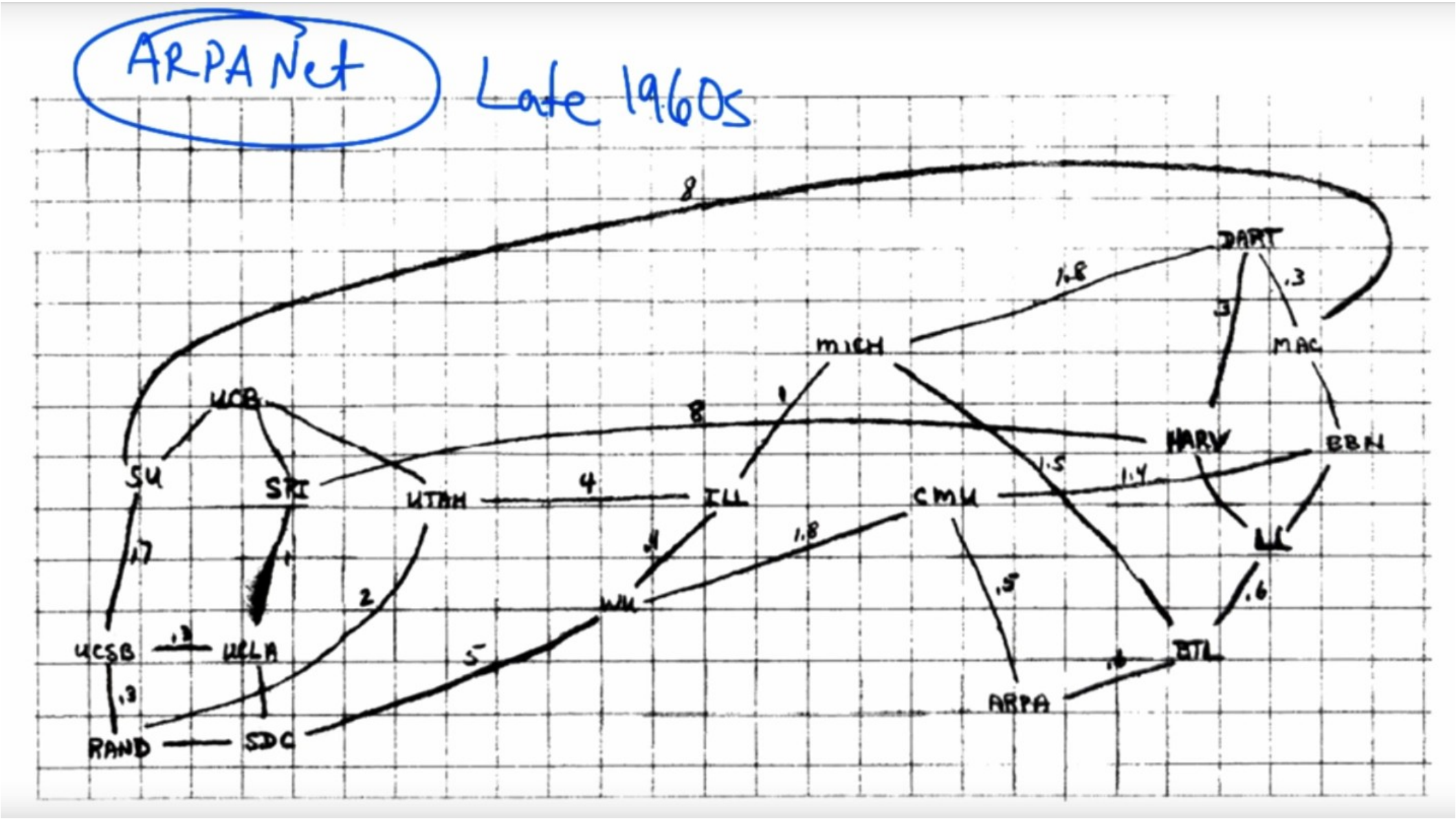
A Brief History of the Internet

- ARPANet (1966-1967) → UCLA, SRI, UCSB, Utah (1969)

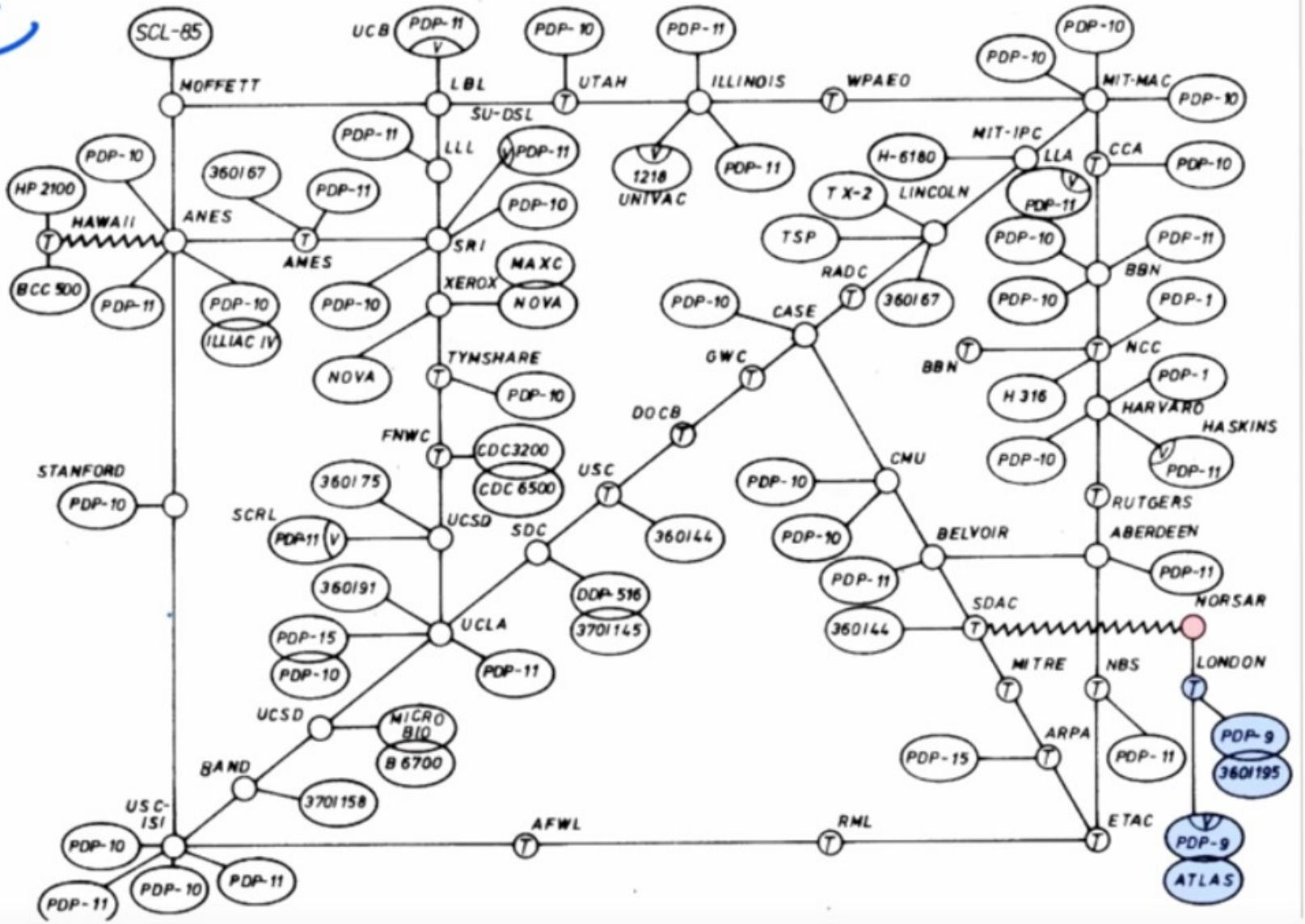
  Goal: Network academic computers

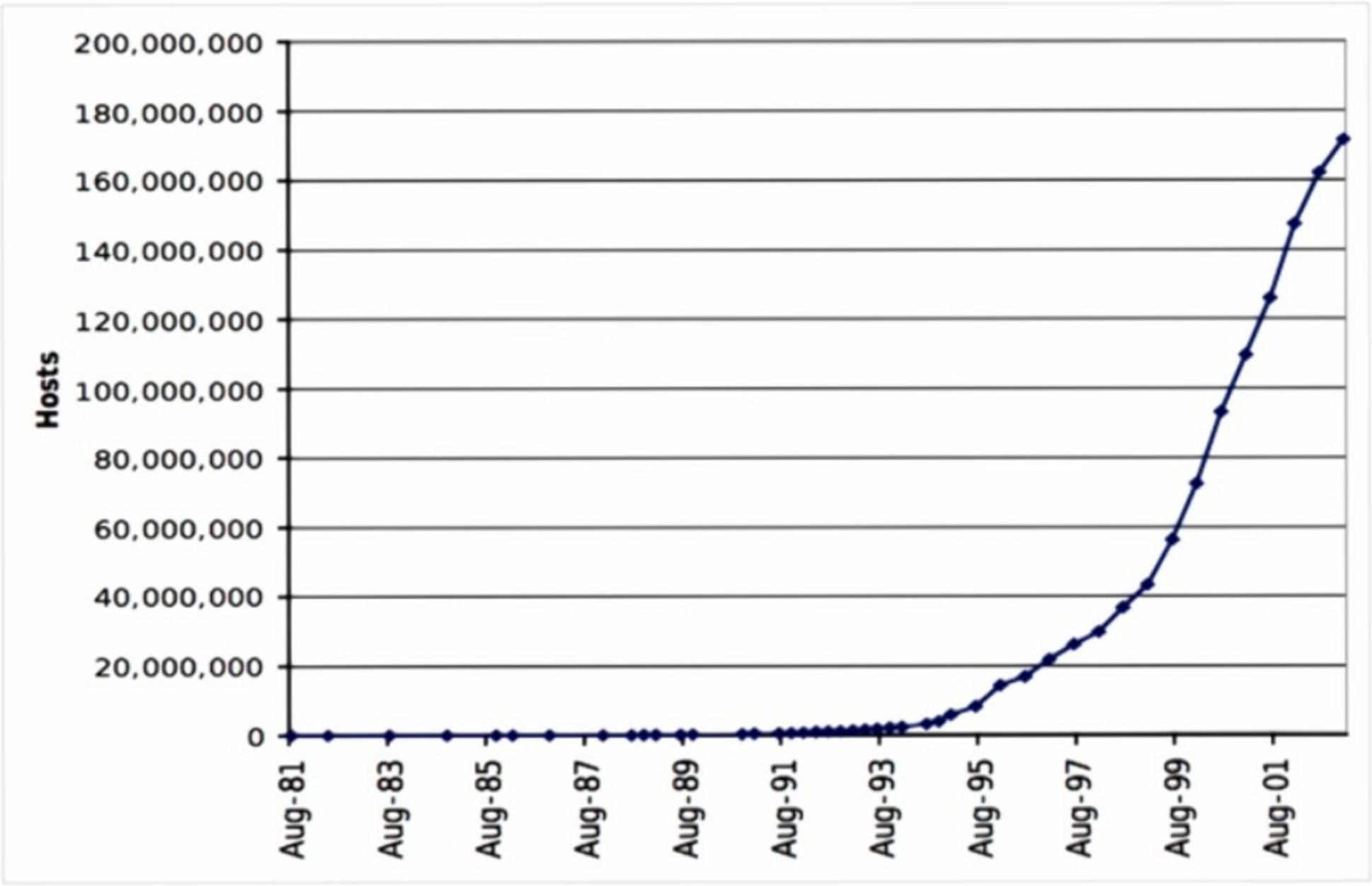
- NPL Net in UK around same time.

1971 → ~20 ARPANET Nodes

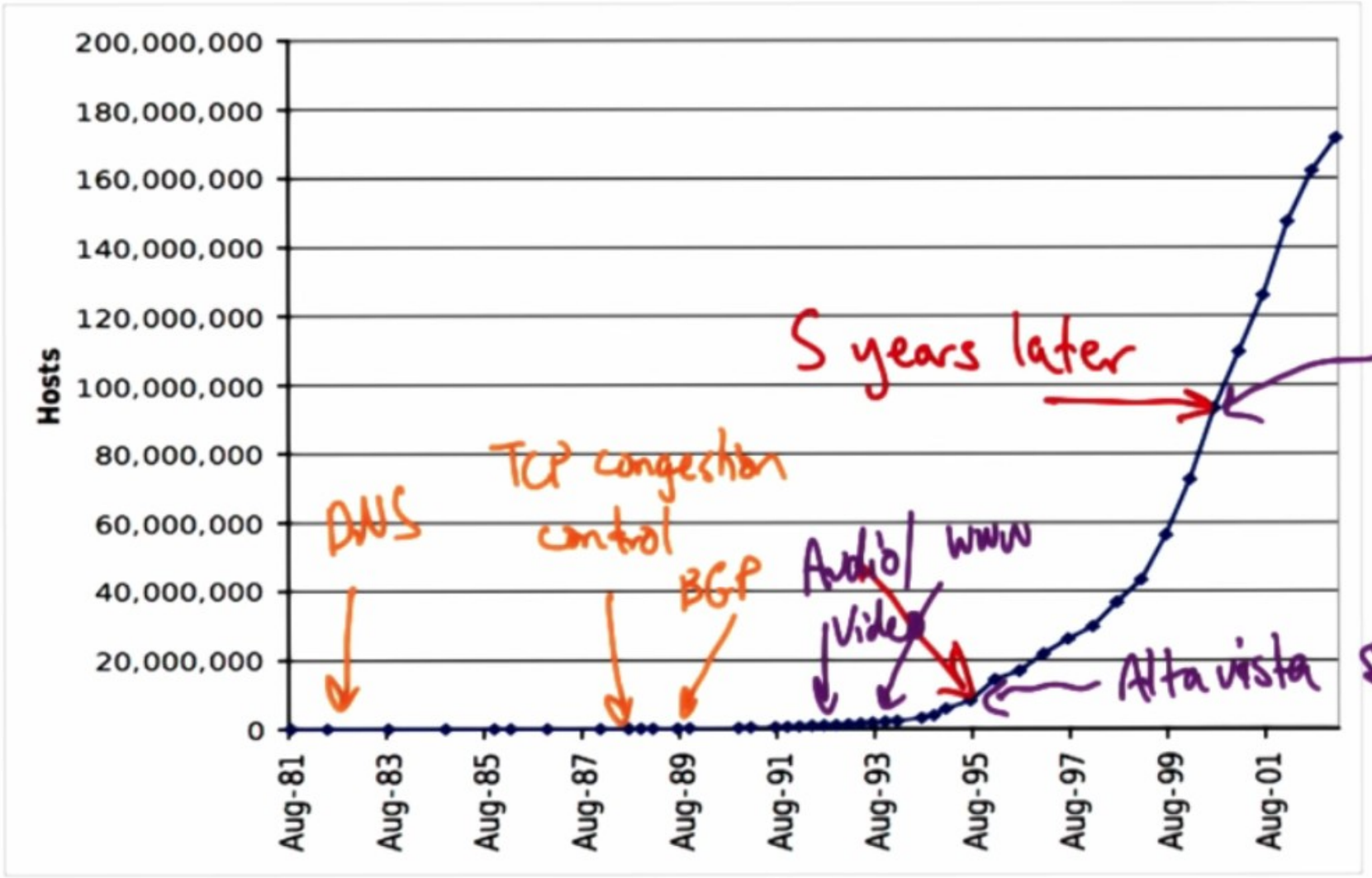

ARPANet
Late 1960s
8
DART
1.8
.3
3
MICH
MAC
UCB
8
1
HARV
BBN
SU
SRI
UTAH
4
ILL
CMU
1.5
1.4
LL
.7
1
.8
1.8
2
.5
.6
UCSB
.3
UCLA
5
.8
BTL
.3
ARPA
RAND
SDC

June 1974


SCL-85
MOFFETT
UCB
PDP-11
LBL
SU-DSL
PDP-10
UTAH
PDP-11
ILLINOIS
WPAFB
PDP-10
PDP-10
MIT-MAC
PDP-10
PDP-11
LLL
PDP-11
1218
UNIVAC
PDP-11
MIT-IPC
H-6180
LLA
CCA
PDP-10
PDP-10
360167
PDP-11
PDP-10
TX-2
LINCOLN
PDP-11
HP 2100
HAWAII
ANES
ANES
SRI
TSP
PDP-10
PDP-11
MAXC
XEROX
RADC
BBN
BCC 500
PDP-11
PDP-10
PDP-10
NOVA
360167
PDP-10
PDP-10
PDP-1
ILLIAC IV
CASE
NOVA
TYMSHARE
GWC
BBN
NCC
PDP-10
PDP-1
H 316
HARVARD
HASKINS
DOCB
FNWC
CDC3200
CDC 6500
CMU
PDP-10
PDP-11
STANFORD
PDP-10
360175
USC
360144
RUTGERS
SCRL
PDP-11
UCSD
SDC
PDP-10
BELVOIR
ABERDEEN
PDP-11
360191
DDP-516
3701145
SDAC
NORSAR
PDP-11
360144
UCLA
PDP-15
PDP-10
PDP-11
MITRE
NBS
LONDON
UCSD
MICRO 810
B 6700
PDP-9
3601195
ARPA
PDP-15
PDP-11
BAND
3701158
USC-ISI
AFWL
RML
ETAC
PDP-9
ATLAS
PDP-10
PDP-11
PDP-10
PDP-11



Source:Internet Software Consortium (http://www.isc.org/)



Source:Internet Software Consortium (http://www.isc.org/)

Problems & Growing Pains

IPv4

All require changes to basic infrastructure!

1. Running out of addresses → only 2^{32} addresses.
2. Congestion Control → insufficient dynamic range (wireless, high-speed intercontinental paths)
3. Routing → no security, easily misconfigured, poor convergence, non-determinism.
4. Security → lack of key management, secure software deployment
5. Denial of service → too easy to send traffic to destination

Internet Design Principles

→ Design Philosophy of the DARPA Internet Protocols,
DAVE CLARK 1988.

Conceptual Lessons → Principles designed for a certain type of network

Technical Lessons → ① Packet switching
② Fate sharing

Goal: "multiplexed utilization of existing interconnected networks"

sharing

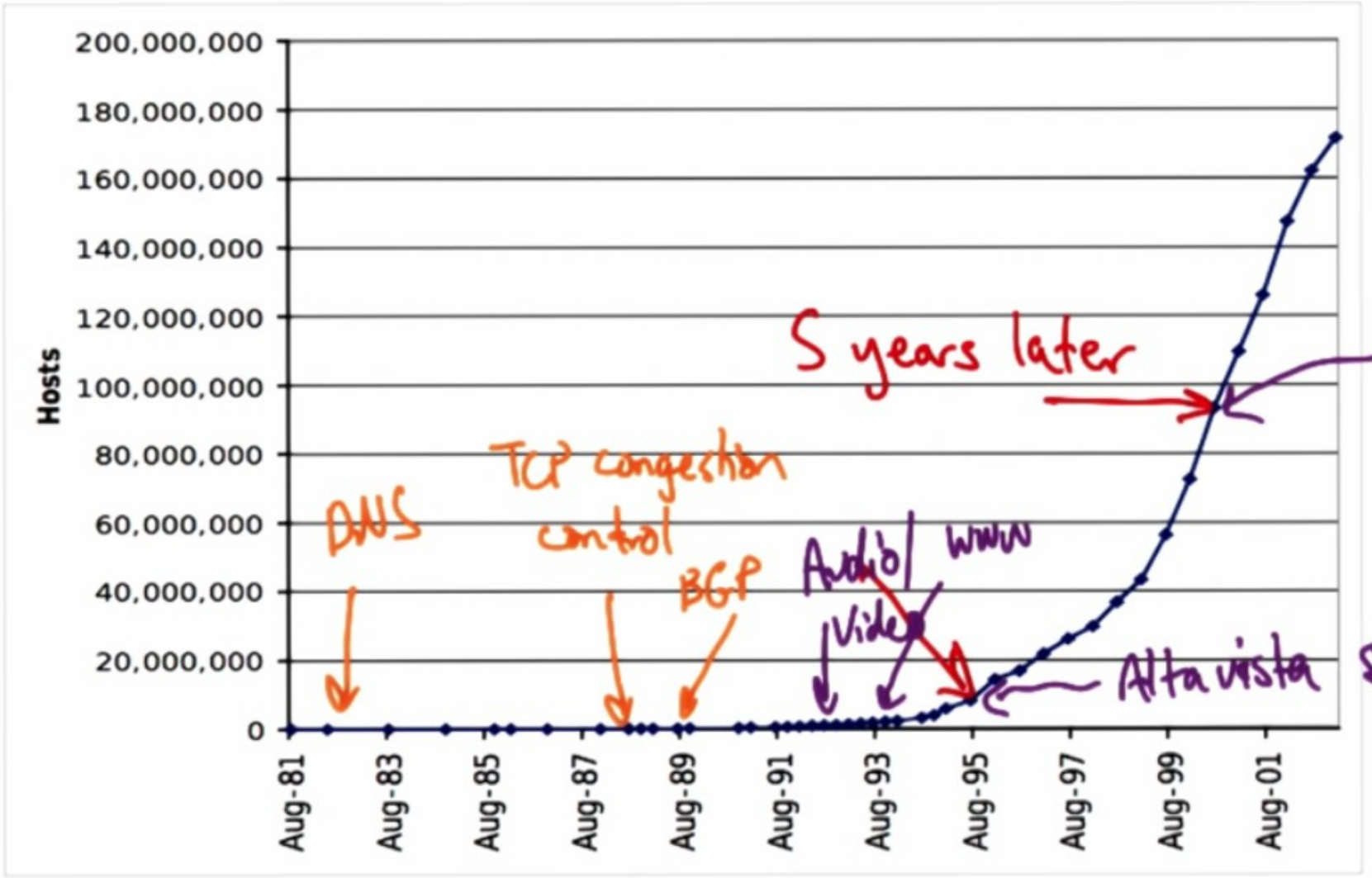
⇓

statistical multiplexing /
→ packet switching

interconnection

⇓

"narrow waist"



Source:Internet Software Consortium (http://www.isc.org/)

Packet Switching

- Information for forwarding traffic contained in destination address of packet.

Packet Switching

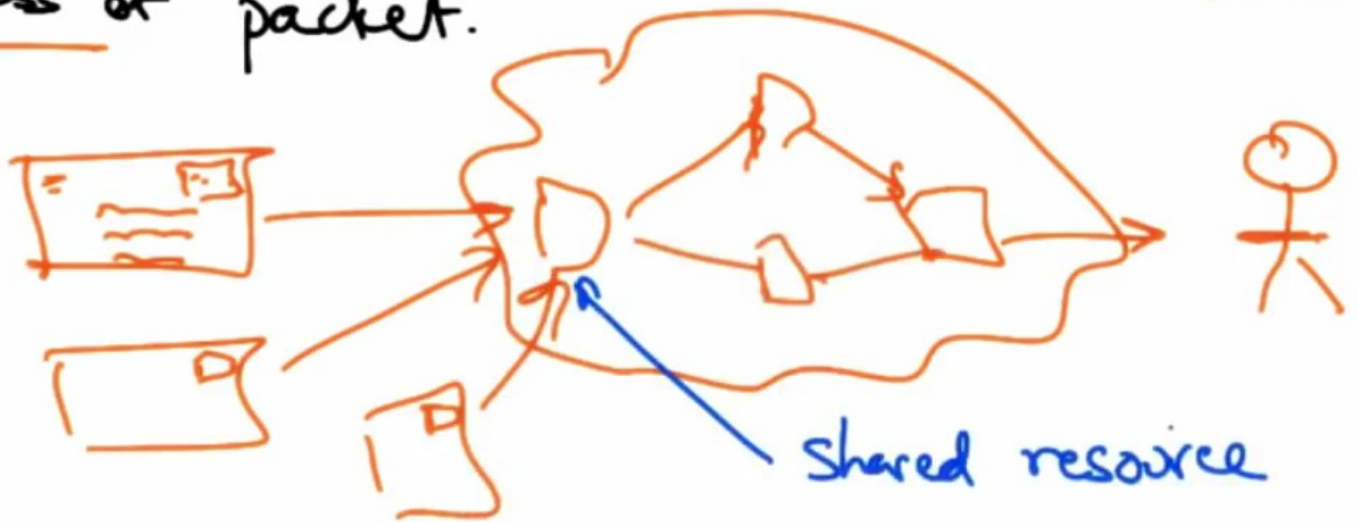
- No state in-network ahead of time
- "Best effort" service

- Information for forwarding traffic contained in destination address of packet.

Packet Switching

- No state in-network ahead of time
- "Best effort" service

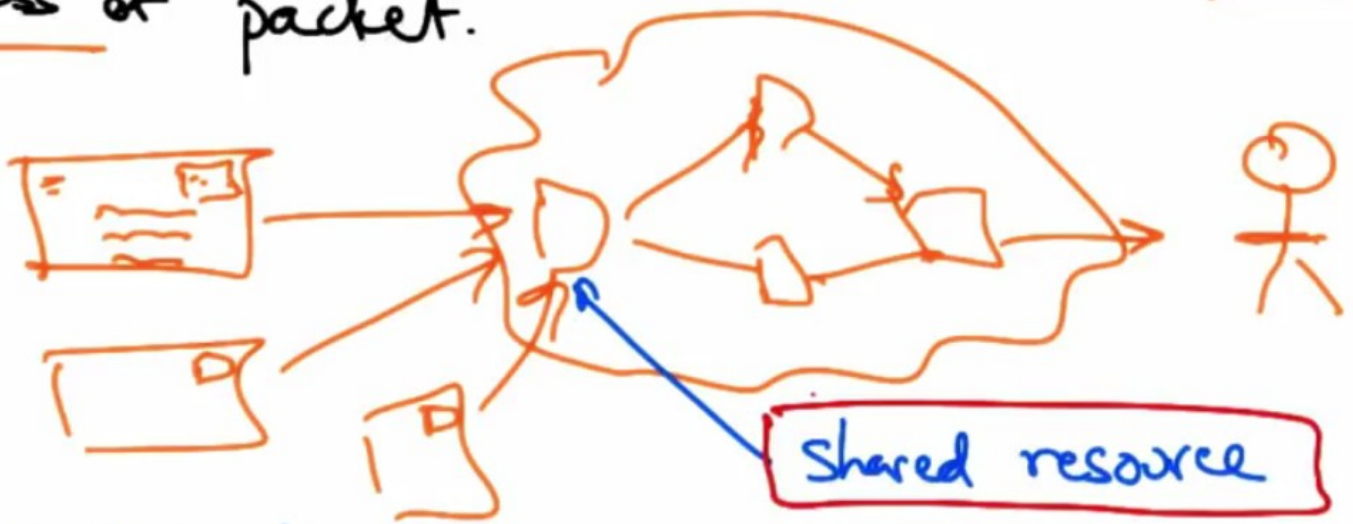
- Information for forwarding traffic contained in destination address of packet.



Packet Switching

- No state in-network ahead of time
- "Best effort" service

- Information for forwarding traffic contained in destination address of packet.



- In contrast to "circuit switching" → out-of-band signaling sets up dedicated path

Quiz: Packet Switching vs. Circuit Switching

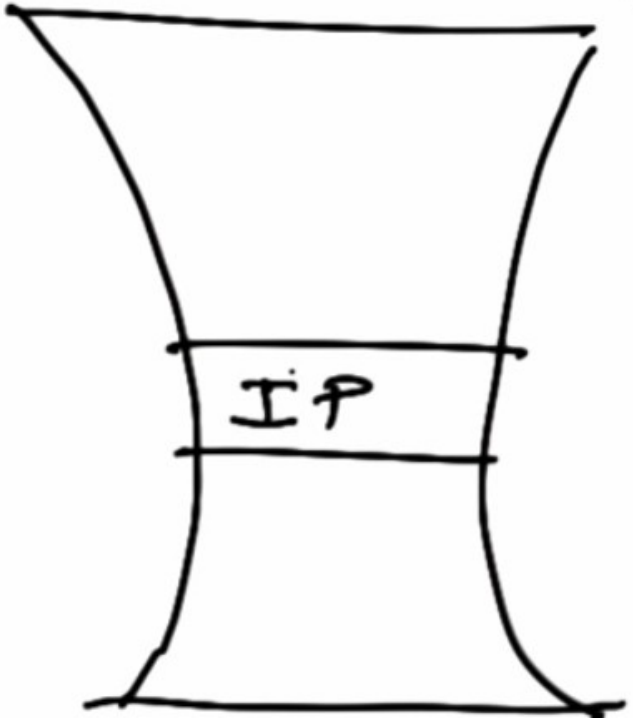
PS	CS	
☐	☐	Variable Delay
☐	☐	"Busy signals"
☐	☐	Sharing of network resources
☐	☐	Dedicated resources between sender & receiver

Quiz: Packet Switching vs. Circuit Switching

PS	CS	
☑	☐	Variable Delay
☐	☑	"Busy signals"
☑	☐	Sharing of network resources
☐	☑	Dedicated resources between sender & receiver

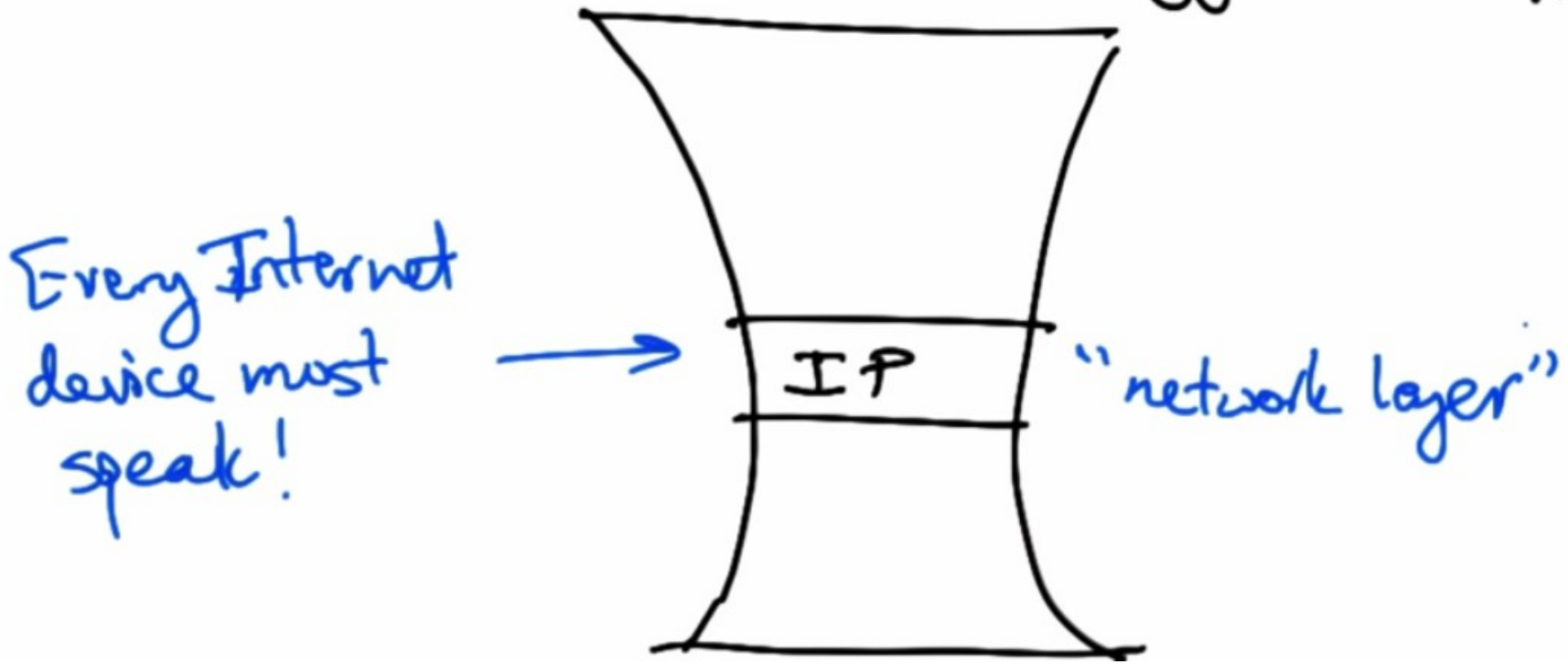
Interconnection: Narrow Waist

Goal: Interconnect many existing networks.
Hide underlying technology from applications



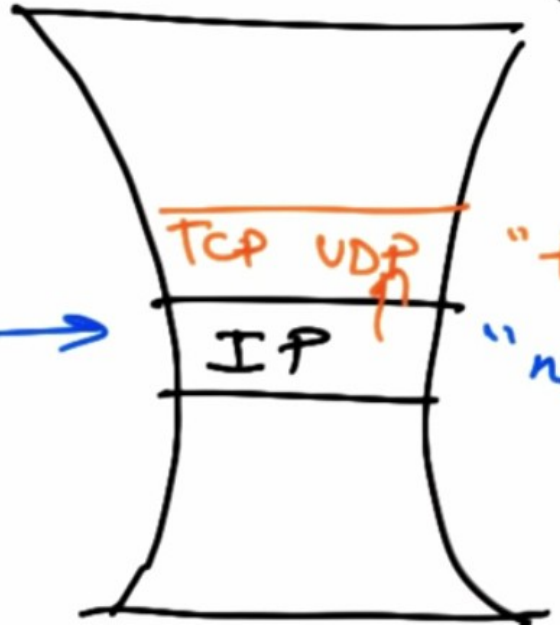
Interconnection: Narrow Waist

Goal: Interconnect many existing networks.
Hide underlying technology from applications



Interconnection: Narrow Waist

Goal: Interconnect many existing networks.
Hide underlying technology from applications



Every Internet device must speak!

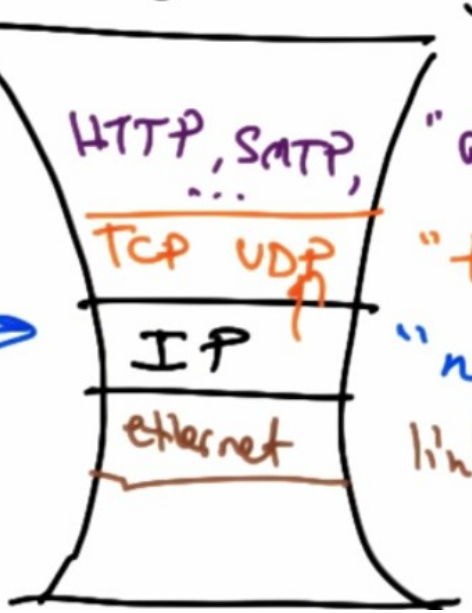
"transport layer"

"network layer"

} e2e connectivity

Interconnection: Narrow Waist

Goal: Interconnect many existing networks.
Hide underlying technology from applications



Every Internet device must speak!

"application layer" } reliable transport, congestion control

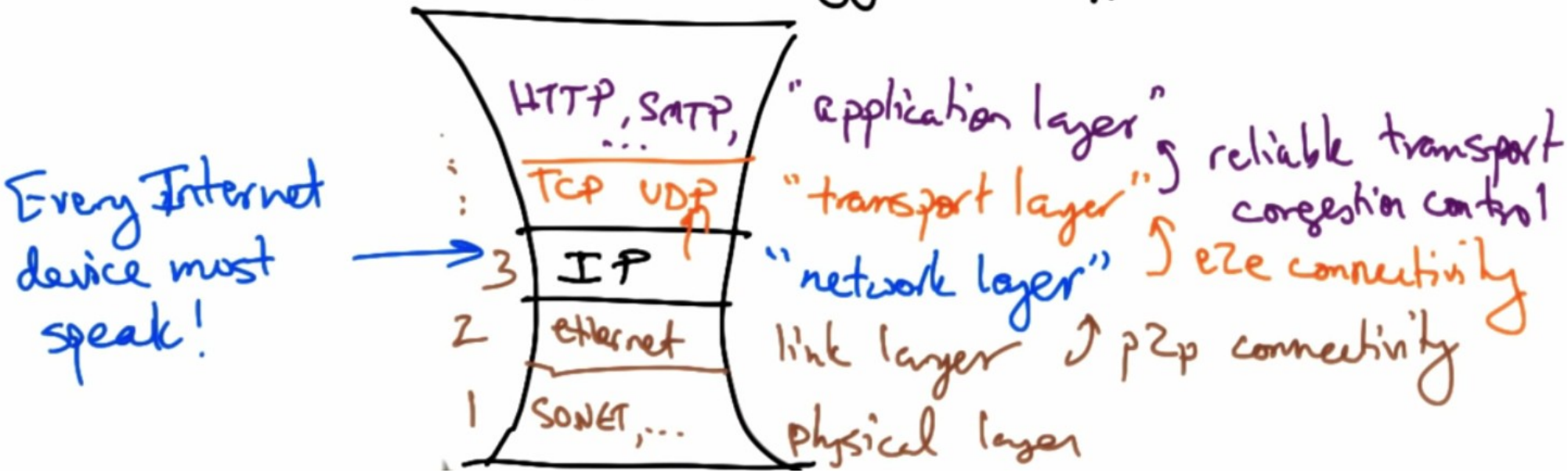
"transport layer" } e2e connectivity

"network layer"

link layer } p2p connectivity

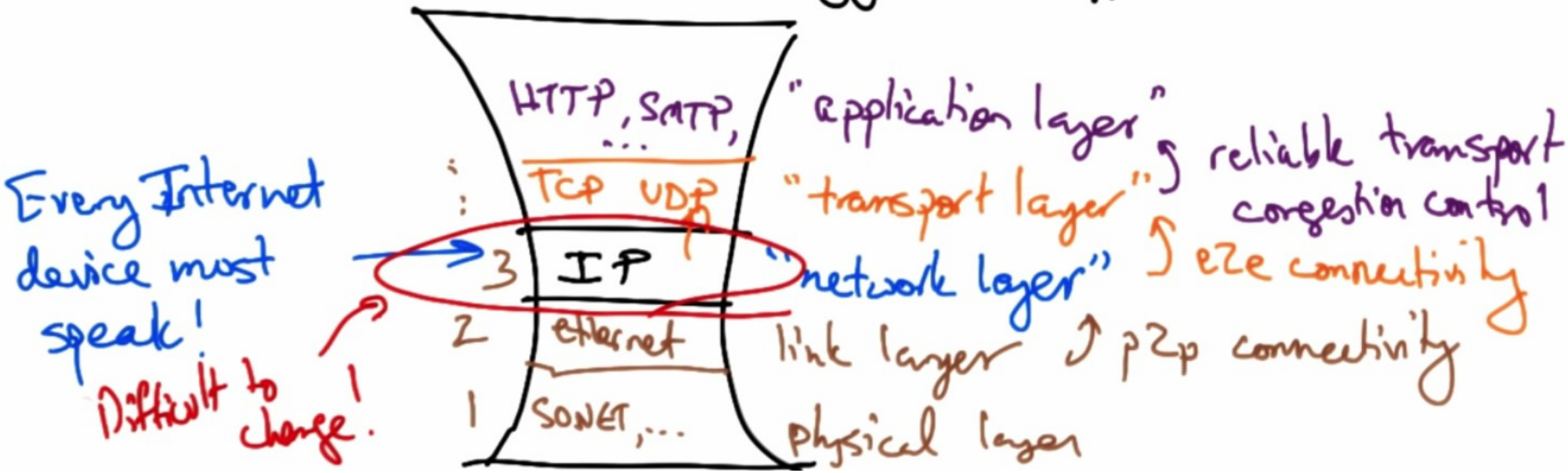
Interconnection: Narrow Waist

Goal: Interconnect many existing networks.
Hide underlying technology from applications



Interconnection: Narrow Waist

Goal: Interconnect many existing networks.
Hide underlying technology from applications



Goals

- ✓ Interconnection
- ✓ Sharing

Goals

- ✓ Interconnection
- ✓ Sharing

- Survivability: network works even if some devices fail.

Goals

- ✓ Interconnection
- ✓ Sharing

- Survivability: network works even if some devices fail.
  - → replication
  - → fate sharing

Goals

- ✓ Interconnection
- ✓ Sharing

- Survivability: network works even if some devices fail.
  - → replication
  - → fate sharing (???)
    - ↗ easier to withstand complex failure
    - ↘ easier engineering

Goals

- ✓ Interconnection
- ✓ Sharing

- Survivability: network works even if some devices fail.
  - → replication
  - → fate sharing (???)
    - ↗ easier to withstand complex failure
    - ↘ easier engineering
- Heterogeneity: TCP/IP

Goals

- ✓ Interconnection
- ✓ Sharing

- Survivability: network works even if some devices fail.
  - → replication
  - → fate sharing (???)
    - ↗ easier to withstand complex failure
    - ↘ easier engineering
- Heterogeneity: ~~TCP/IP~~
  - streaming, DNS

Goals

- ✓ Interconnection
- ✓ Sharing

- Survivability: network works even if some devices fail.
  - → replication
  - → fate sharing (???)
    - ↗ easier to withstand complex failure
    - ↘ easier engineering
- Heterogeneity
  - ① ~~TCP~~ / IP
  - ② "Best effort"
    - streaming, DNS

Goals

- ✓ Interconnection
- ✓ Sharing

- Survivability: network works even if some devices fail.
  - → replication
  - → fate sharing (???)
    - ↗ easier to withstand complex failure
    - ↘ easier engineering
- Heterogeneity
  - ① ~~TCP~~ / IP
    - streaming, DNS
  - ② "Best effort"
    - + simple
    - × debugging is hard

Goals

- ✓ Interconnection
- ✓ Sharing

- **Survivability**: network works even if some devices fail.
  - → replication
  - → fate sharing (???)
    - ↗ easier to withstand complex failure
    - ↘ easier engineering
- **Heterogeneity**: ① ~~TCP~~ IP ② "Best effort"
  - ~~TCP~~: streaming, DNS
  - + simple
  - × debugging is hard
- **Distributed Management**: Addressing (ARIN, RIPE, etc.)

Goals

- ✓ Interconnection
- ✓ Sharing

- **Survivability**: network works even if some devices fail.
  - → replication
  - → fate sharing (???)
    - ↗ easier to withstand complex failure
    - ↘ easier engineering
- **Heterogeneity**: ① ~~TCP~~ IP ② "Best effort"
  - ~~TCP~~: streaming, DNS
  - + simple
  - × debugging is hard
- **Distributed Managent**:
  - Addressing (ARIN, RIPE, etc.)
  - Naming (DNS)
  - Routing (BGP)

Goals

✓ Interconnection

✓ Sharing

✓ Survivability: network works even if some devices fail.

→ replication

→ fate sharing (???)
- → easier to withstand complex failure
- → easier engineering

✓ Heterogeneity: ① ~~TCP~~ IP ② "Best effort"
+ simple
× debugging is hard

~~TCP~~: streaming, DNS

✓ Distributed Management:
+ organic growth
+ stable management
× no "owner"

Addressing
Naming (DNS)
Routing (BGP)

✓ Cost
- Ease of attachment
- Accountability

What's Missing?

- × Security
- × Availability
- × Mobility
- × Scaling
- ⋮

Quiz

- [ ] Security
- [ ] Heterogeneity
- [ ] Interconnection
- [ ] Sharing
- [ ] Mobility

Quiz

- [ ] Security
- [x] Heterogeneity
- [x] Interconnection
- [x] Sharing
- [ ] Mobility

End-to-End Argument

Saltzer, Reed, Clark (1981)

"The function in question can completely and correctly be implemented only with the knowledge and help of the application standing at the end points of the communication system. Therefore, providing that questioned function as a feature of the communication system itself is not possible. (Sometimes an incomplete version of the function provided by the communication system may be useful as a performance enhancement.)"

End-to-End Argument

Saltzer, Reed, Clark (1981)

"The function in question can completely and correctly be implemented only with the knowledge and help of the application standing at the end points of the communication system. Therefore, providing that questioned function as a feature of the communication system itself is not possible. (Sometimes an incomplete version of the function provided by the communication system may be useful as a performance enhancement.)"

1. Error handling in file transfer
2. End-to-end encryption
3. TCP/IP split in error handling

→ "Dumb network, intelligent endpoints"

File Transfer

(1) Read file from disk

(2) Communication system sends file

(3) Transmits packets

(4) Give file to file transfer program

(5) Write to disk

File Transfer

What can go wrong?

1. Read/write
2. Breaking up/assembling
3. Communication system

× ① Read file from disk

× ② Communication system sends file

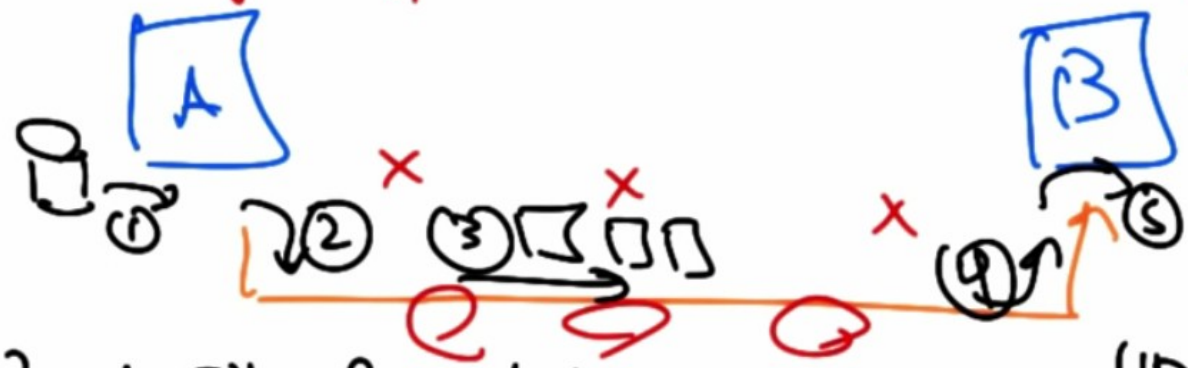
× ③ Transmits packets

× ④ Give file to file transfer program

× ⑤ Write to disk

File Transfer

All solutions require app level checks!



What can go wrong?

1. Read / write
2. Breaking up / assembling
3. Communication system

- ✗ (1) Read file from disk
- ✗ (2) Communication system sends file
- ✗ (3) Transmits packets
- ✗ (4) Give file to file transfer program
- ✗ (5) Write to disk

End-to-End Argument

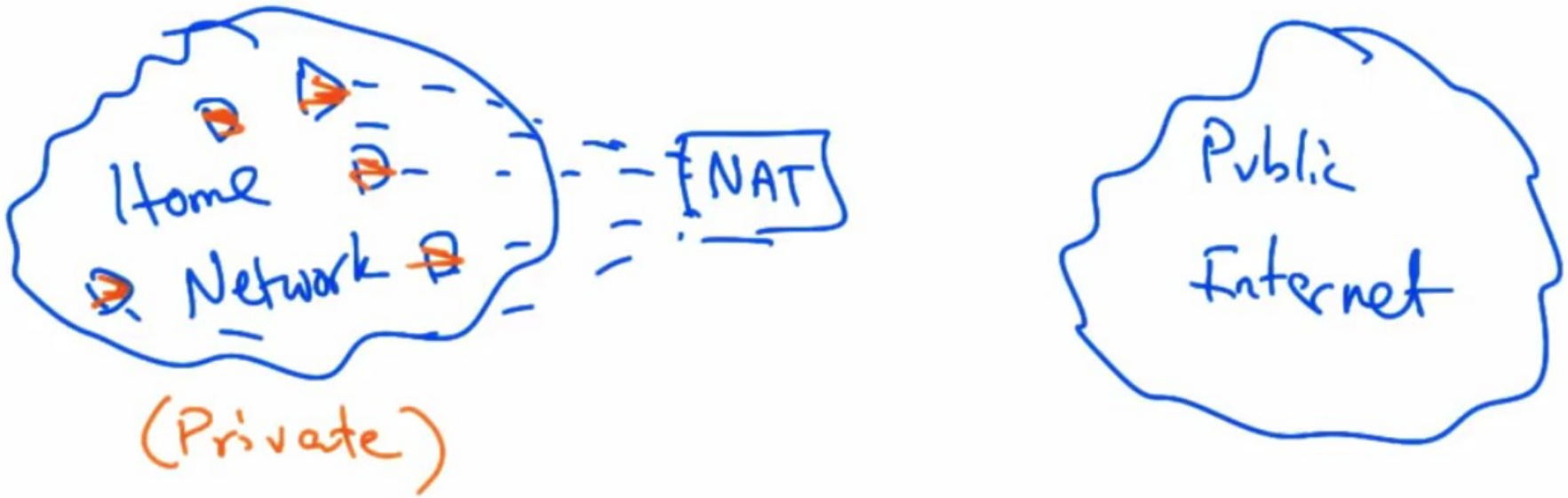
Violations:

- NAT
- VPN tunnels
- TCP splitting
- Spam → What about spam filters?
- P2P systems
- Caches

Questions:

- What's in/out?
  - Routing
  - Multicast
  - QoS
  - NAT
  - :

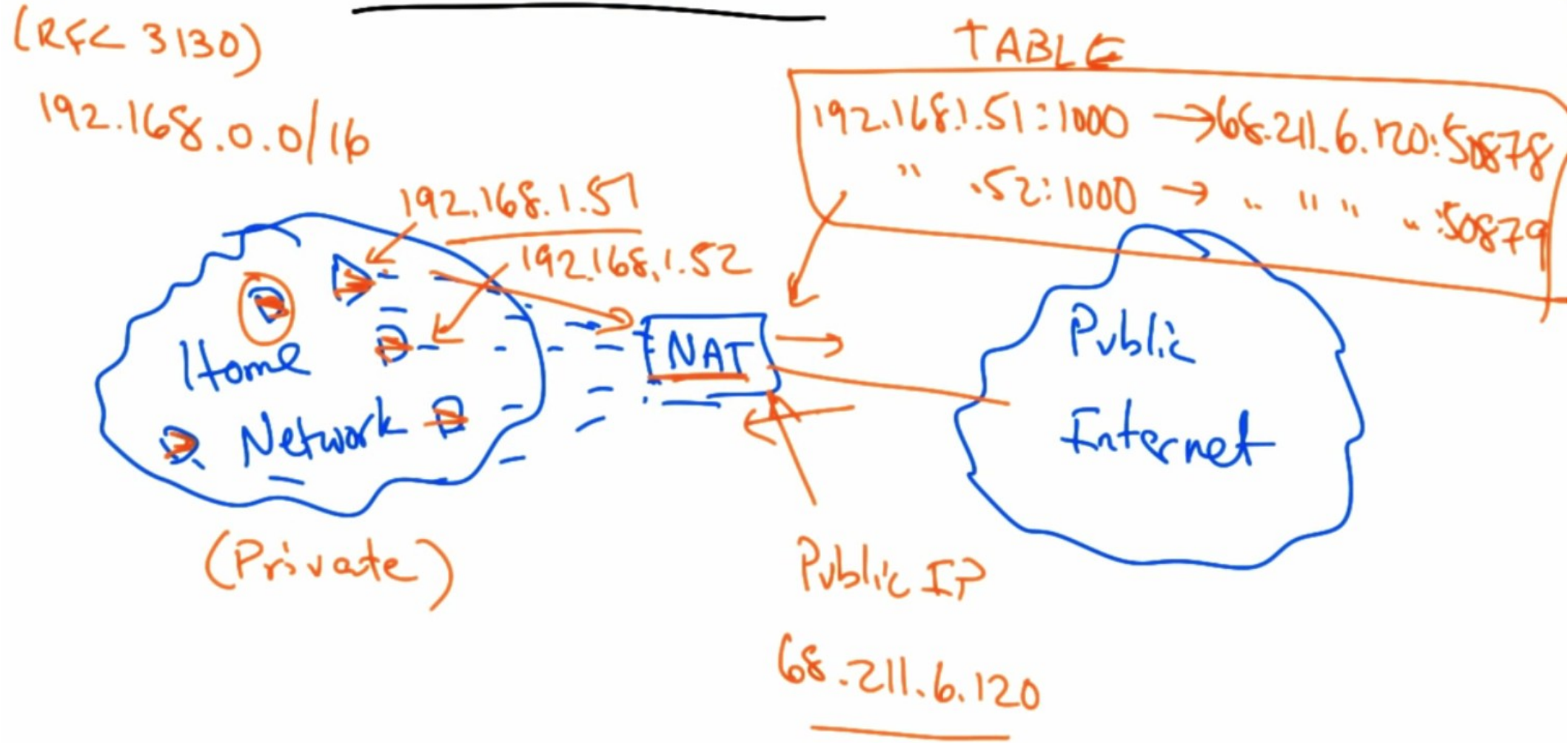
Violation: NAT



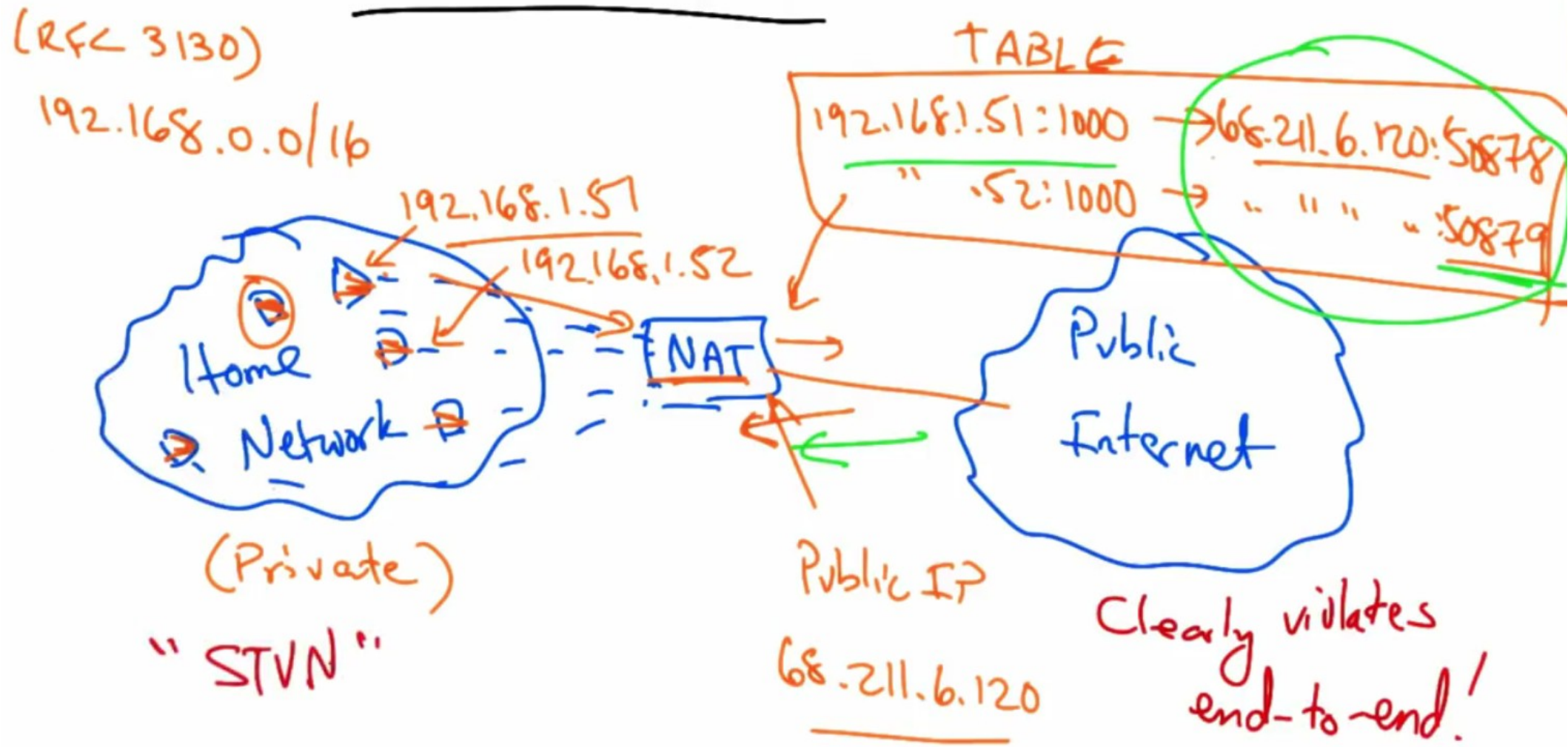
Violation: NAT

(RFC 3130)

192.168.0.0/16



Violation: NAT



Switching and Bridging

Problem: How hosts find each other on a subnet.

How subnets are interconnected

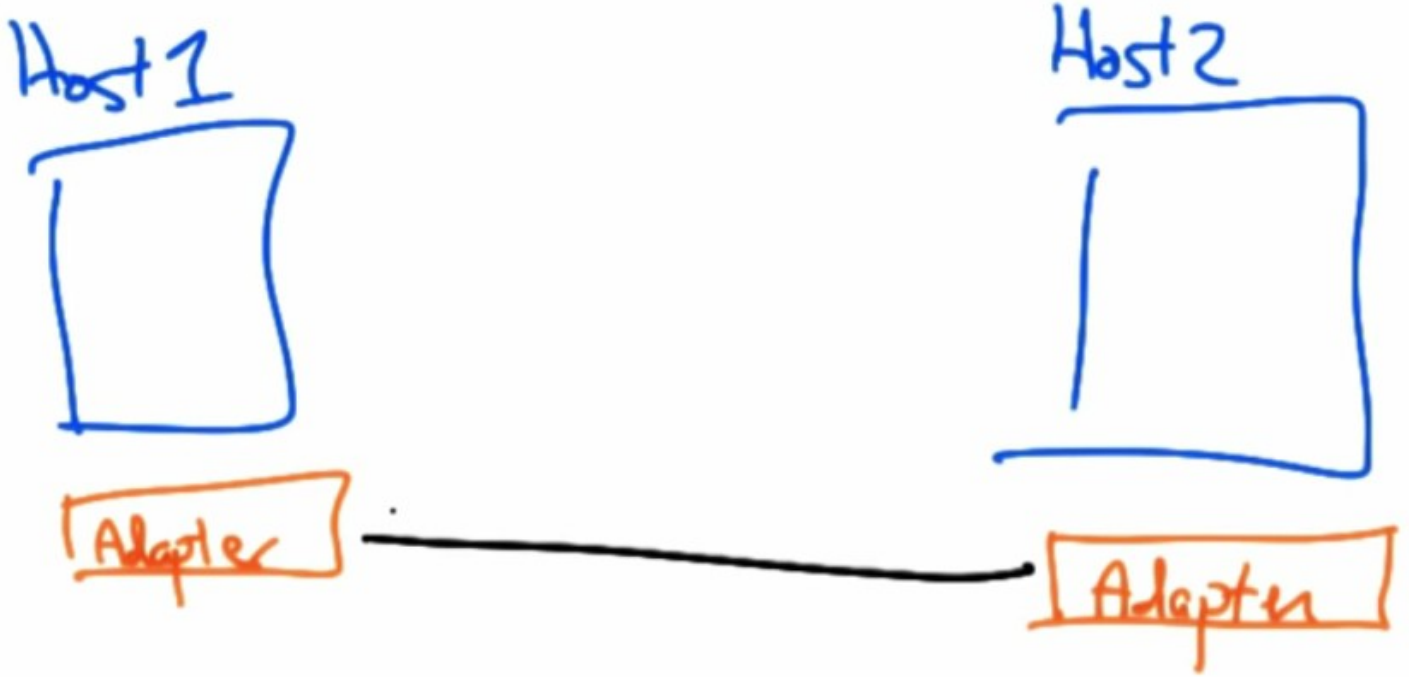
Also:

Switches vs. hubs

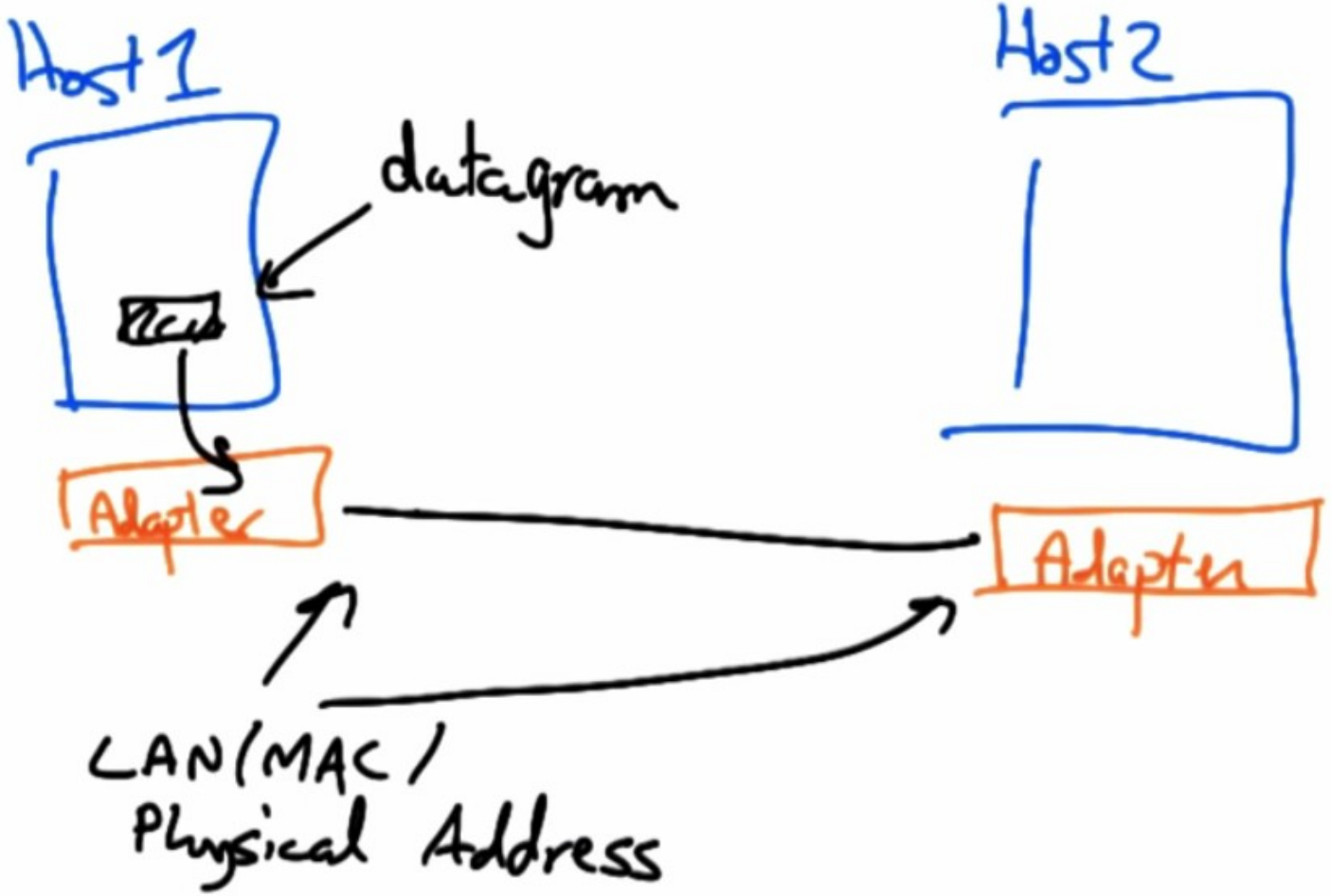
Switches vs. routers

How to scale Ethernet

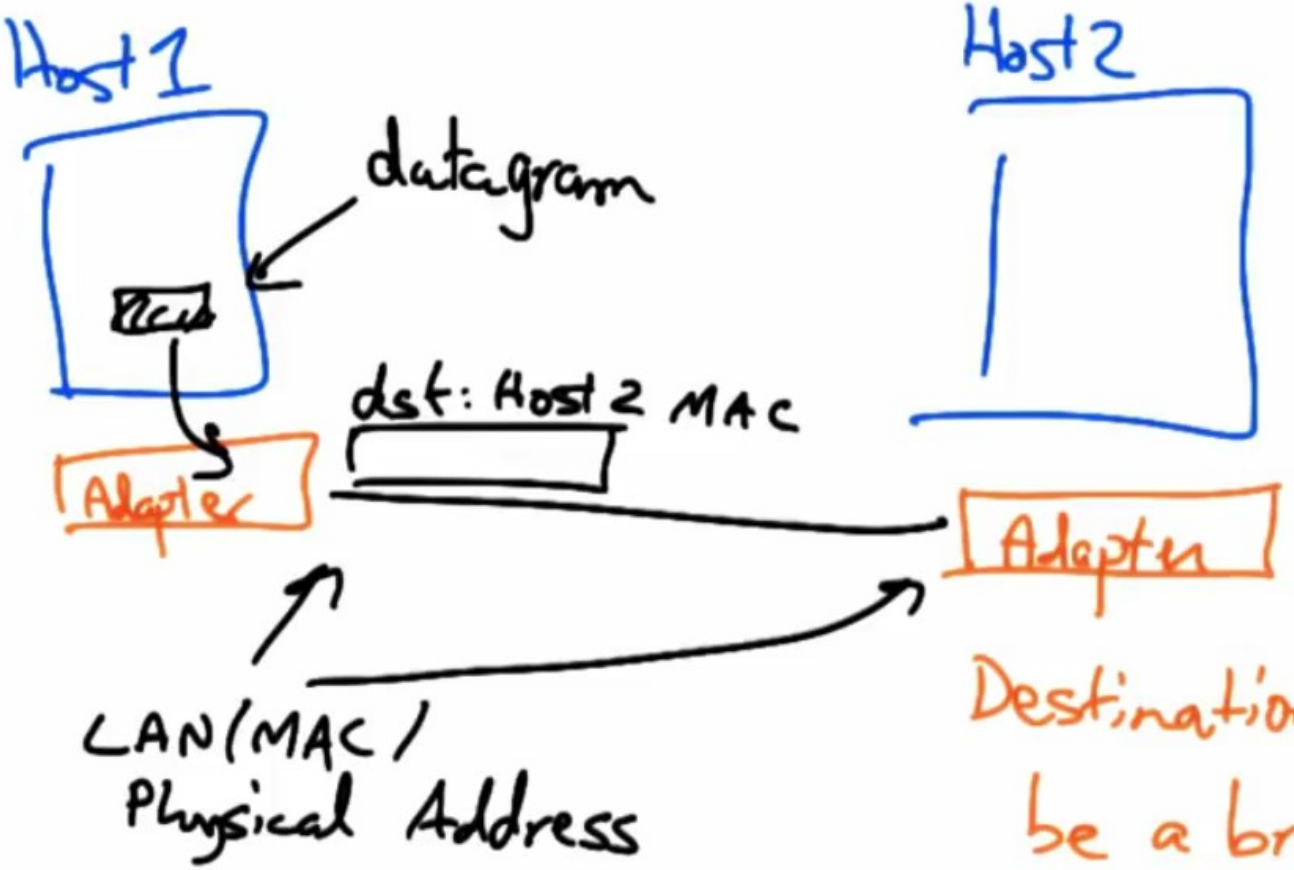
Bootstrapping: Networking Two Hosts



Bootstrapping: Networking Two Hosts



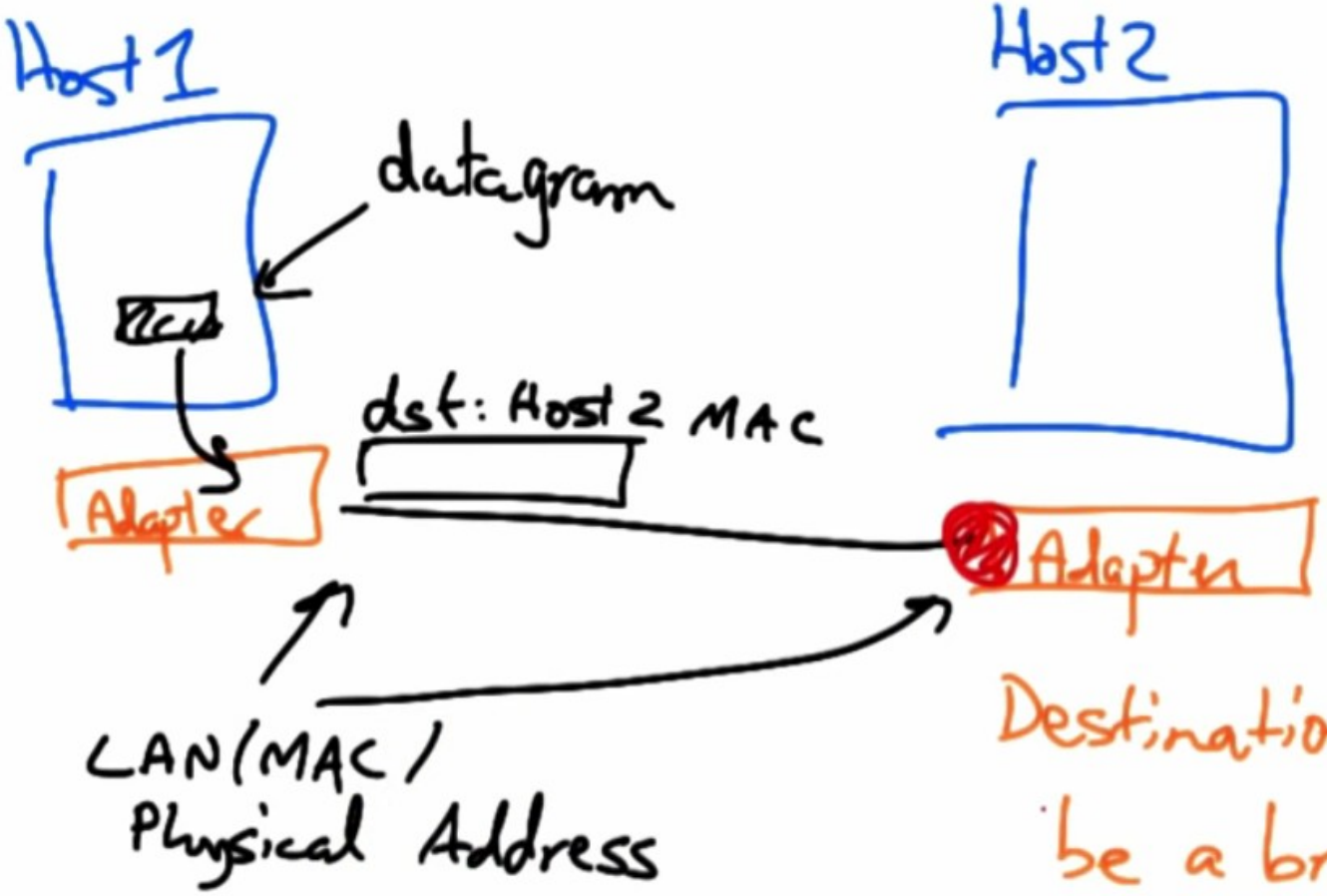
Bootstrapping: Networking Two Hosts



Destination can also be a broadcast MAC address.

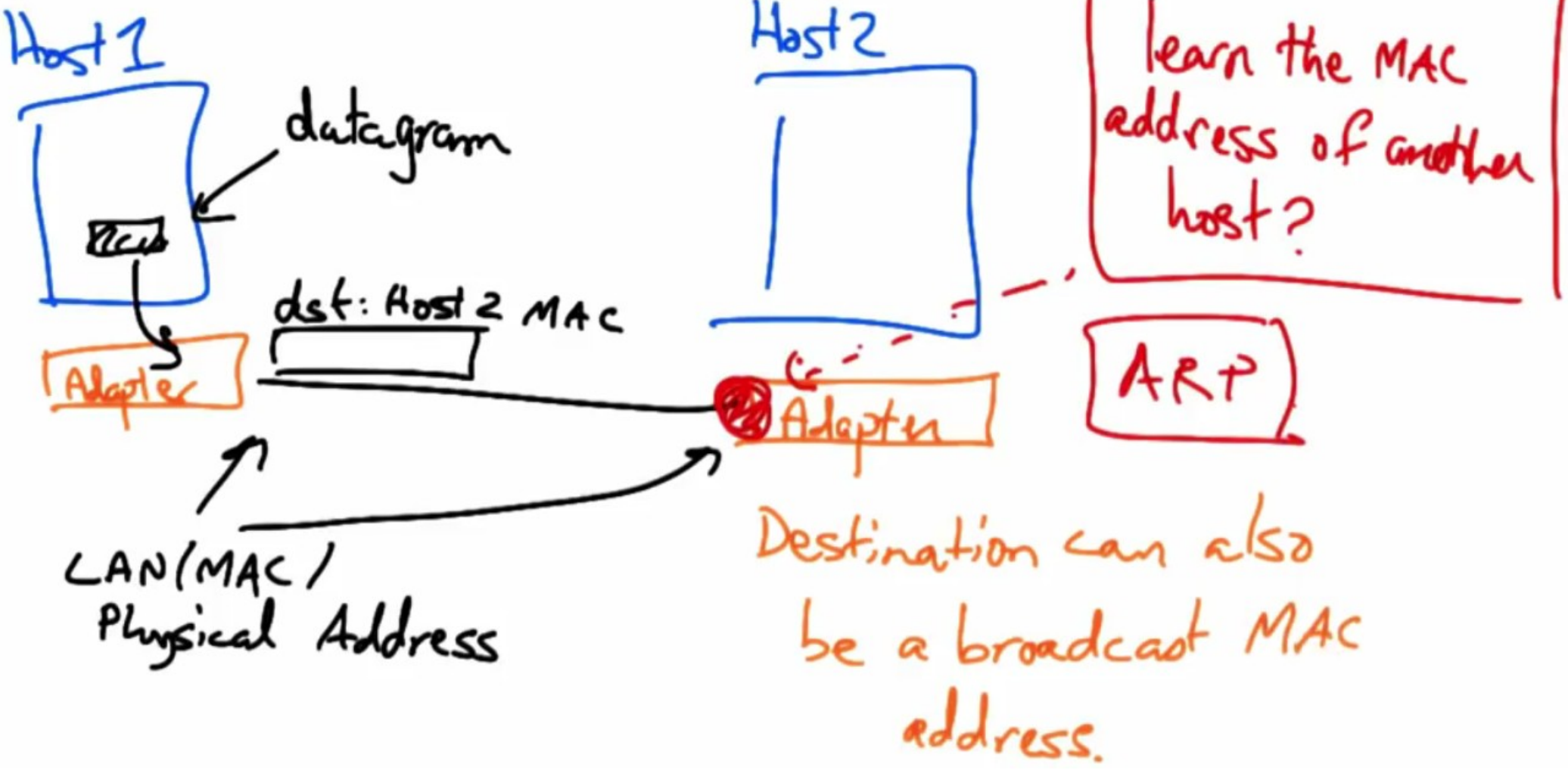
Bootstrapping: Networking Two Hosts

How does a host learn the MAC address of another host?

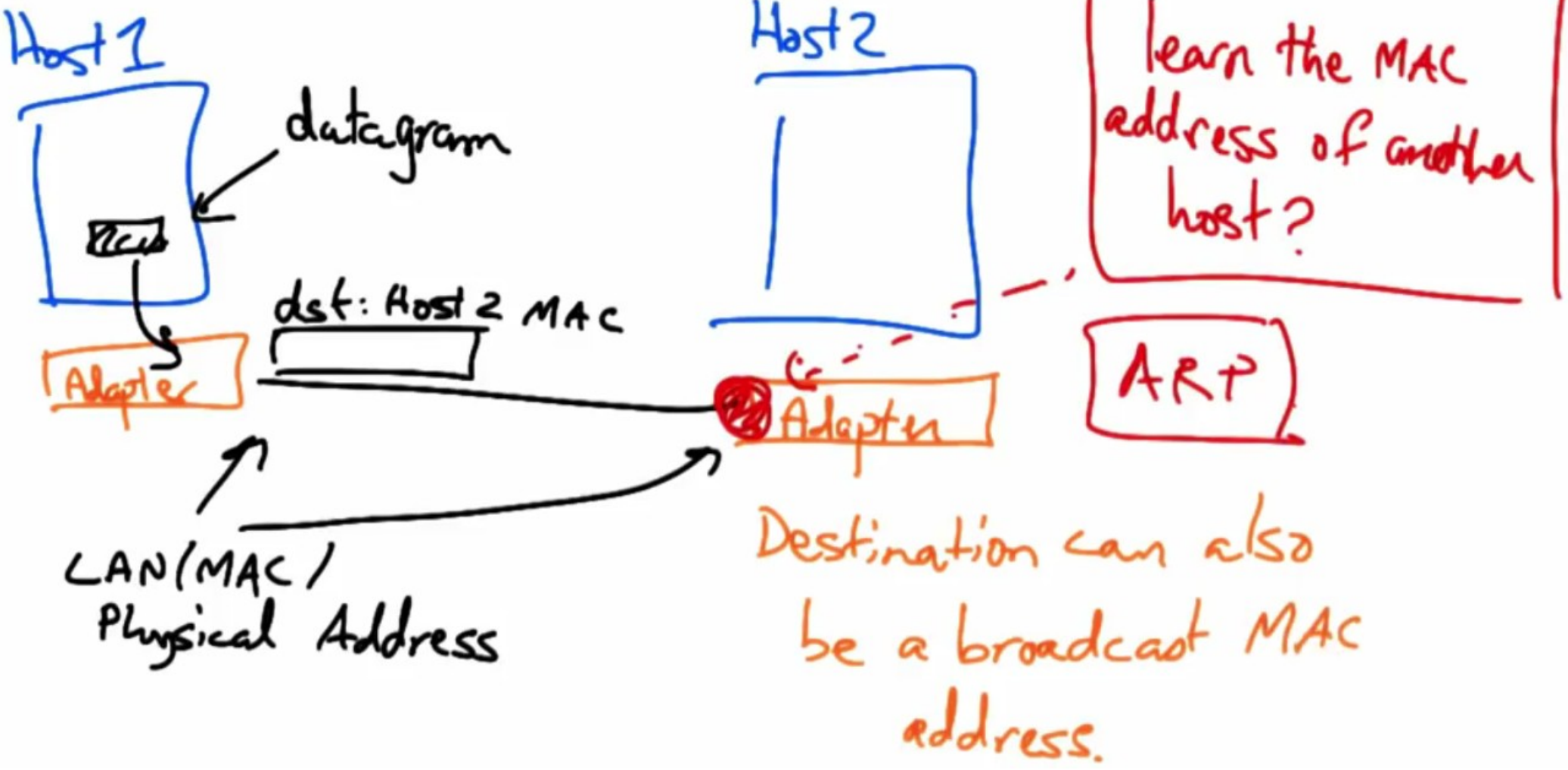


Destination can also be a broadcast MAC address.

Bootstrapping: Networking Two Hosts



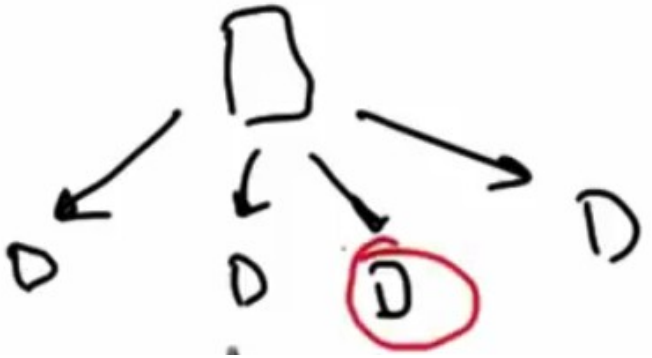
Bootstrapping: Networking Two Hosts



ARP: Address Resolution Protocol

Host queries with an IP address:

"Who has IP address 130.207.160.47?"

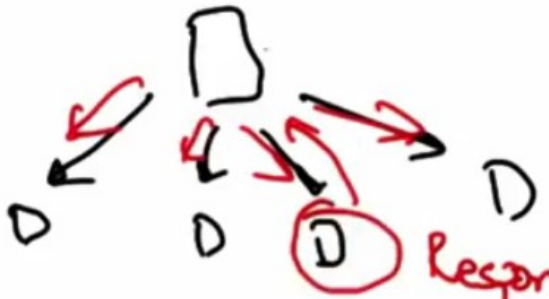


Broadcast

ARP: Address Resolution Protocol

Host queries with an IP address:

"Who has IP address 130.207.160.47?"



Broadcast

Response is the MAC address corresponding to 130.207.160.47

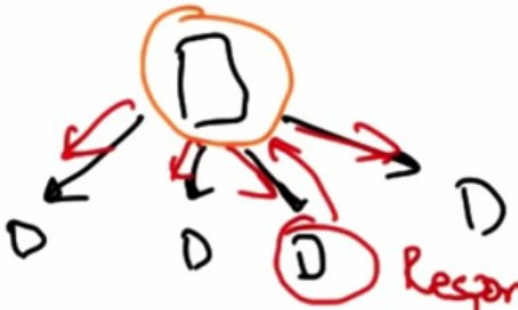
ARP: Address Resolution Protocol

Host queries with an IP address:

"Who has IP address 130.207.160.47?"

ARP Table

IP	MAC
130.207.160.47	ab:cd:ef:....



Broadcast

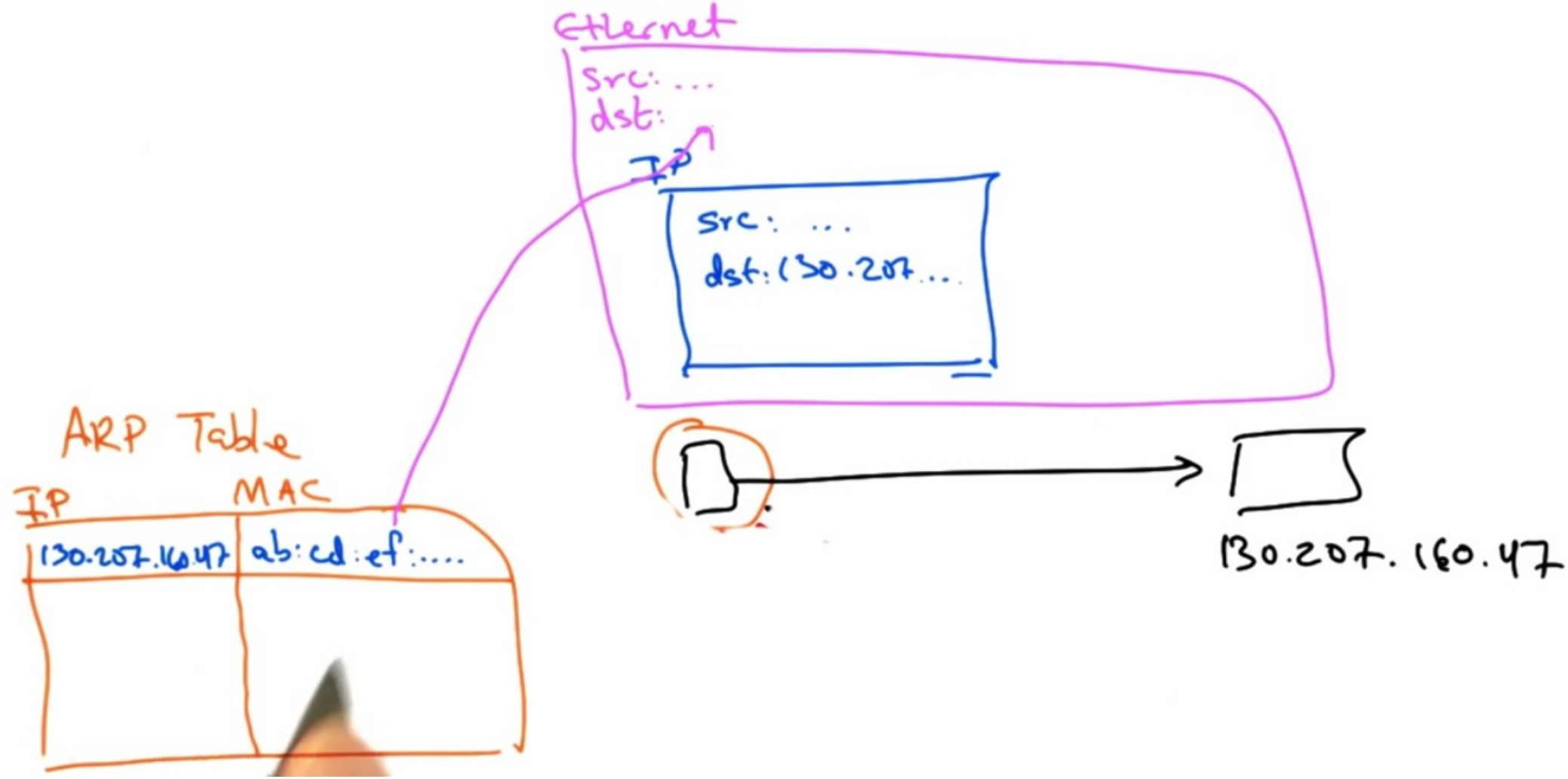
Response is the MAC address corresponding to 130.207.160.47

Encapsulation

ARP Table

IP	MAC
130.207.160.47	ab:cd:ef:....

Encapsulation



Quiz: ARP

What are the queries and responses in ARP?

- ☐ Query: Broadcast asking about IP
  Response: Unicast with MAC address
- ☐ Query: Unicast asking about IP
  Response: Broadcast with MAC address
- ☑ Query: Broadcast asking about MAC address
  Response: Unicast with IP address

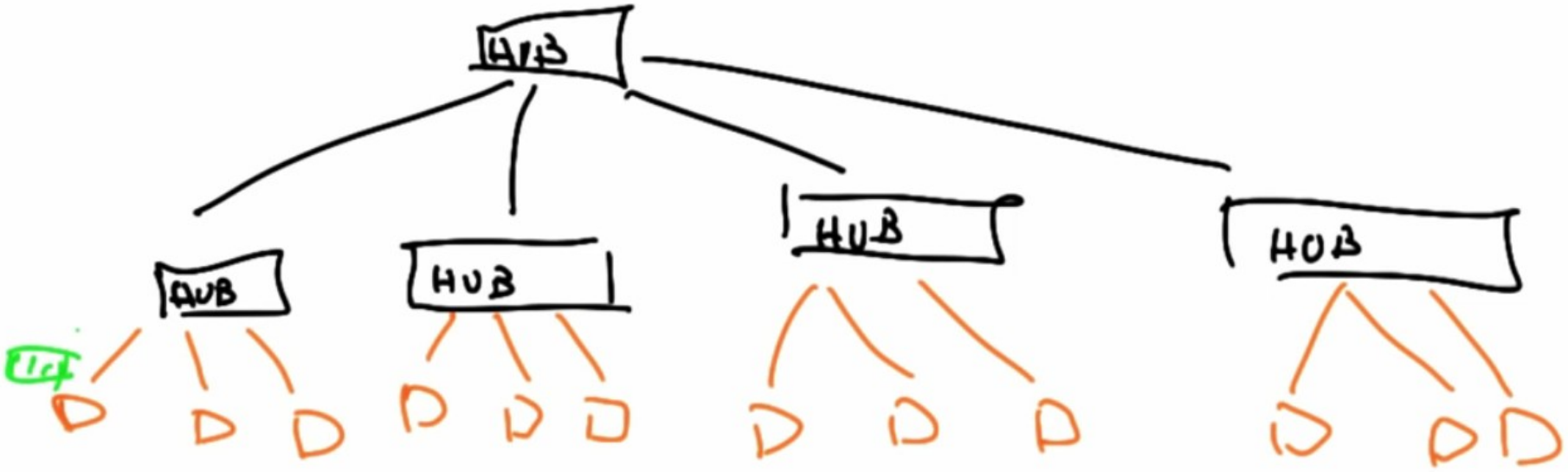
Quiz: ARP

What are the queries and responses in ARP?

- [x] Query: Broadcast asking about IP
  Response: Unicast with MAC address
- [ ] Query: Unicast asking about IP
  Response: Broadcast with MAC address
- [ ] Query: Broadcast asking about MAC address
  Response: Unicast with IP address

Interconnecting LANs with Hubs

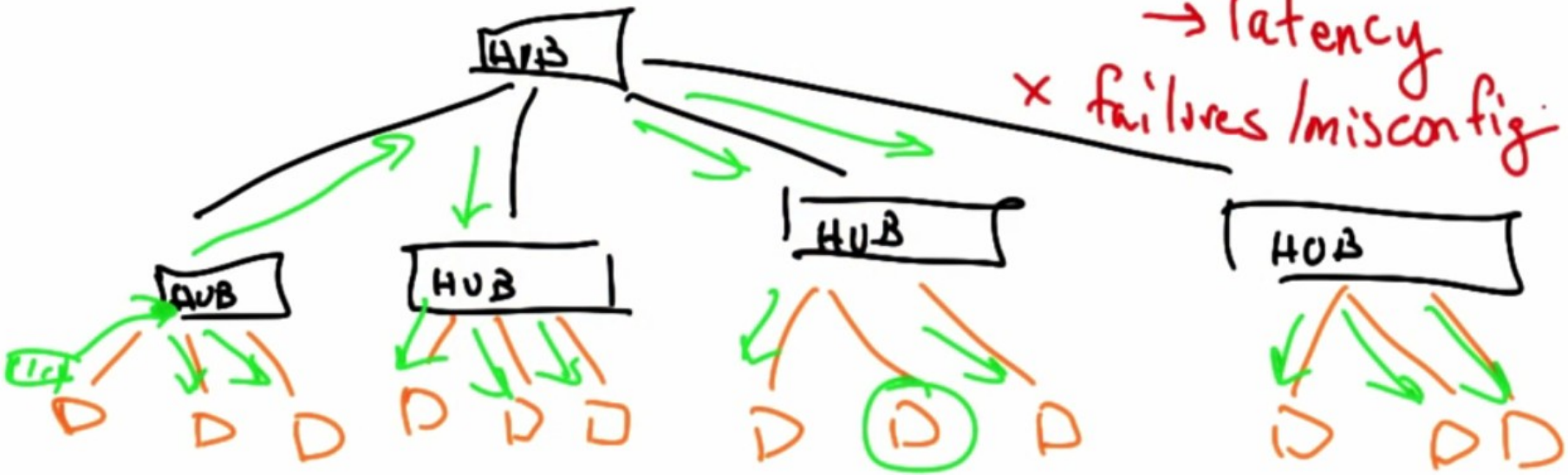
Hub: Broadcast medium



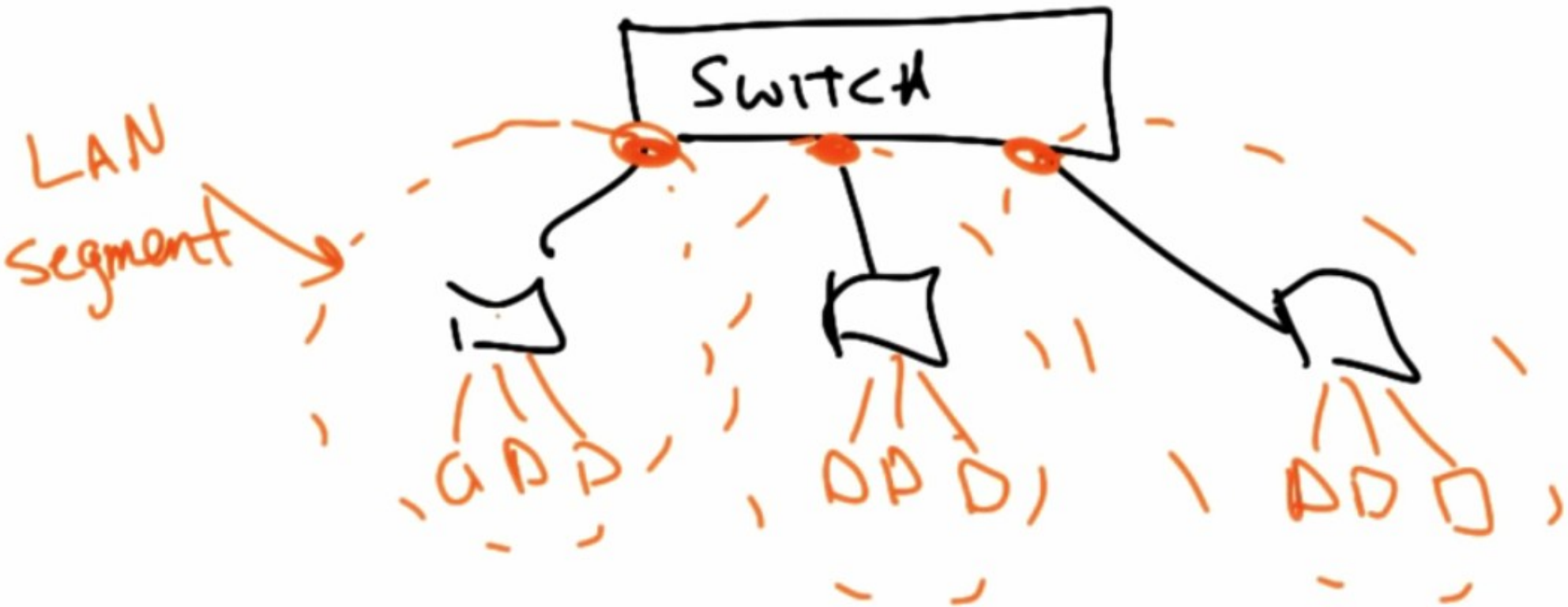
Interconnecting LANs with Hubs

Hub: Broadcast medium

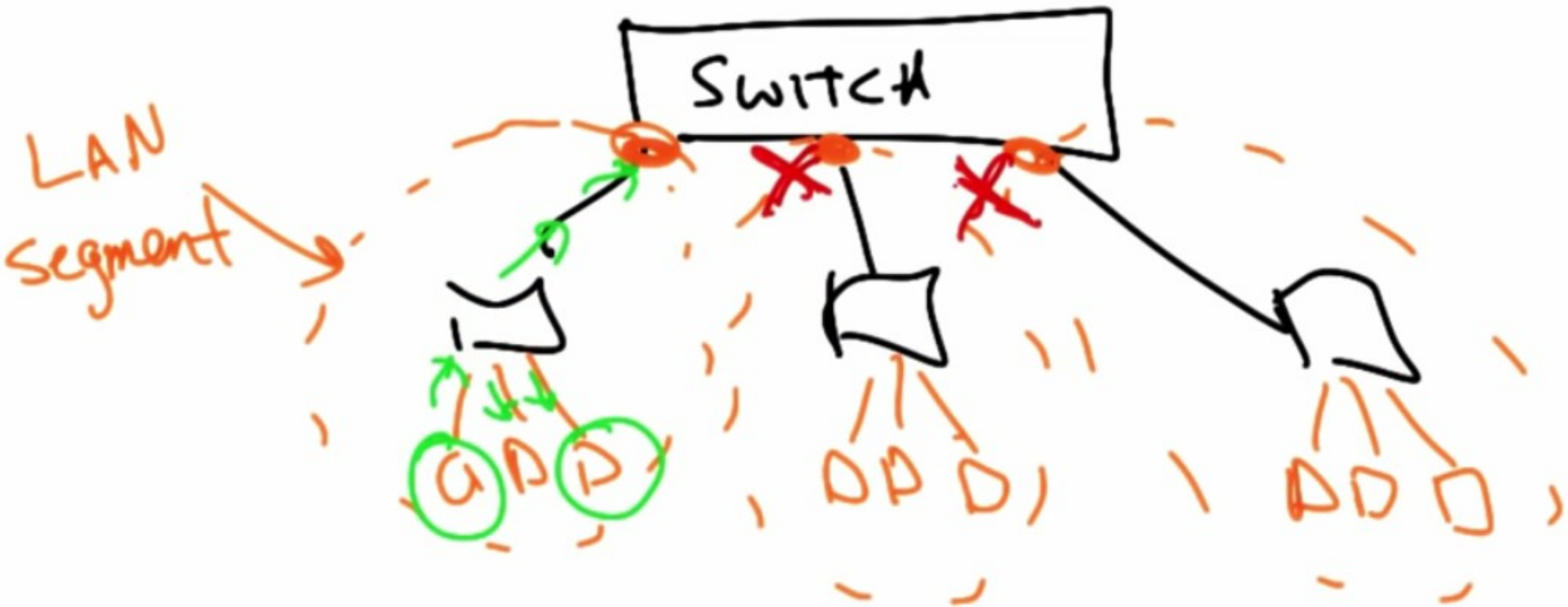
× flooding
× collisions of frames
→ latency
× failures / misconfig.



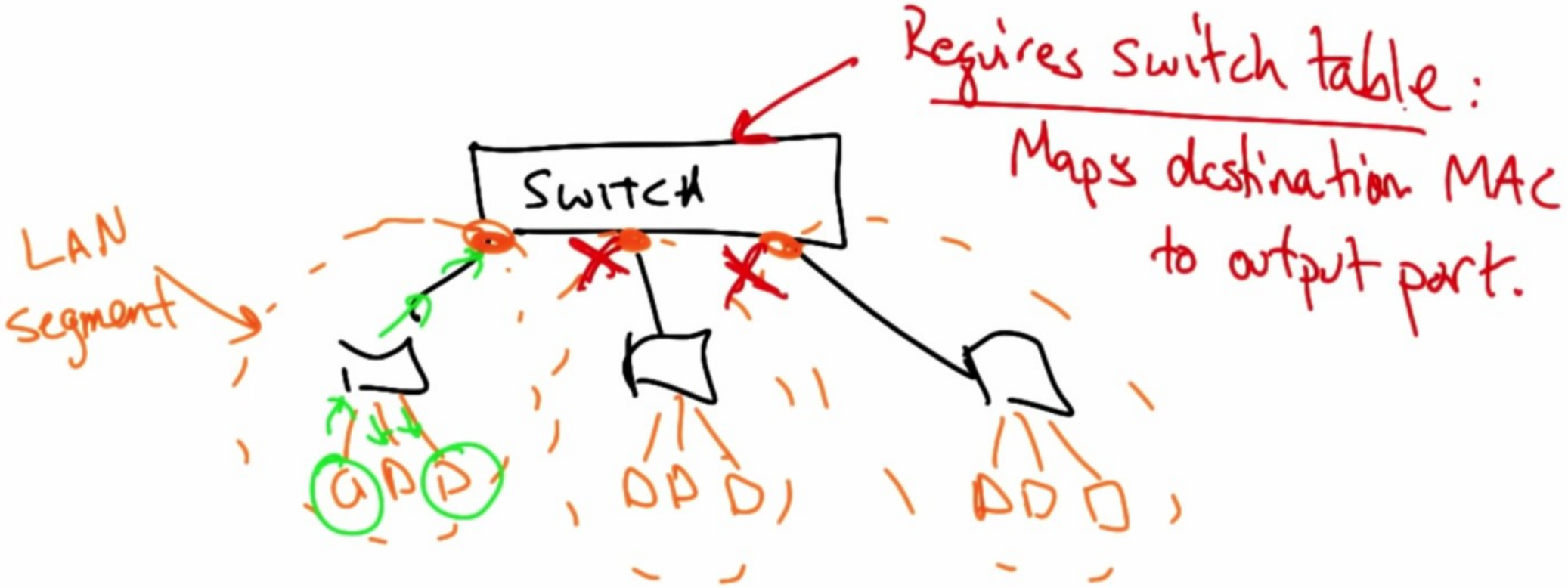
Switches: Traffic Isolation



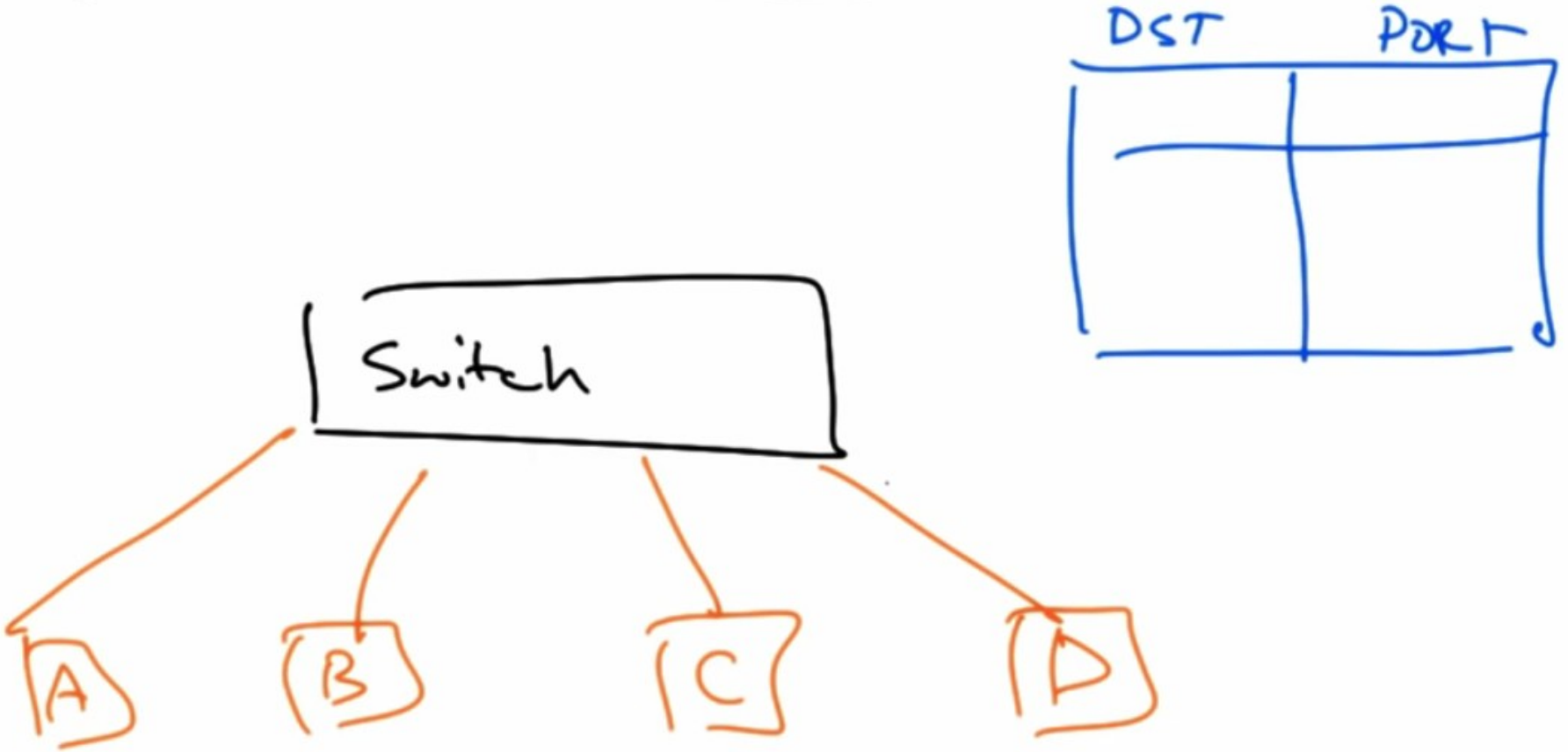
Switches: Traffic Isolation



Switches: Traffic Isolation

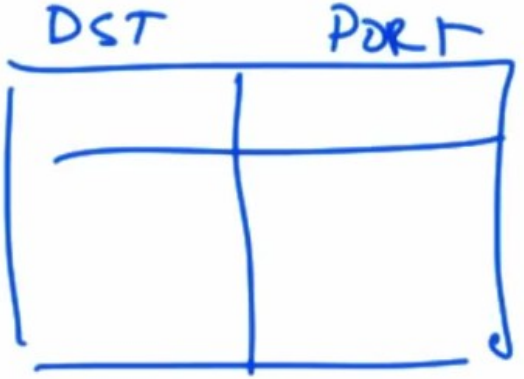


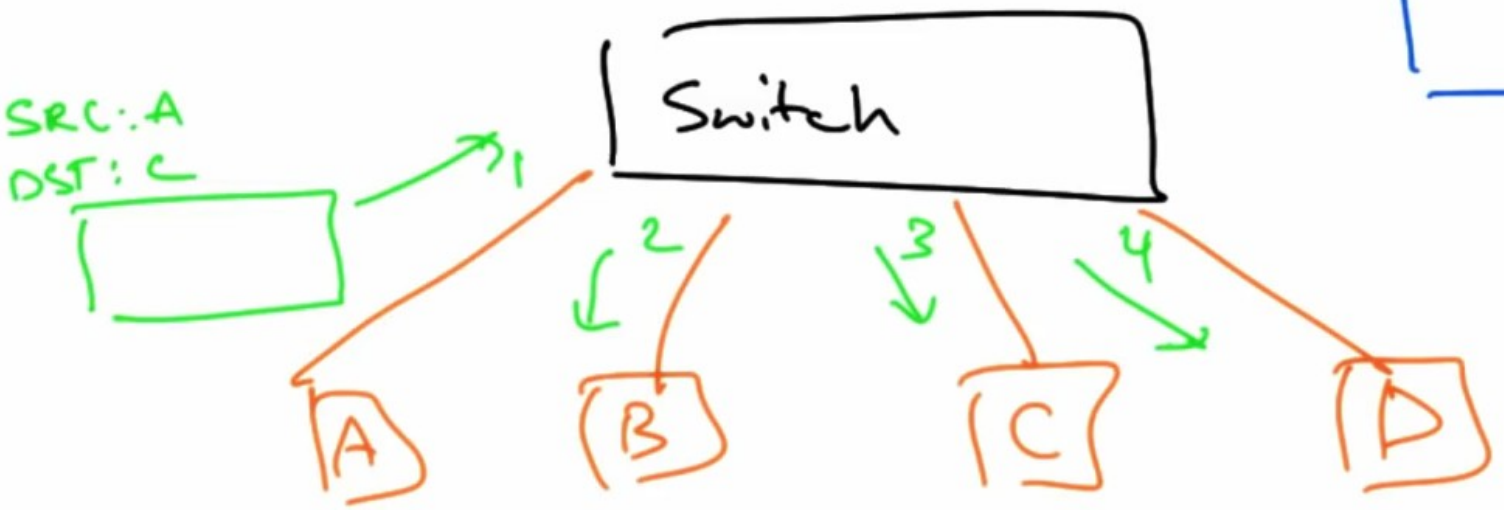
Learning Switches



Learning Switches

① If no entry in forwarding table ⇒ FLOOD

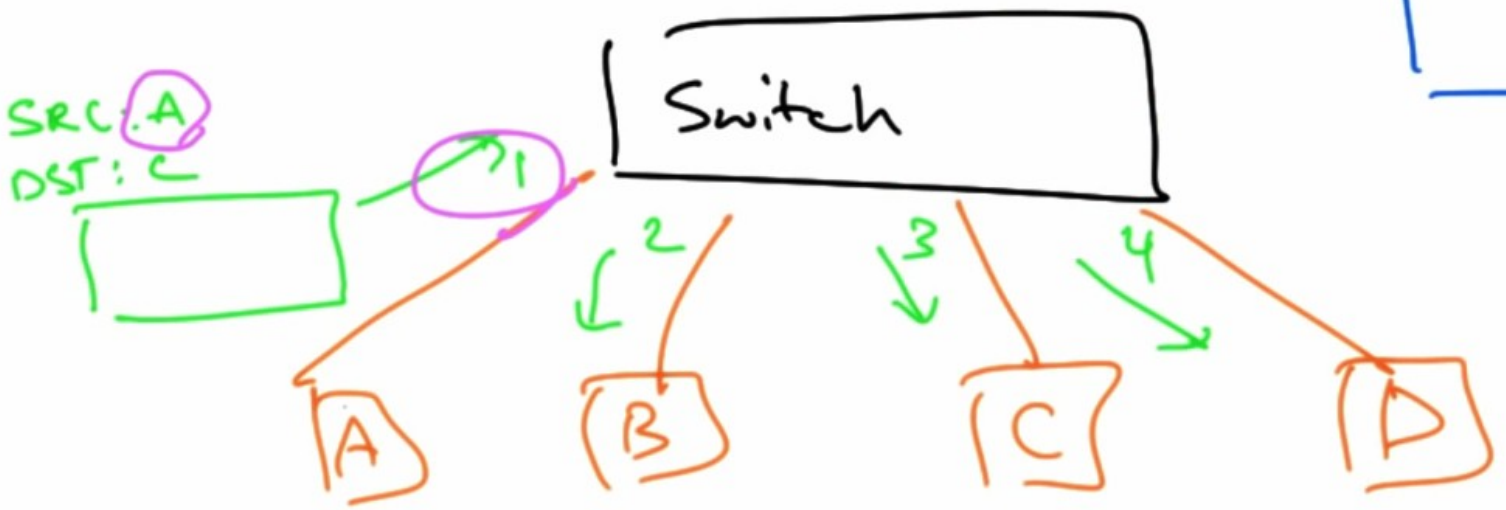




Learning Switches

DST	PORT
A	1

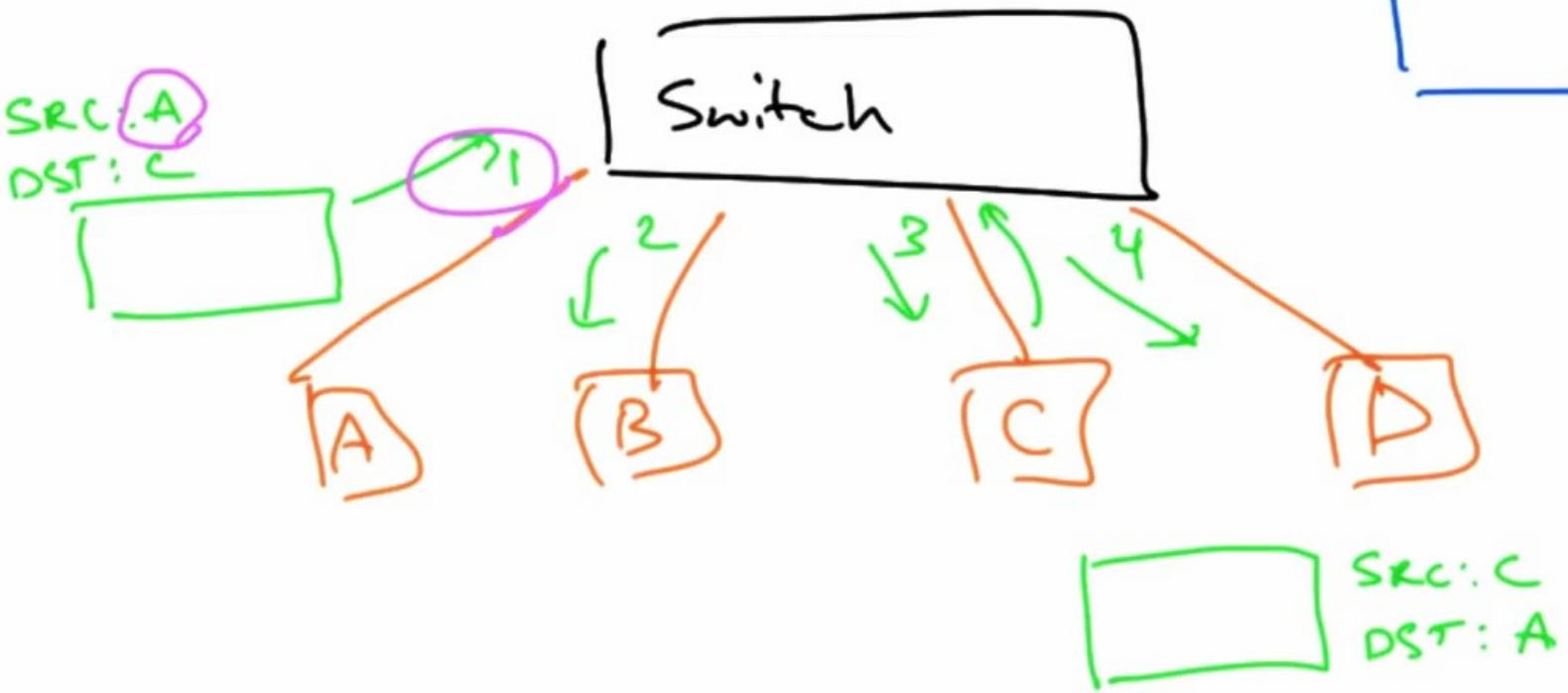
(1) If no entry in forwarding table => FLOOD



Learning Switches

(1) If no entry in forwarding table ⇒ FLOOD

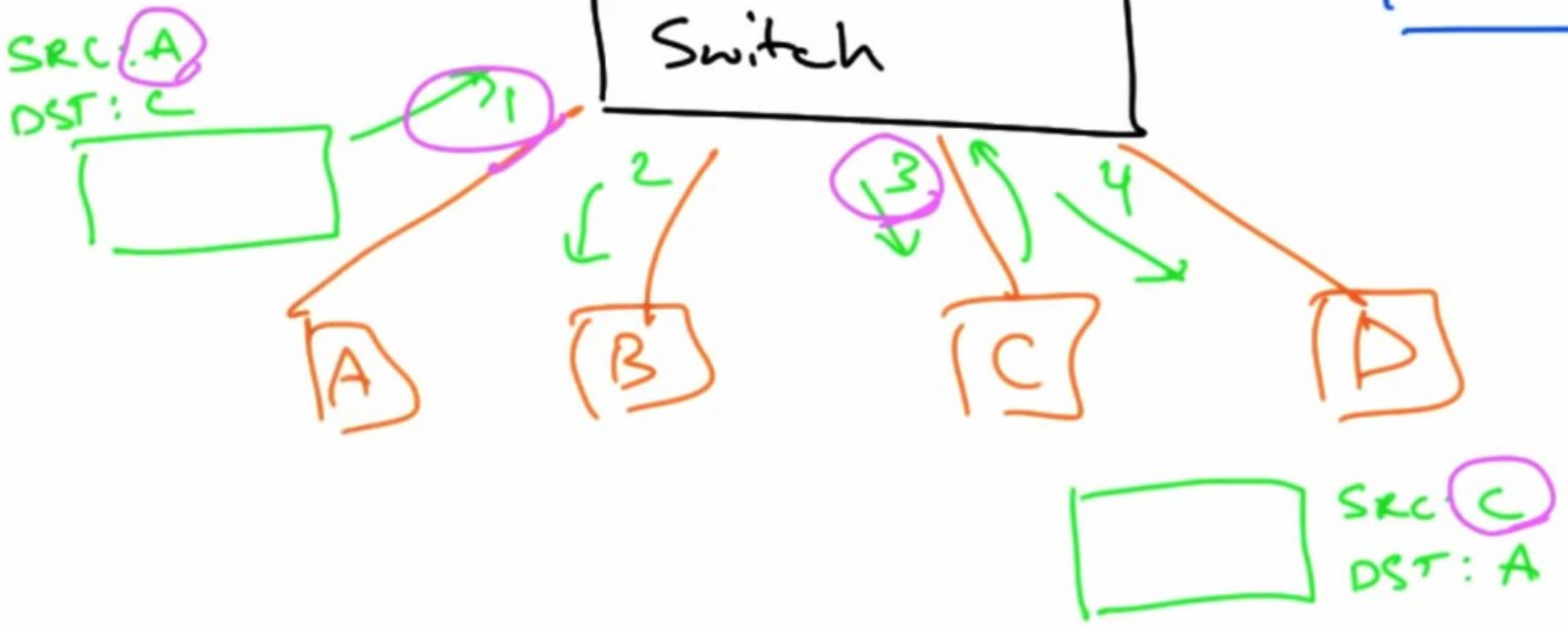
DST	PORT
A	1



Learning Switches

1. If no entry in forwarding table ⇒ FLOOD
2. Otherwise, send to output port in table

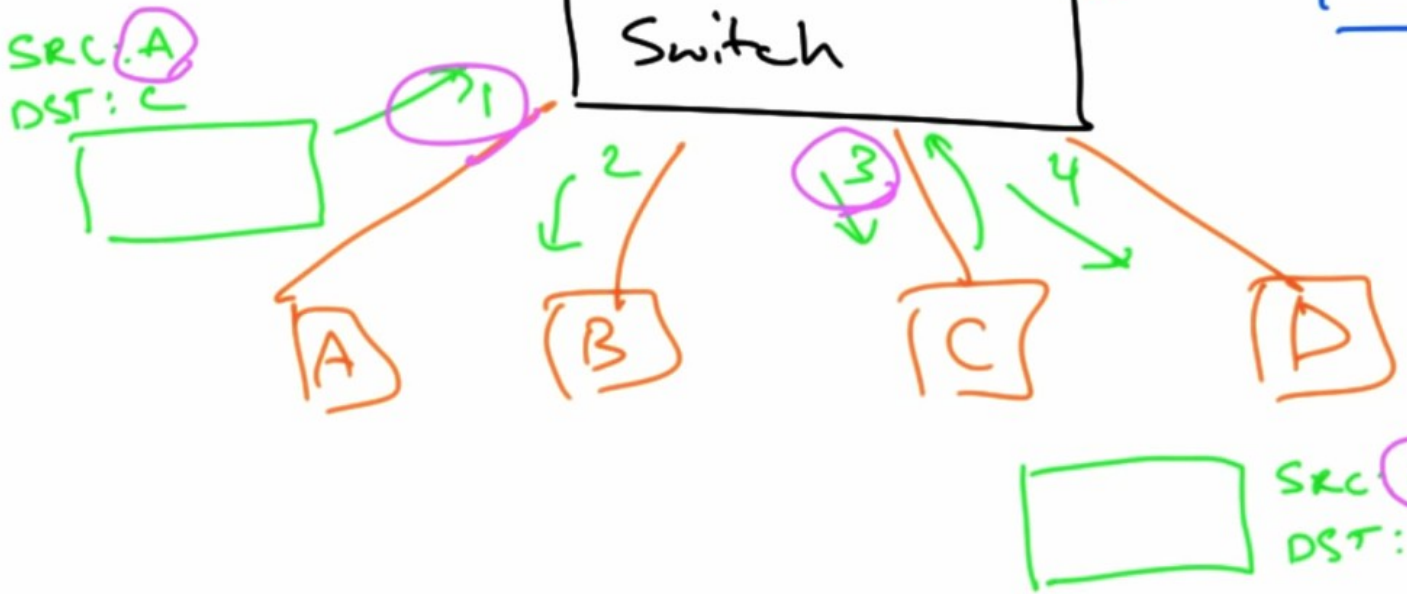
DST	PORT
A	1
C	3



Learning Switches

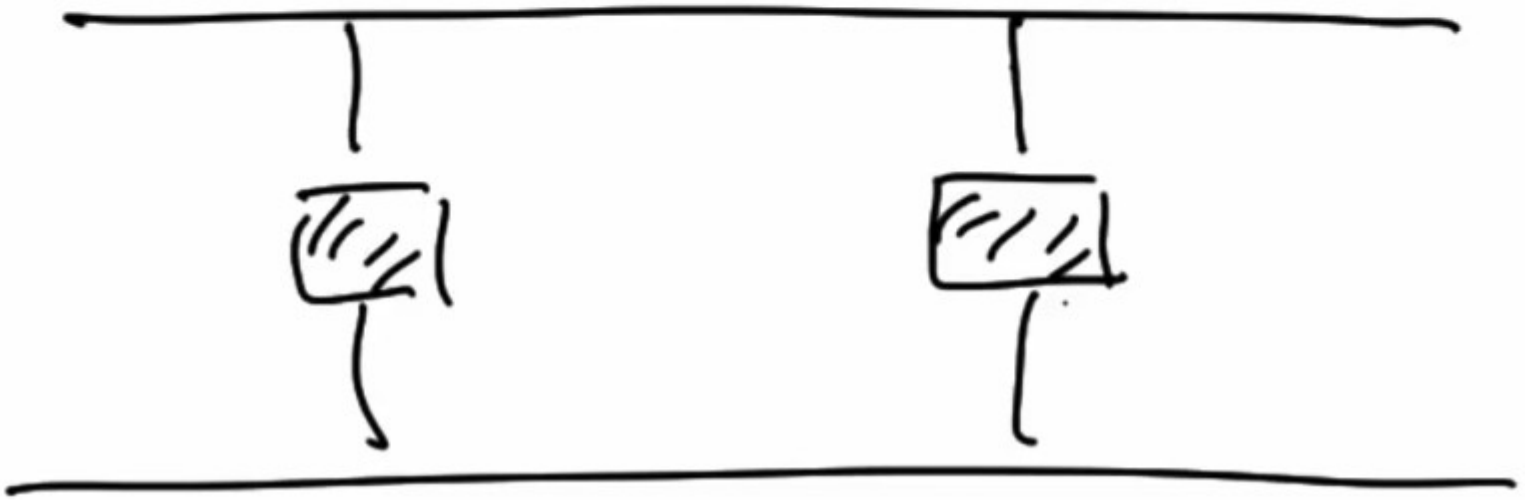
1. If no entry in forwarding table ⇒ FLOOD
2. Otherwise, send to output port in table

DST	PORT
A	1
C	3



+ Broadcast frames (e.g., ARP)

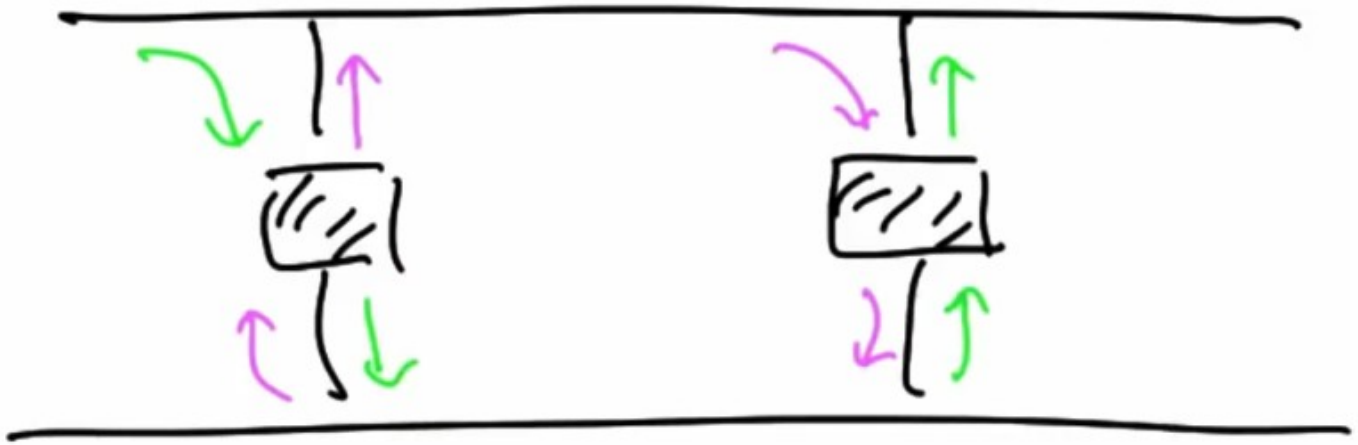
Loops & Broadcasts



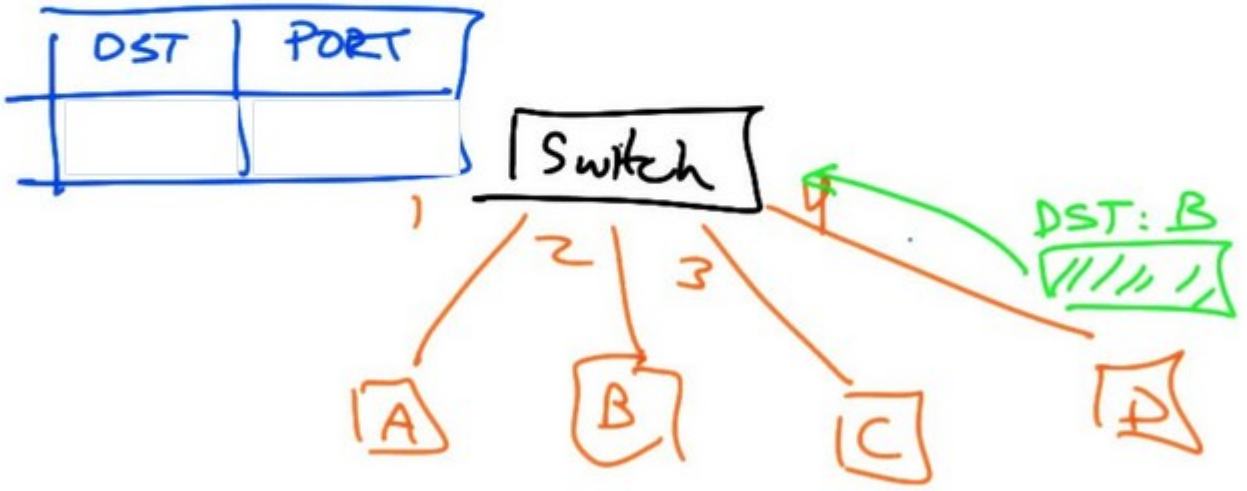
Loops & Broadcasts

Cycles ⇒

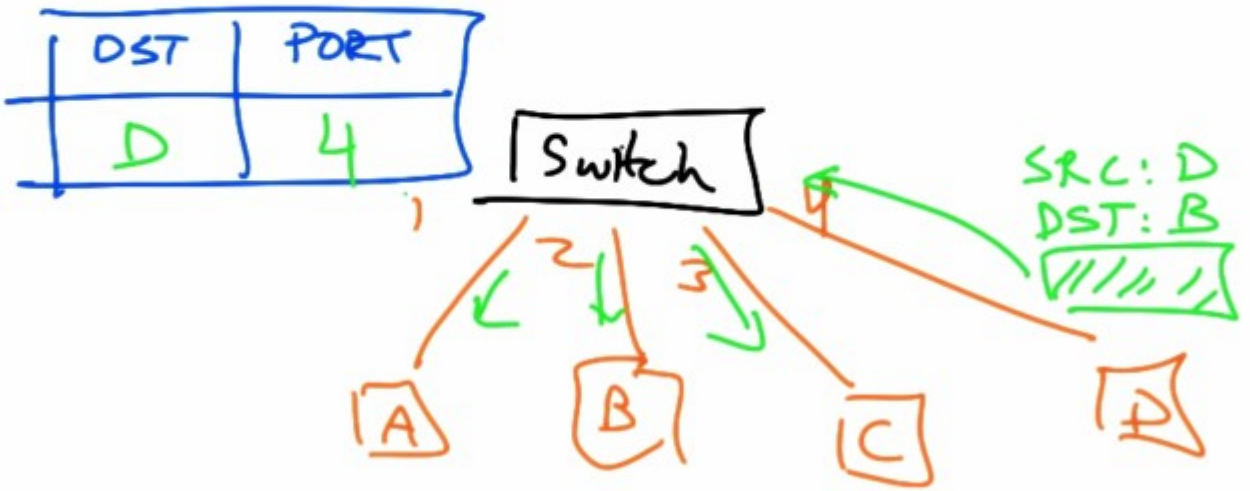
1. Loops
2. Broadcast storms



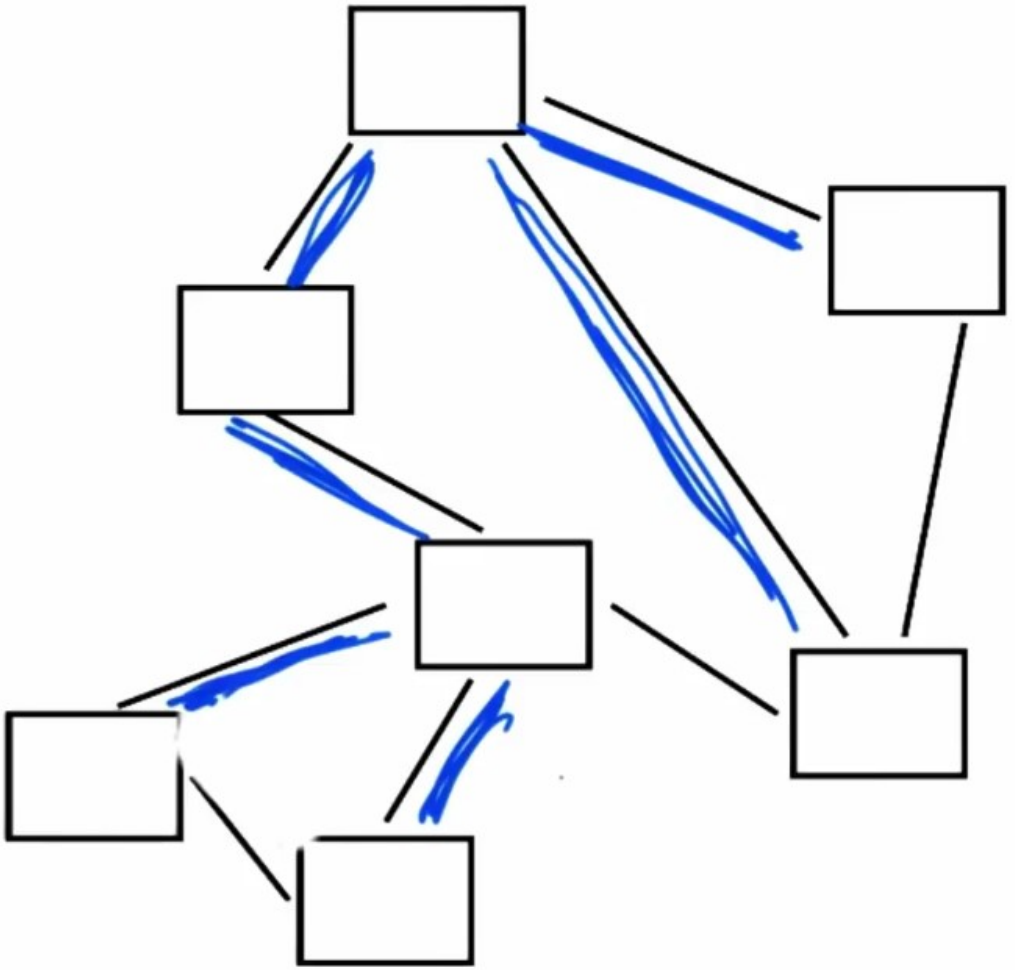
Quiz: Learning Switches



Quiz: Learning Switches

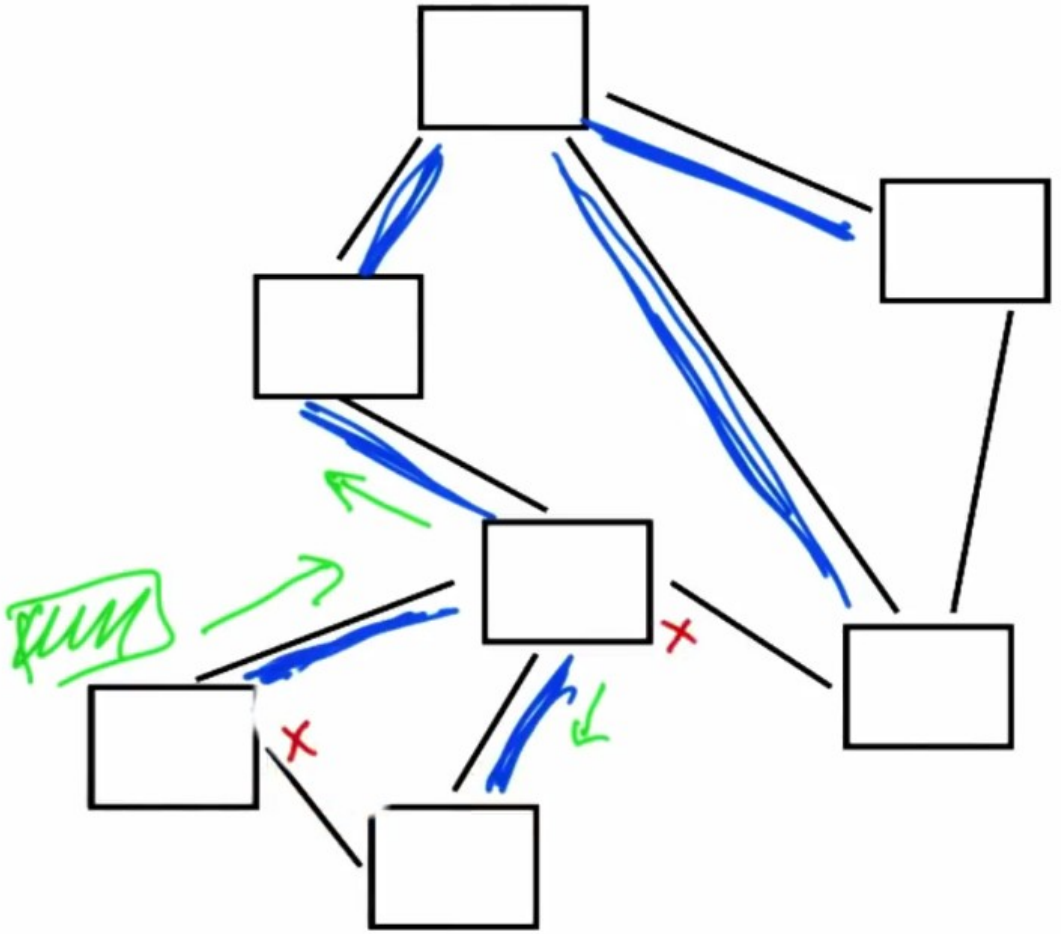


Spanning Tree



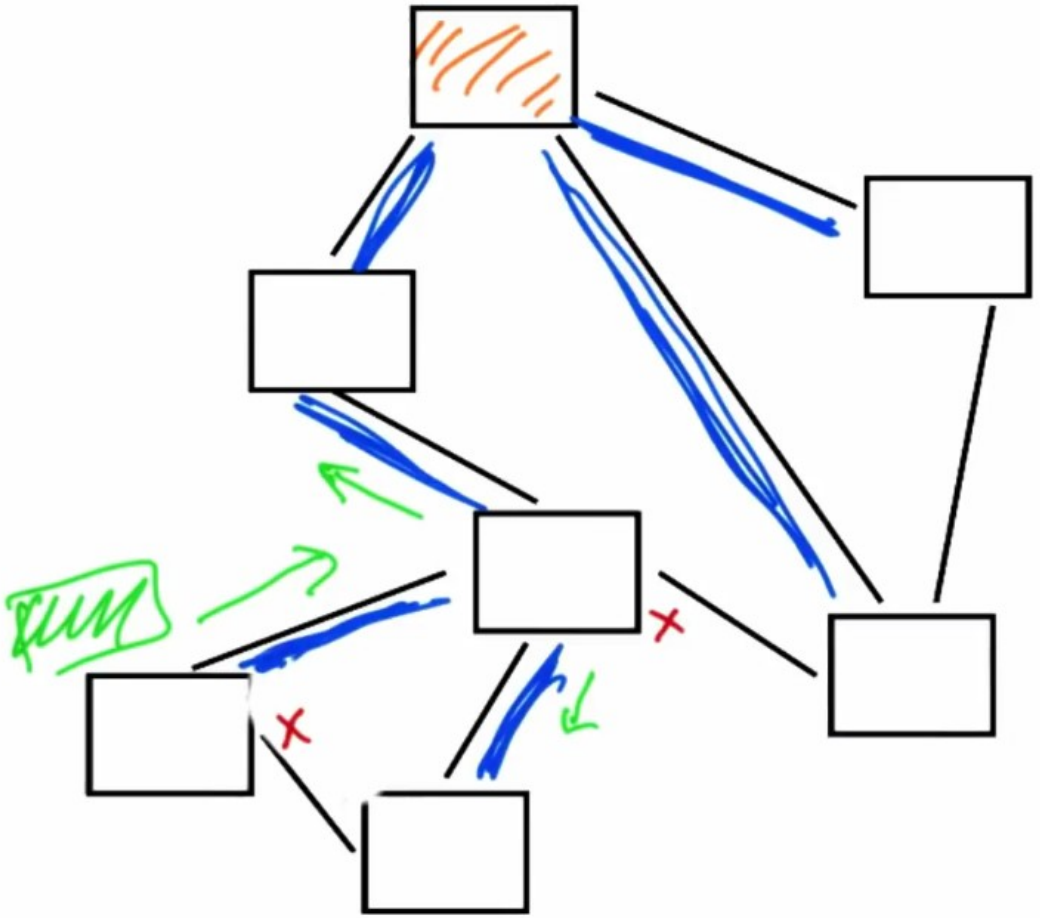
Allow loop-free forwarding on a topology that may contain cycles.

Spanning Tree



Allow loop-free forwarding on a topology that may contain cycles.

Spanning Tree

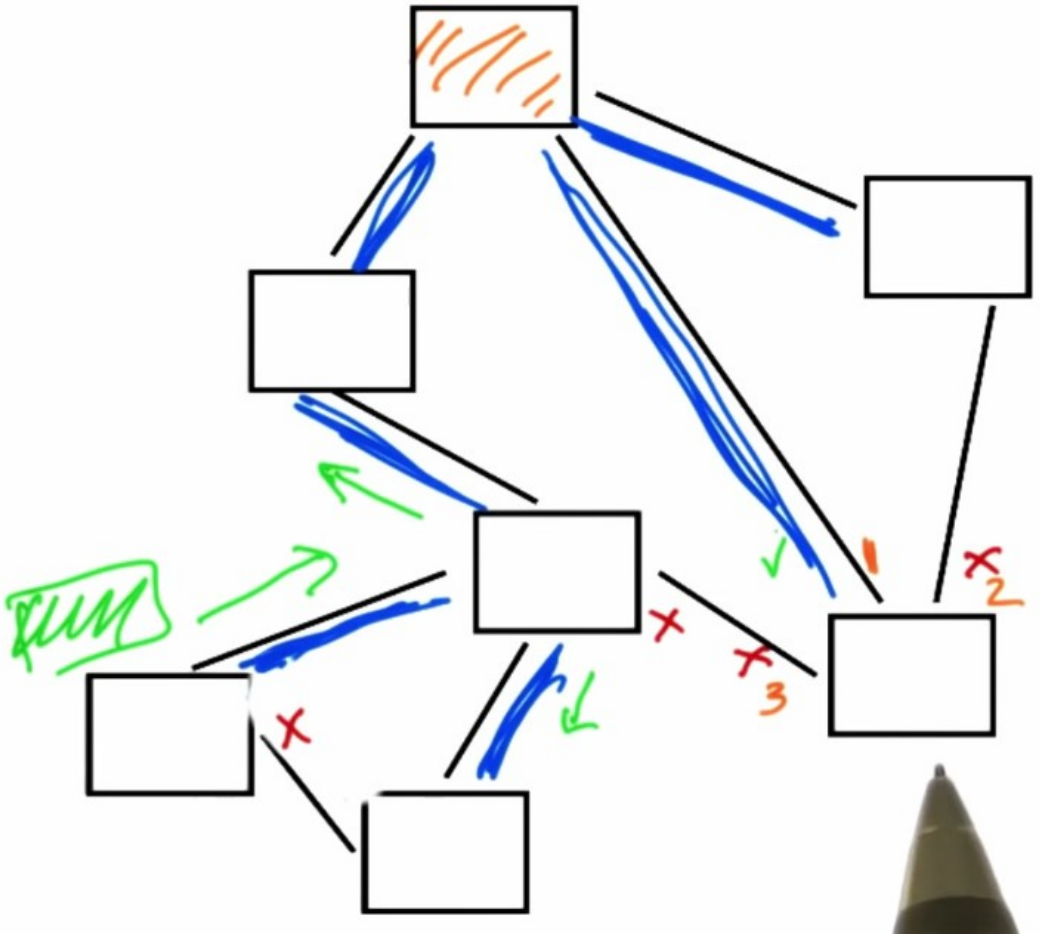


Construct Spanning Tree

1. Elect root (e.g., switch w/ smallest ID)
2. At each switch:

exclude link if not on shortest path to root.

Spanning Tree



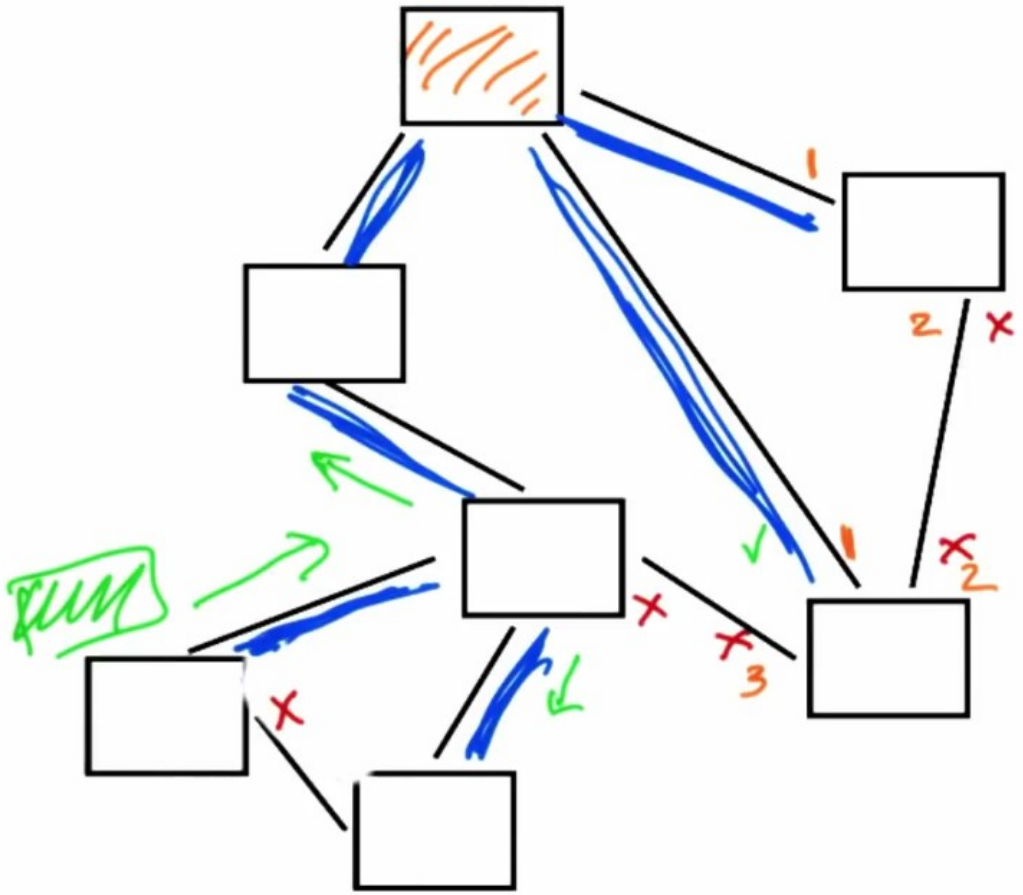
Construct Spanning Tree

(1) Elect root (e.g., switch w/ smallest ID)

(2) At each switch:

exclude link if not on shortest path to root.

Spanning Tree



Construct Spanning Tree

(1) Elect root (e.g., switch w/ smallest ID)

(2) At each switch:

exclude link if not on shortest path to root.

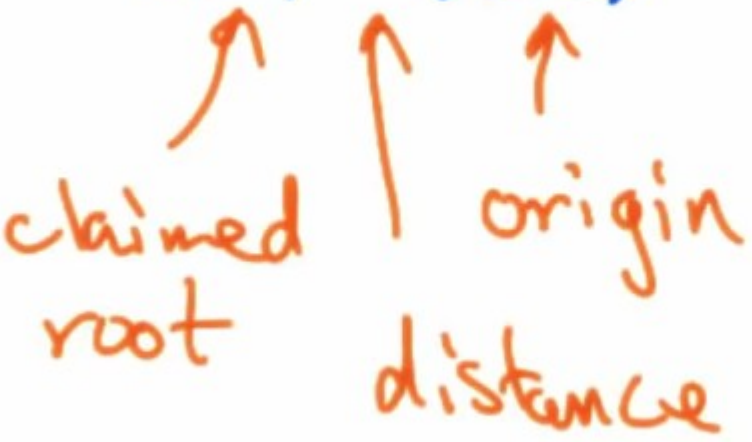
Initially: every node thinks it is the root!

→ (1) update view of root

(2) compute distance to new root

Example

Format: (Y, d, X)



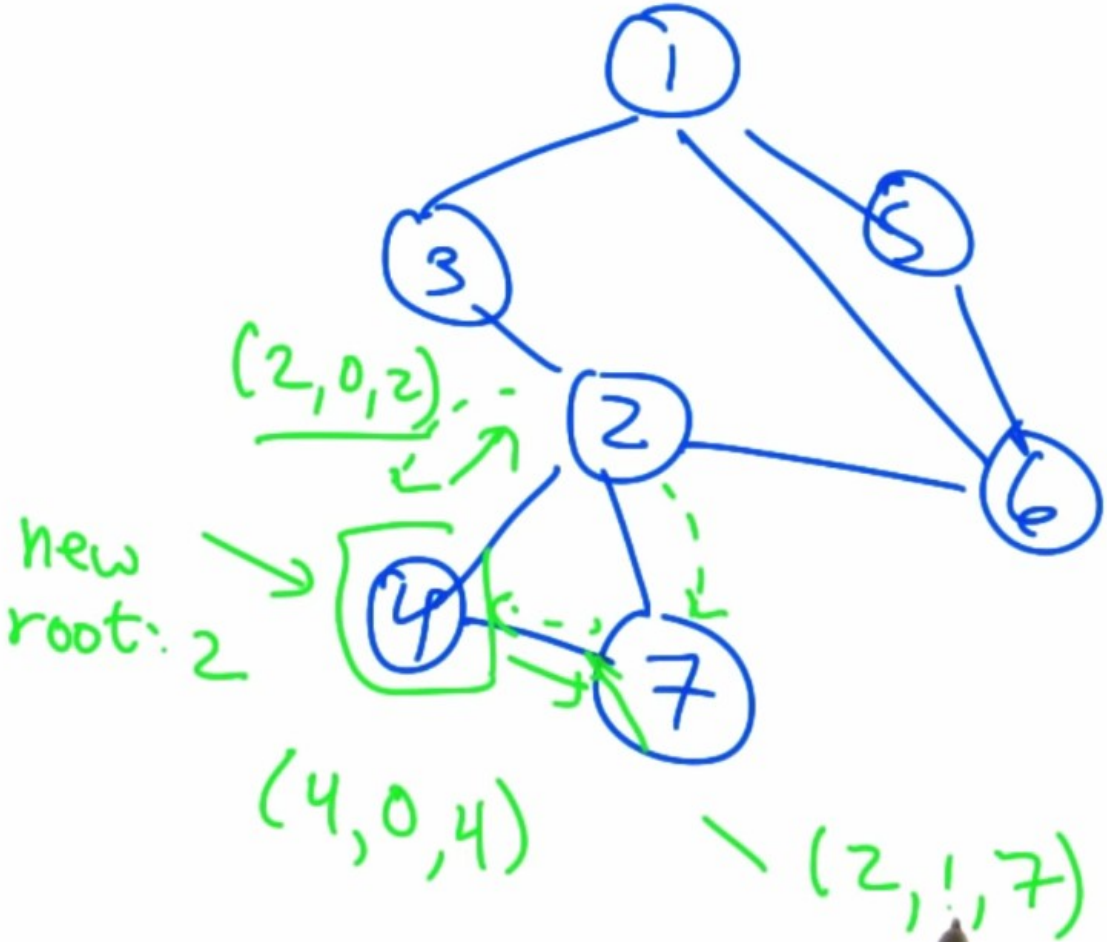
Initially: "$(x, 0, x)$"

Example

Format: (Y, d, X)

- Y: claimed root
- d: distance
- X: origin

Initially: "$(x, 0, x)$"

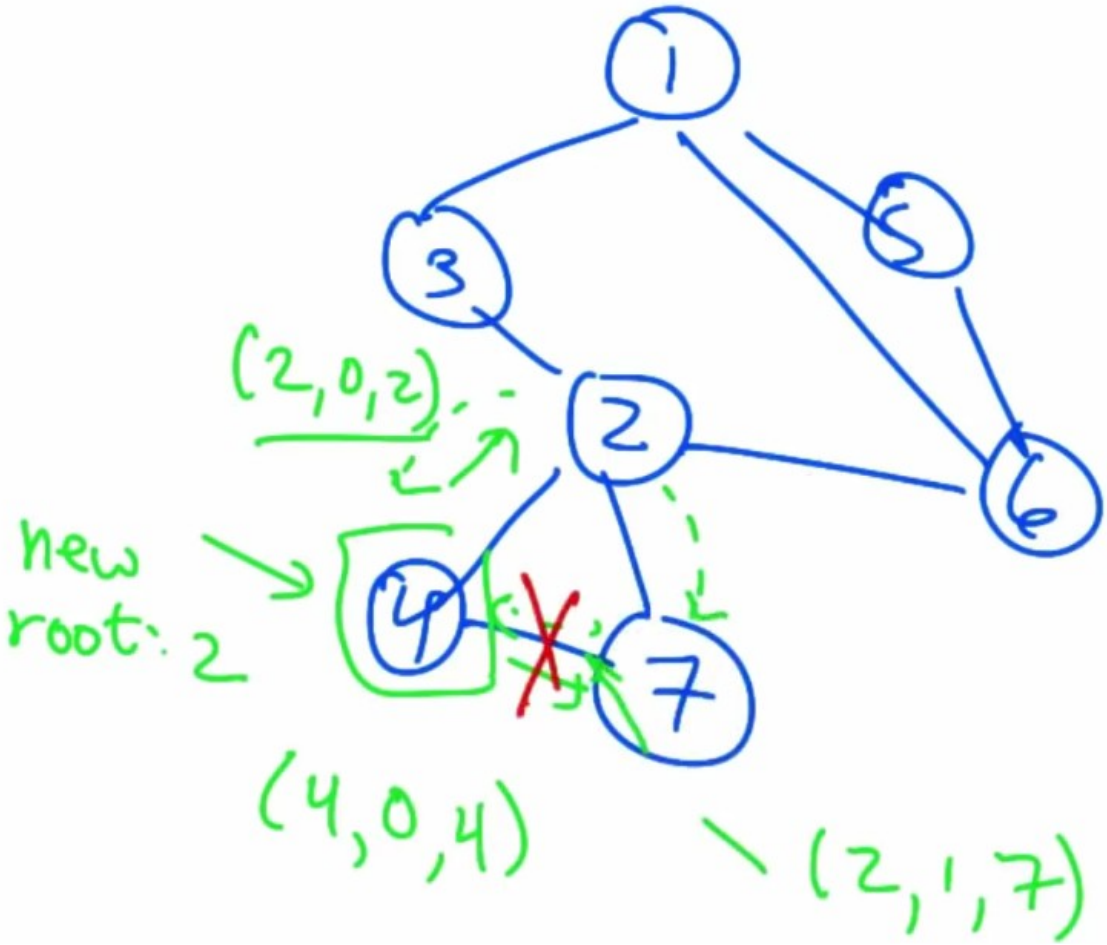


Example

Format: (Y, d, X)

- Y: claimed root
- d: distance
- X: origin

Initially: "$(x, 0, x)$"



Switches vs. Routers

Switches (Layer 2 / "Ethernet")

- Auto-configuring
- Forwarding tends to be fast

Routers (Layer 3 / IP)

- Not restricted to spanning tree

Major limitation

Broadcast

- spanning tree
- ARP queries

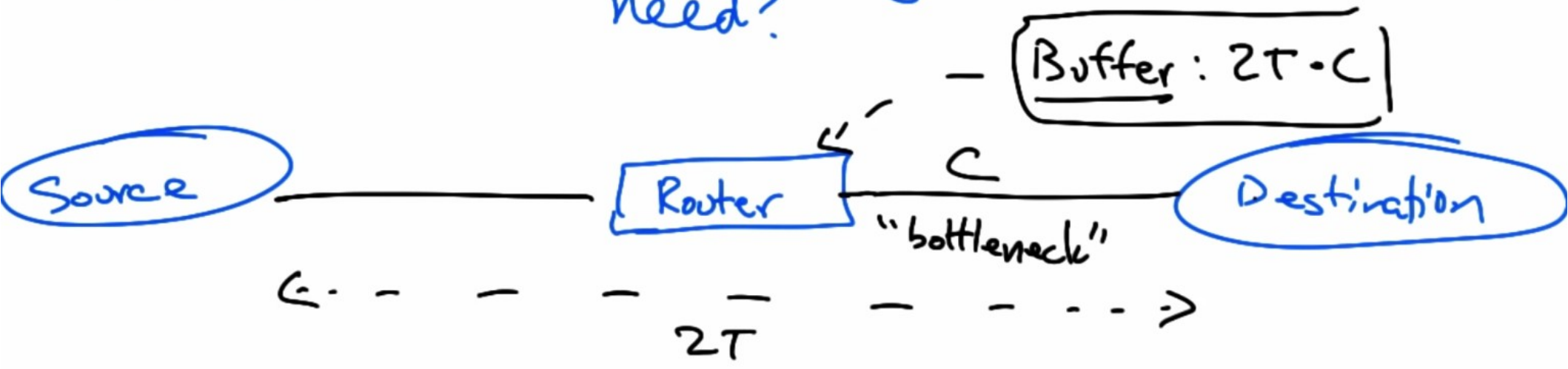
SDN

Buffer Sizing

Question: How much buffering do routers/switches need?

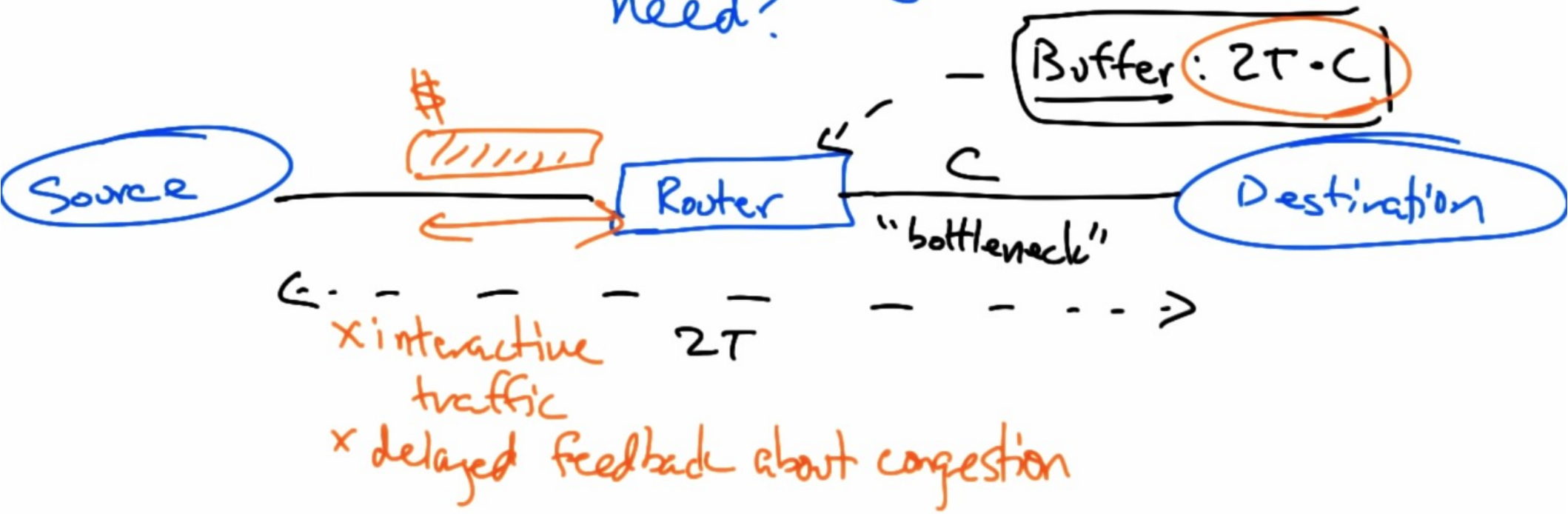
Buffer Sizing

Question: How much buffering do routers/switches need?

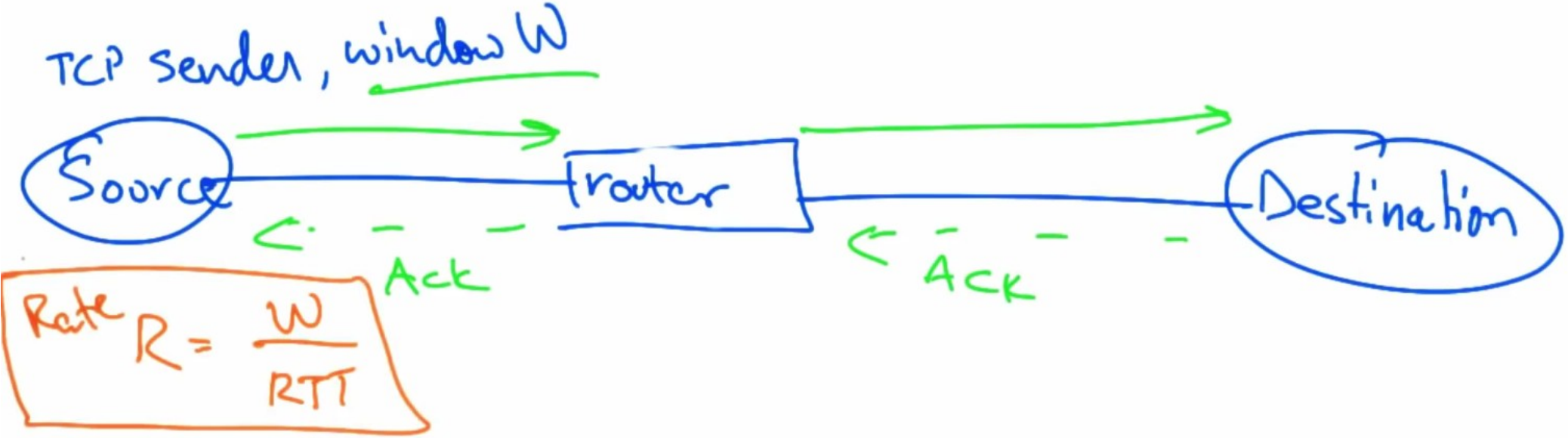


Buffer Sizing

Question: How much buffering do routers/switches need?



Buffer Sizing for a TCP Sender

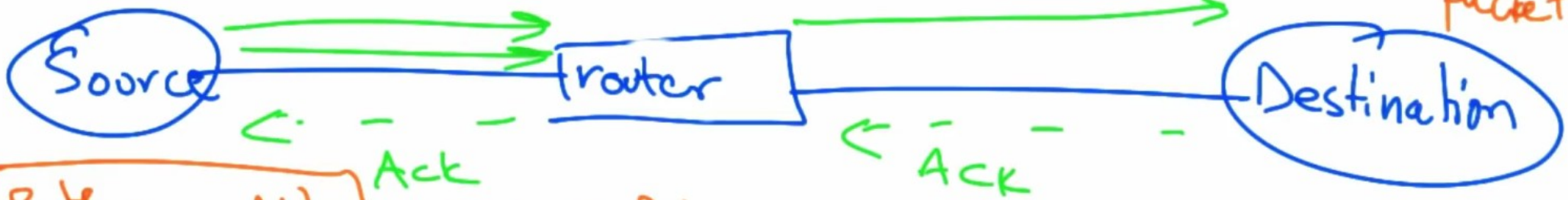


Buffer Sizing for a TCP Sender

TCP sender, window W

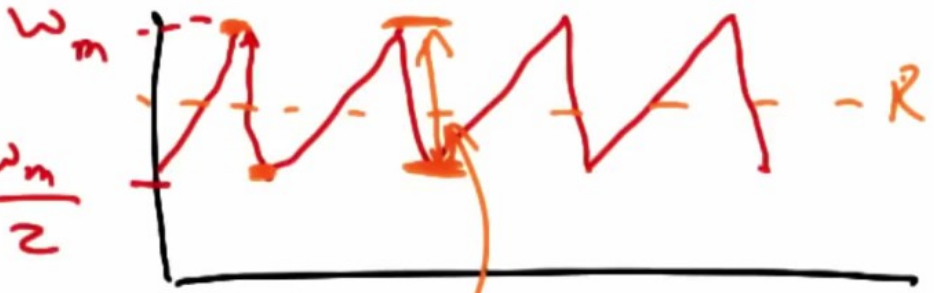
TCP: AIMD

W ACKs → Send W+1 packets



Rate $R = \frac{W}{RTT}$

RTT: T, B/C

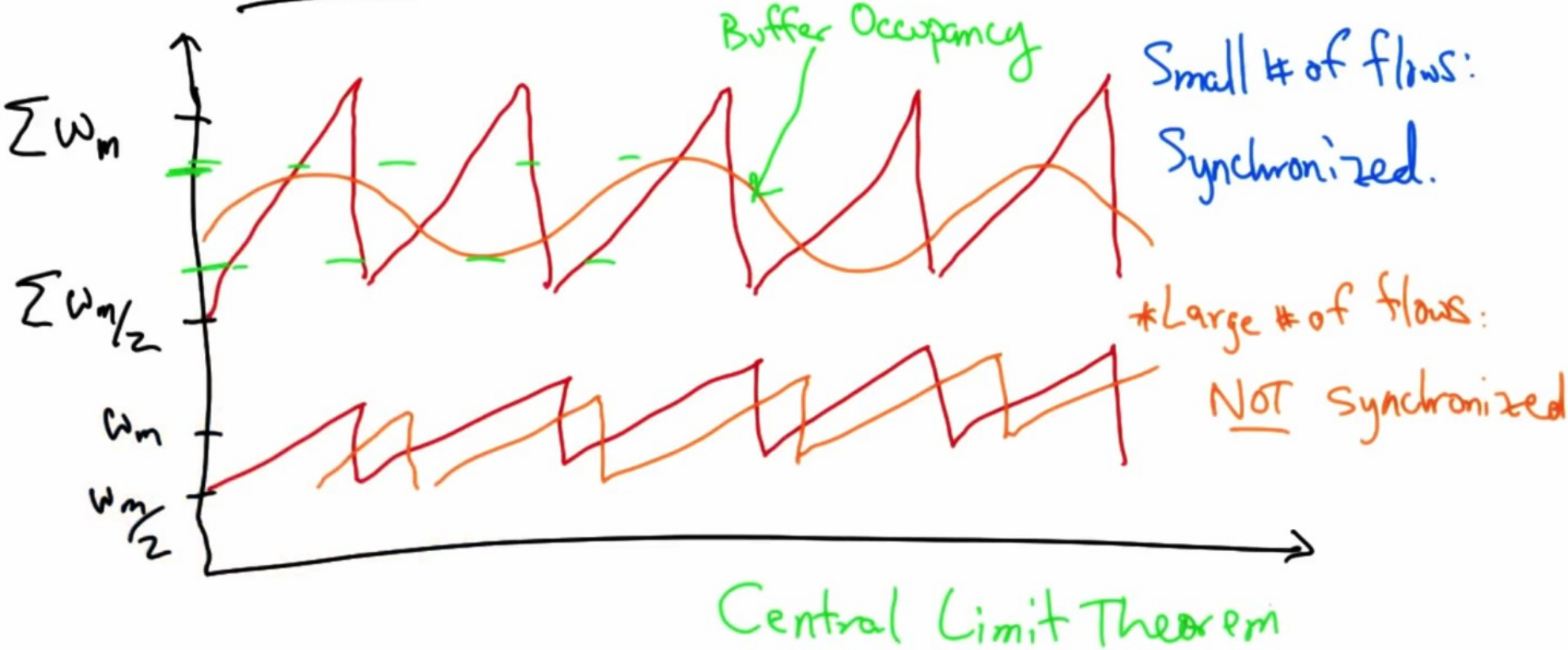


$$\frac{W_{old}}{2T + B/C} = \frac{W_{old}/2}{2T}$$

$$\frac{W_{old}}{RTT_{old}} = \frac{W_{new}}{RTT_{new}}$$

$$B = 2T \cdot C$$

If TCP flows Are Synchronized



Central Limit Theorem (CLT)

More variables ⇒ narrower Gaussian

(congestion windows of flows)

(fluctuation of sum of all of the congestion windows)

$1/\sqrt{n}$

$2T \cdot C \longrightarrow \frac{2T \cdot C}{\sqrt{n}}$