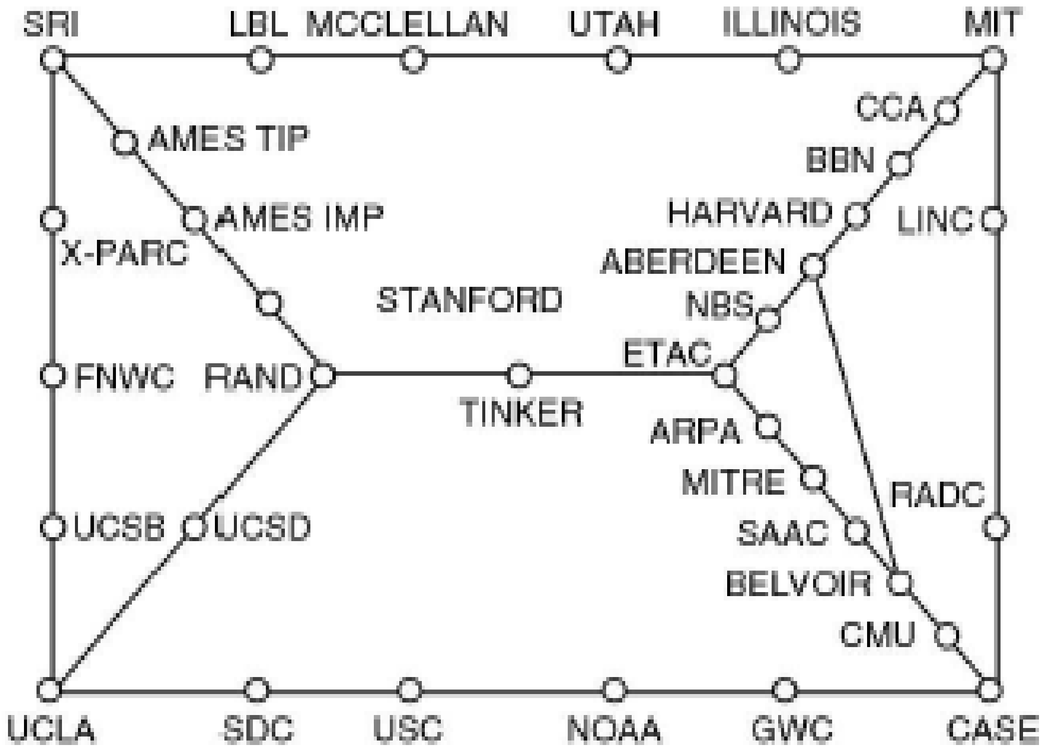
ARPANET

April 1972

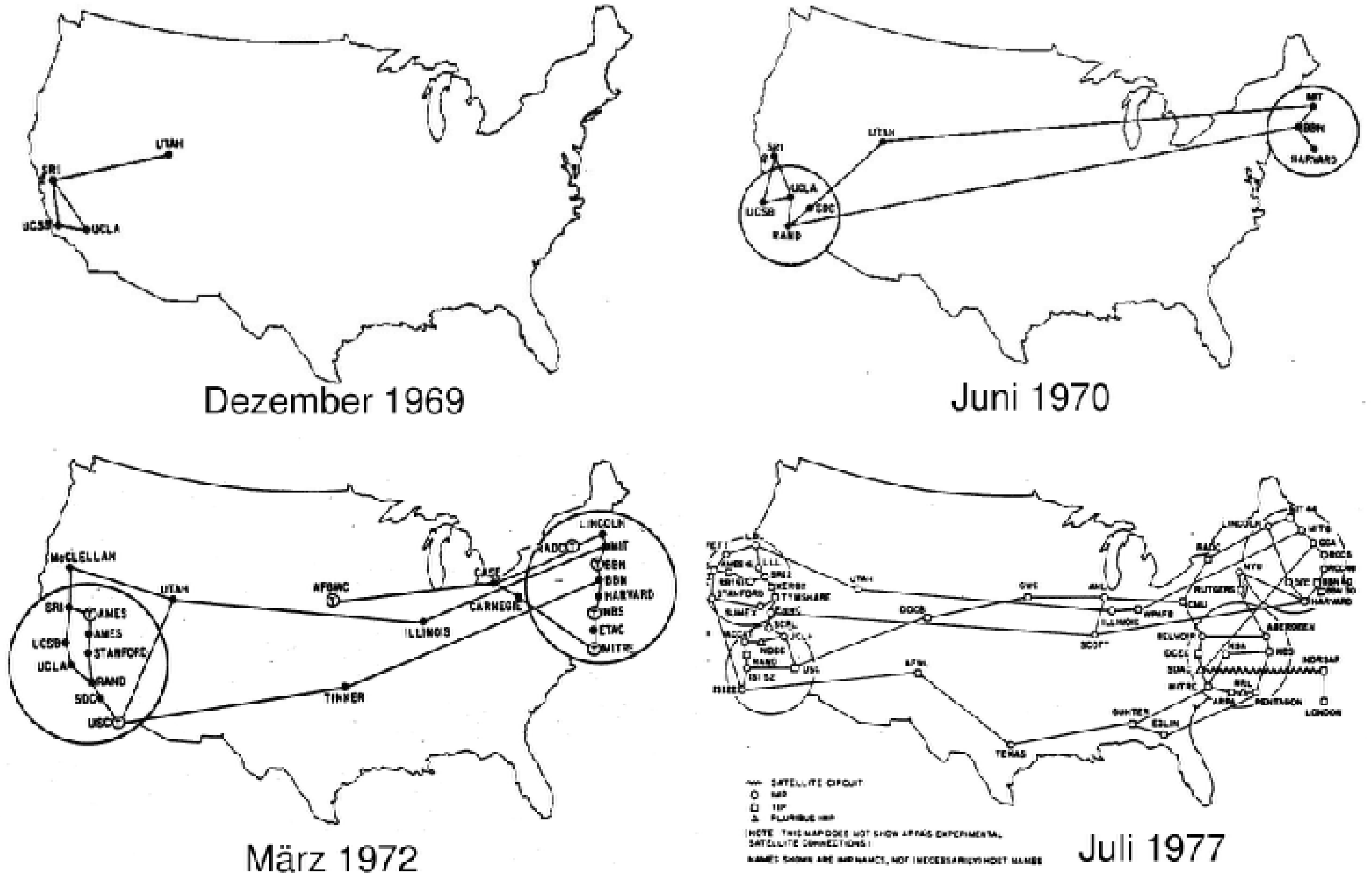


ARPANET

September 1972



ARPANET


Dezember 1969
UTAH
SRI
UCSB
UCLA
Juni 1970
UTAH
SRI
UCLA
UCSB
SDC
RAND
MIT
BBN
HARVARD
März 1972
Juli 1977

The first message over the Internet : „LO”

29. Oct. 1969

Leonard Kleinrock from UCLA tries to log in a Stanford computer

UCLA

We typed the L… Do you see it?

Yes! We see the L

Stanford

We typed the O… Do you see it?

Yes! We see the O

We typed the G.

… and the system crashed …

* http://ftp.cs.ucla.edu/csd/first_words.html

The 70’s about Ethernet, TCP/IP and email…

1971	**Network Control Program (NCP)** Predecessor of TCP/IP
1972	**Email and Telnet**
1973	**Ethernet**
1974	**TCP/IP** Paper of Vint Cerf and Bob Kahn

80’s when TCP/IP went mainstream

1983	**NCP to TCP/IP**
	Domain Name Service (DNS)
1985	**NSFNet (TCP/IP)**
198x	**First Internet crashes caused by congestion**
1986	**Van Jacobson saves the Internet**
	congestion control

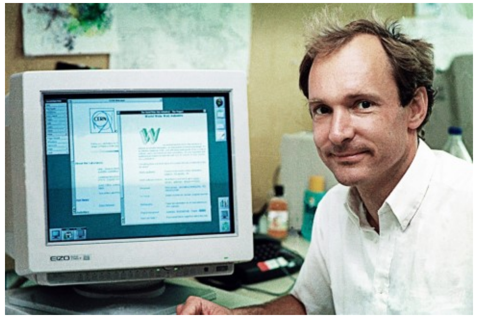
Van Jacobson

90's – the Internet going commercial…

1989 **ARPANET closed**

Birth of the WEB

Tim Berners Lee (CERN)



1993 **First search engine (Excite)**

1995 **NSFNet closed**

1998 **Google reinvents searching**

The new millenium bringing Web 2.0

1998	**IPv6 standardization**
2004	**Facebook goes online**
2006	**Google buys YouTube**
2007	**Netflix strats streaming videos**
2007	**First iPhone with mobile Internet access**

Fast Internet access everywhere, every device needs an Internet connection

2009 **Mining of the Bitcoin genesis block**

Fast mobile Internet access: 4G/LTE

IoT boom

Internet of Everything

2018 **Only 26% of Alexa Top 1000 sites reachable over IPv6**

Soon? **Encrypted transport protocols**

For example QUIC

Ultra-fast mobile access – 5G

To be continued…