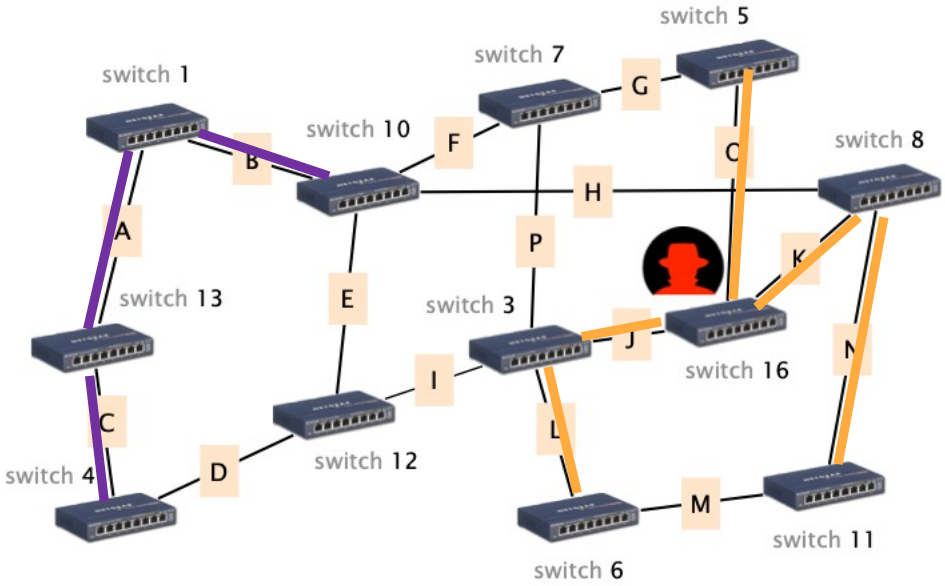
# CommNet Q&A Session: Answering received questions

Georgia, Yu
23/07/24

# Questions on

- **Ethernet/IP**
- Routing
- Transport
- Applications
- Other

# Ex. 3.1 b) – Tie breaking in STP upon receiving equal-cost BDPUs to same root from diff switches

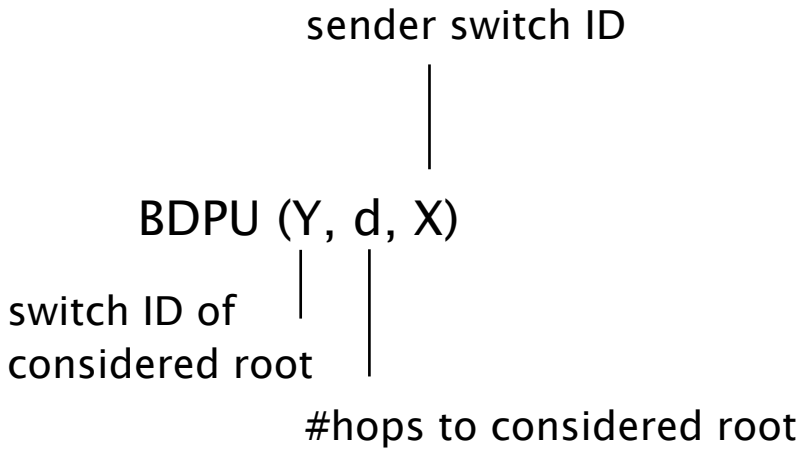



Each of switches 12 and 7 has equal cost (2) from both legitimate and attacker root switches

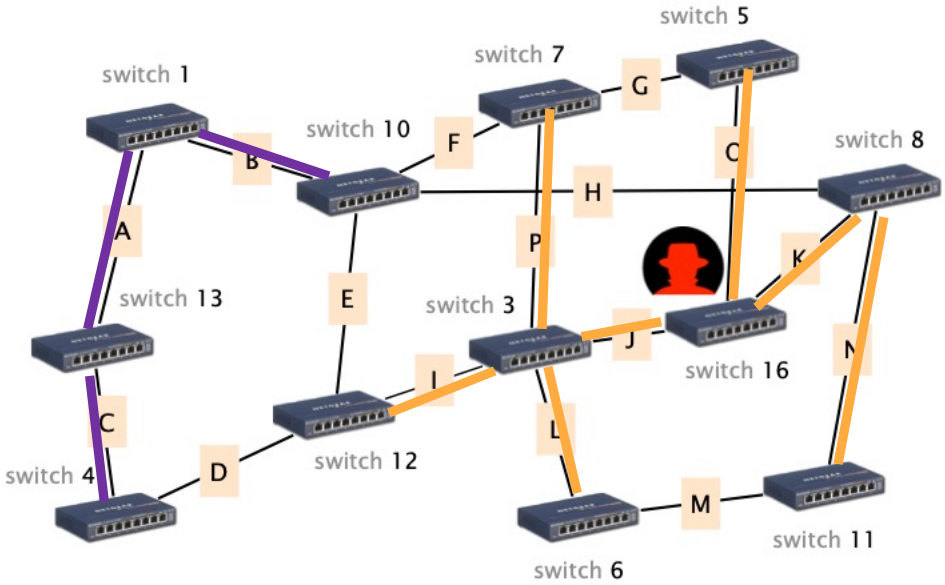
- are links I & P necessarily on attacker's tree?
- does it depend on attacker's MAC address?

# Ex. 3.1 b) – Tie breaking in STP upon receiving equal-cost BDPUs to same root from diff switches

- Pick the BPDU with lower sender switch ID



# Ex. 3.1 b) – Tie breaking in STP upon receiving equal-cost BDPUs to same root from diff switches




Each of switches 12 and 7 has equal cost (2) from both legitimate and attacker root switches

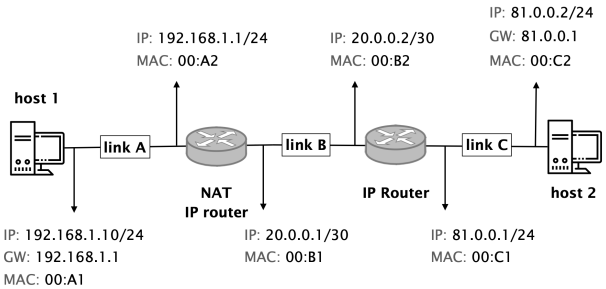
- are links I & P necessarily on attacker's tree?
- does it depend on attacker's MAC address?

- Yes, I & P are necessarily on attacker's tree, due to STP's tie-breaking (see previous slide)
- No, it does not depend on attacker's MAC, but on senders'/neighbors' switch IDs (may or may not be the MAC)
  - e.g., switch 12 receives BDPU (1, 1, 3) and BDPU (1, 1, 10) and picks to connect through switch 3 (3<10)

## Ex. 99.12 – Does a router replace the MAC address in a packet with its own when forwarding packets from Internet to a client in local network and vice versa?

- Yes, whenever a router forwards a packet in network X,
  it uses its own MAC attached to network X
  as the src MAC in the packet header

# Ex. 99.13 – Why do MACs and ports change along the path of a packet?




| | src MAC | dst MAC | src IP | dst IP | src TCP port | dst TCP port |
|---|---|---|---|---|---|---|
| link A | 00:A1 | 00:A2 | 192.168.1.10 | 81.0.0.2 | 1337 | 80 |
| link B | 00:B1 | 00:B2 | 20.0.0.1 | 81.0.0.2 | rand | 80 |
| link C | 00:C1 | 00:C2 | 20.0.0.1 | 81.0.0.2 | rand | 80 |

What will be the src/dst MACs, IPs, ports of a packet as it goes from host 1 to host 2?

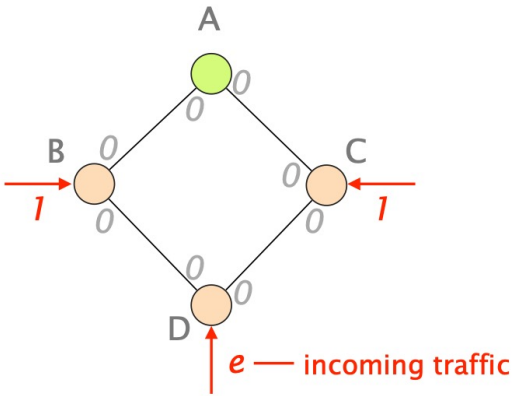
here, MACs change across links as each link is in a **different local network** and forwarding within a network is done using MACs of local devices

in general, ports do not change along a packet's path;
here, src ports **change due to NAT** (to ensure unique mappings even if multiple hosts use same local port & communicate with same remote process)

# Questions on

- Ethernet/IP
- **Routing**
- Transport
- Applications
- Other

# Ex. 1.9 – Forwarding w/ dynamic, load-based weights; Where does the “extra load” come from?



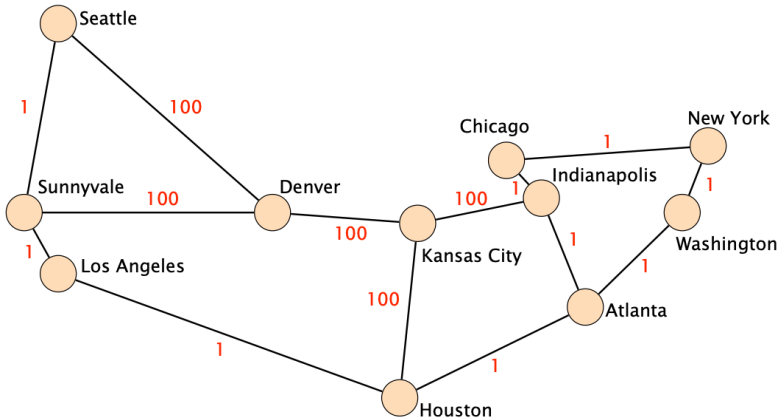

Network topology with directional link weights.

| | Link Load | | | | | | | | Next Hop | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | A→B | A→C | B→A | B→D | C→A | C→D | D→B | D→C | B | C | D |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | A | A | B |
| 1 | 0 | 0 | 1 + e | 0 | 1 | 0 | e | 0 | D | A | C |
| 2 | 0 | 0 | 0 | 1 | 2 + e | 0 | 0 | 1 + e | A | D | B |

sum load: 2 + 2e ≠ sum load: 4 + 2e

The sum loads differ as the same load may **cross multiple links** on a path;
e.g., in step 2, load e crosses both D->C and C->A

# Ex. 1.10 c) – Configure link weights so that *only* Denver-Kansas uses direct link




In the solution, Denver-Kansas link
has a weight of 100;
is that necessary? could it be 1?

No, 100 is not necessary, but it cannot be 1:
it must be ≥5, so that Indianapolis-Denver
does not go through Kansas-Denver
(we must ensure that all pairs that
must cross at least one 100-weight link,
are not using Denver-Kansas)

# Explain Poisoned Reverse; Do routing tables have inf even if there is no failure?

- Poisoned Reverse
  - If the path from router z to router x goes through router y, then router z tells router y that its cost to router x is infinity
- Indeed, this results in router y installing an infinity in its routing table, which indicates that y cannot route to x through z

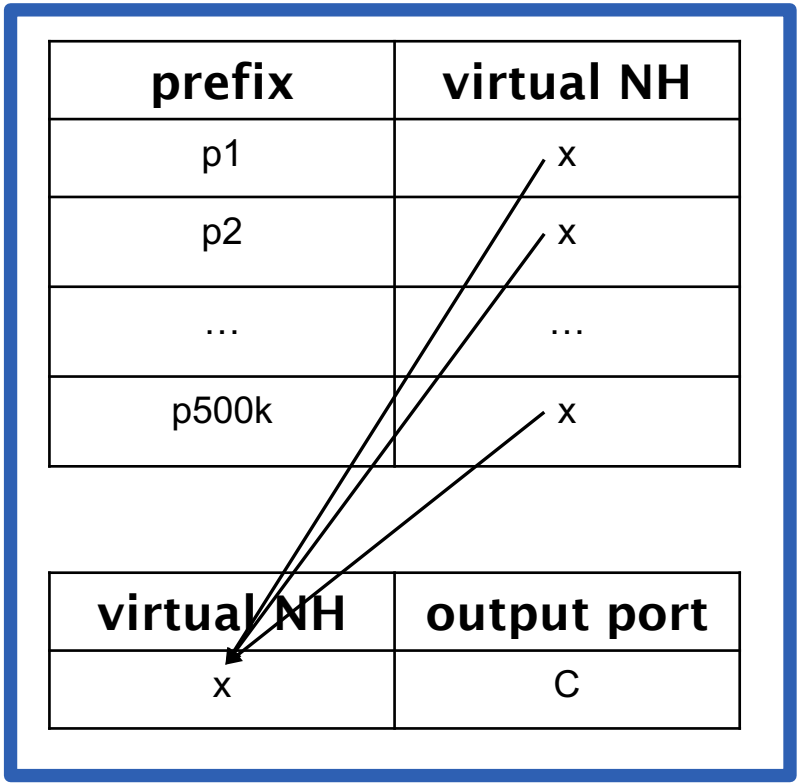
(see also slides from Week 8's exercise session)

# Ex. 99.2 b) – What is a virtual next hop? How is it used?

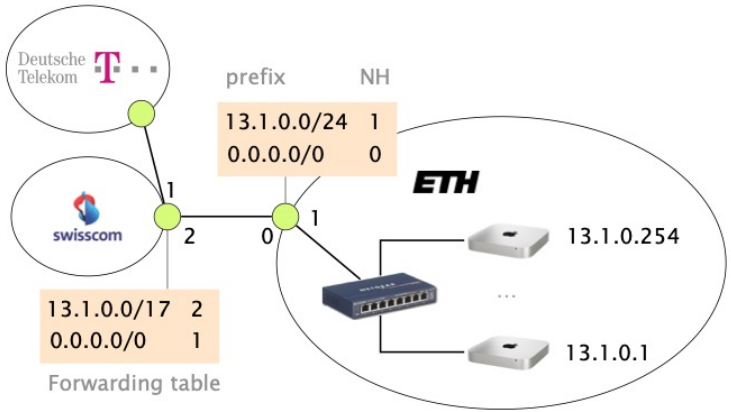
- A virtual next hop is a **reference** to another table (in the same router)
- We use it to reduce the number of required updates upon failure
  - for all prefixes that are routed the same,
    we can map them to the same virtual next hop,
    and the virtual next hop to a port
  - when port goes down, we only need to **update the port in one entry**
    instead of updating the port in multiple entries, one per prefix routed through the output port

# Ex. 99.2 b) – What is a virtual next hop? How is it used?

| prefix | output port |
|---|---|
| p1 | C |
| p2 | C |
| … | … |
| p500k | C |



# Ex. 4.2 d) – Forwarding misconfigurations




Where are my IP packets going?

Swisscom receives pkt for 13.1.66.1 from DT; pkt will be looping between Swisscom/2 & ETH/0 (until TTL=0)

- Why does ETH send the packet back to Swisscom? Isn't ETH's responsibility to drop packets destined to 13.1.66.1?

- Option 1): ETH is misconfigured; it shouldn't send back pkts for its /17, instead, its border router would drop it after failing to resolve IP through ARP
- Option 2): Swisscom & DT are misconfigured; they should only route towards ETH pkts for the /24

# Questions on

- Ethernet/IP
- Routing
- **Transport**
- Applications
- Other

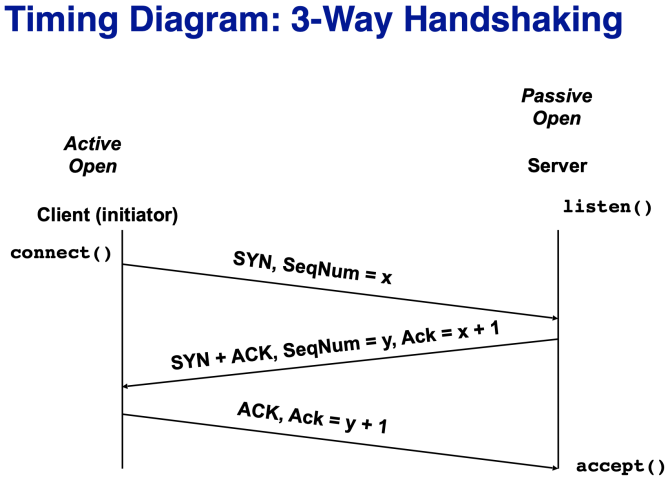
# Difference between sockets and UDP/TCP ports

- Socket
  - “an operating system (OS) abstraction which
    provides applications with a uniform interface to the network”
  - so, a socket is associated with a process,
    is created and lives in the OS, and maintains state
- UDP/TCP port
  - socket identifier (along with IPs)
    put in the packet so that OS knows to which process to deliver data

(see also lecture slides from Week 11)

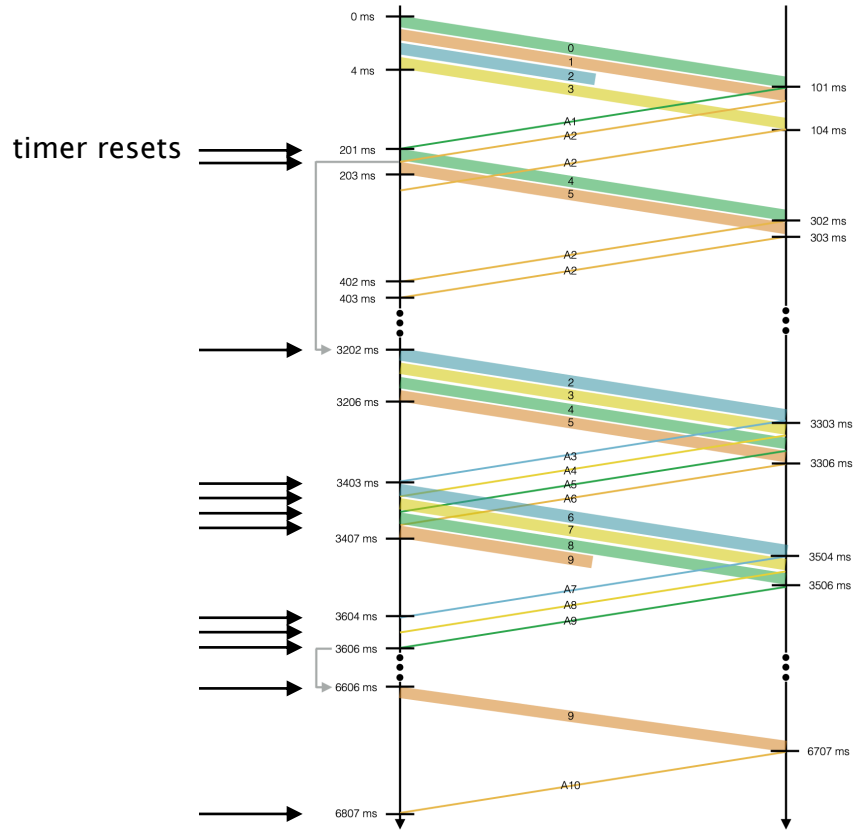
# Ex. 9.5 – Why does the TCP handshake take 1RTT and not 1.5RTT?

- TCP connection establishment takes 1.5RTTT
- But from client's perspective,
  0.5RTT of the above time is "hidden"
  - client can **already send data**
    on the third segment (ACK)
    i.e., after 1RTT

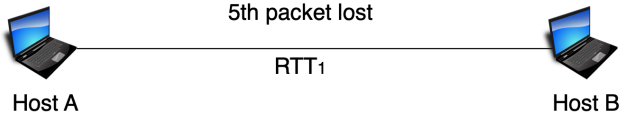


# Ex. 99.8 – Clarify when does timer reset in GBN

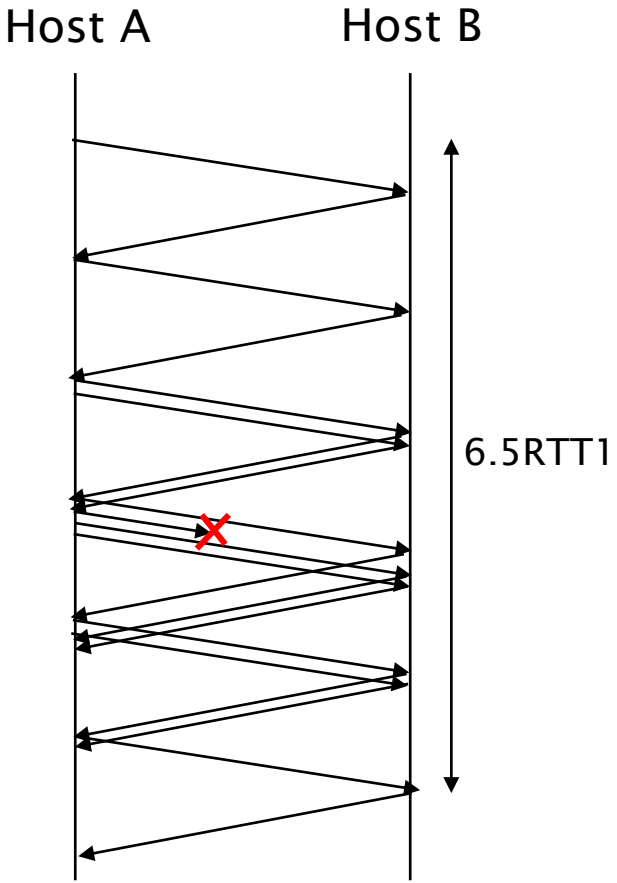
- Single timer that resets
  - at each *new* ACK or
  - immediately after a timeout



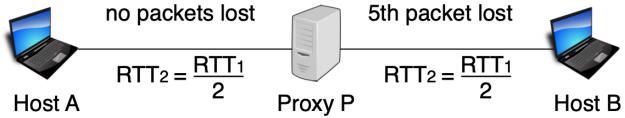
# Ex. 10.2 c) – Explain the solution




In this topology, sending 10 data segments takes 6.5RTT1

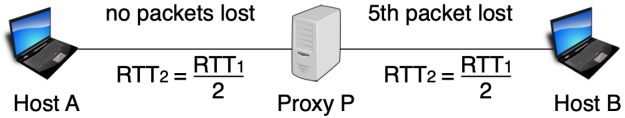


# Ex. 10.2 c) – Explain the solution



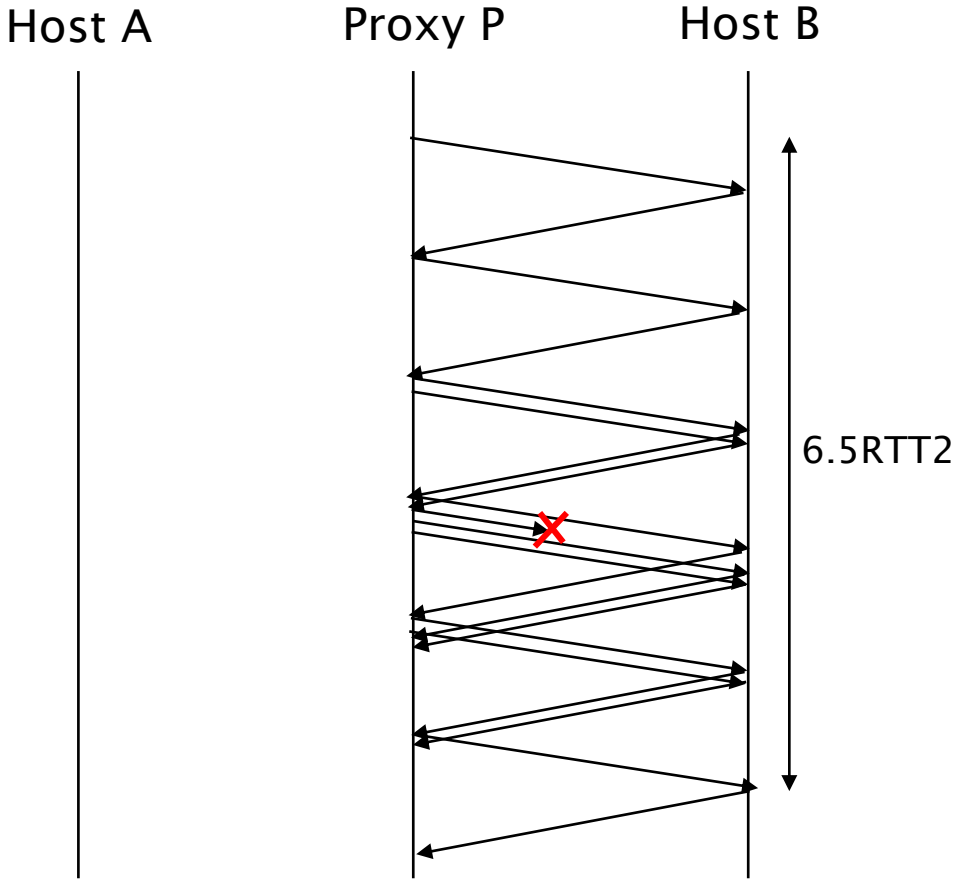

Then, in the topology with the proxy,
sending 10 data segments
takes ?? RTT2

# Ex. 10.2 c) – Explain the solution

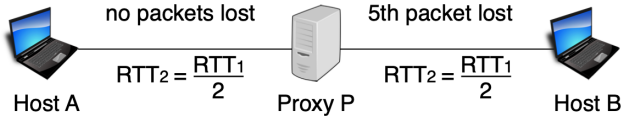



Then, in the topology with the proxy,
sending 10 data segments
takes 0.5RTT2 + 6.5RTT2

If Host A was at the location of Proxy P…



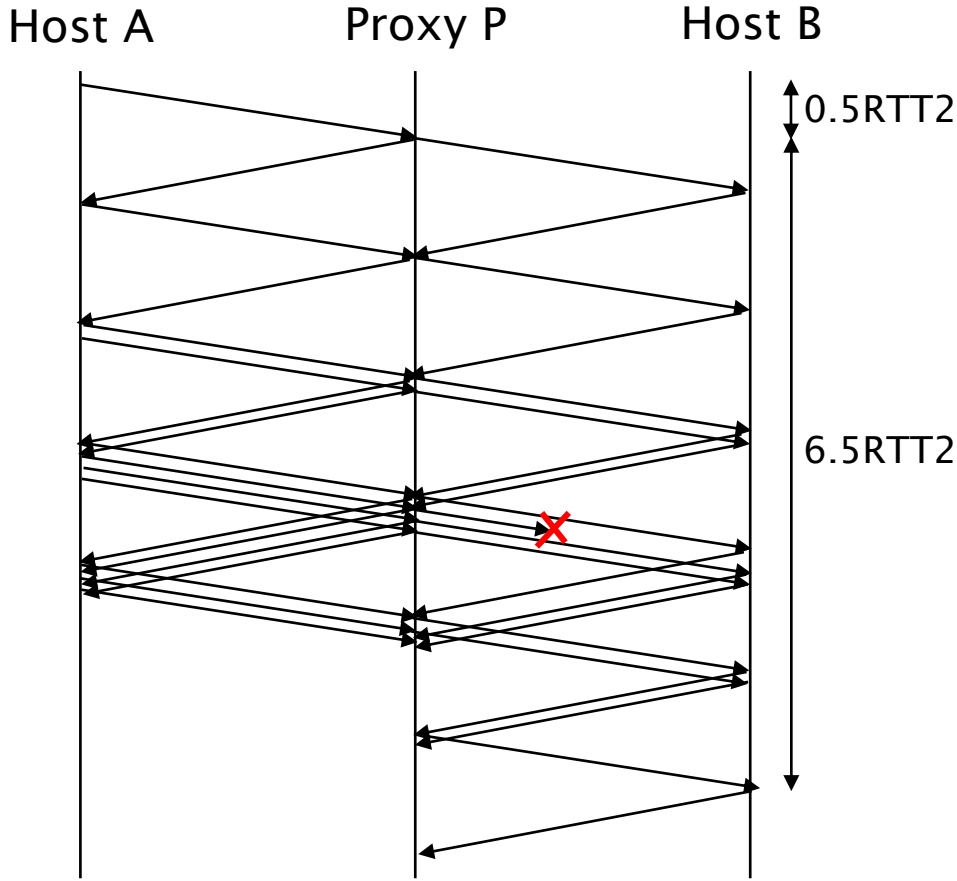
# Ex. 10.2 c) – Explain the solution




Then, in the topology with the proxy,
sending 10 data segments
takes 0.5RTT2 + 6.5RTT2

After 0.5RTT2,

- whenever Proxy P must send data to B, it has already received them from A
- Host A – Proxy P communication overlaps with Proxy P – Host B communication



# Questions on

- Ethernet/IP
- Routing
- Transport
- **Applications**
- Other

# Ex. 10.4 b) – When creating a DNS sub-domain, why add records at name servers of its *parent* domain?

- Name resolution (e.g., of fun.nsg.ee.ethz.ch) happens top-to-bottom, where the name server of a parent domain (e.g., ee.ethz.ch) points to the name server of its immediate sub-domain (nsg.ee.ethz.ch), until the name is resolved
- So, we must let the name servers responsible for each parent domain know which name servers are responsible for its immediate children and their IPs

# Ex. 10.4 e) – Explain technique to scale DNS resolution when name servers of sub-domain are overloaded

DNS entries before

nsg.ee.ethz.ch NS ns1.nsg.ee.ethz.ch
ns1.nsg.ee.ethz.ch A 129.132.20.1

DNS entries w/ technique 1

nsg.ee.ethz.ch NS ns1.nsg.ee.ethz.ch
ns1.nsg.ee.ethz.ch A 129.132.20.1

ns1.nsg.ee.ethz.ch A 129.132.20.2

DNS entries w/ technique 2

nsg.ee.ethz.ch NS ns1.nsg.ee.ethz.ch
ns1.nsg.ee.ethz.ch A 129.132.20.1

nsg.ee.ethz.ch NS ns2.nsg.ee.ethz.ch
ns2.nsg.ee.ethz.ch A 129.132.20.3

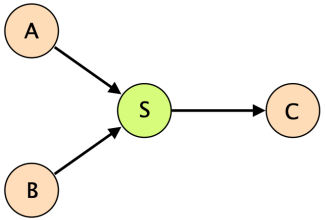
# Ex. 99.18 & 99.19 (email-related questions) – Relevant for the exam?

- No, except for 99.18 e)

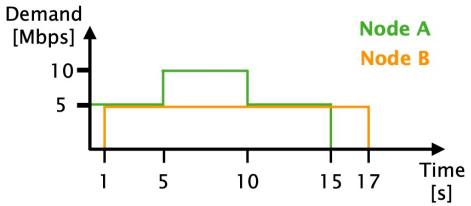
# Questions on

- Ethernet/IP
- Routing
- Transport
- Applications
- **Other**

# Ex. 1.7 f) – Time to send data with bw reservations




Network with a shared link.




Demand distributions for node A and B.

How long does it take to send data if A and B use circuit switching (reserving for the peak demand)?

Question assumes that

- we **reserve once** at t=0
- for peak demand during **entire time interval**
- we know peak demands at t=0