CSC0056: Data Communication

Leaky Bucket Flow Control, Time Synchronization, and Course Review

Instructor: Chao Wang 王超

Department of Computer Science and Information Engineering

NATIONAL TAIWAN NORMAL UNIVERSITY

Outline of lecture17

- Leaky bucket flow control (illustrated using the blackboard)
- Time synchronization among communication hosts
  - NTP and PTP
- Course review

References

- Mills, et al. Network Time Protocol Version 4: Protocol and Algorithms Specification. RFC 5905. (https://tools.ietf.org/html/rfc5905)
- Mills, D.L., "Computer Network Time Synchronization - the Network Time Protocol", CRC Press, 304 pp, 2006.
- IEEE. 2008. IEEE standard for a precision clock synchronization protocol for networked measurement and control systems. IEEE Std 1588-2008 (Revision of IEEE Std 1588-2002) (July 2008), 1–300.
- **S.Y. Wang**, H.W. Hu, and Y.B. Lin, "Design and Implementation of TCP-Friendly Meters in P4 Switches," IEEE/ACM Transactions on Networking, Volume: 28, Issue: 4, August 2020. (using ideas inspired by the leaky bucket flow control)

The need for time synchronization

- Some embedded devices simply do not have battery powered hardware clock

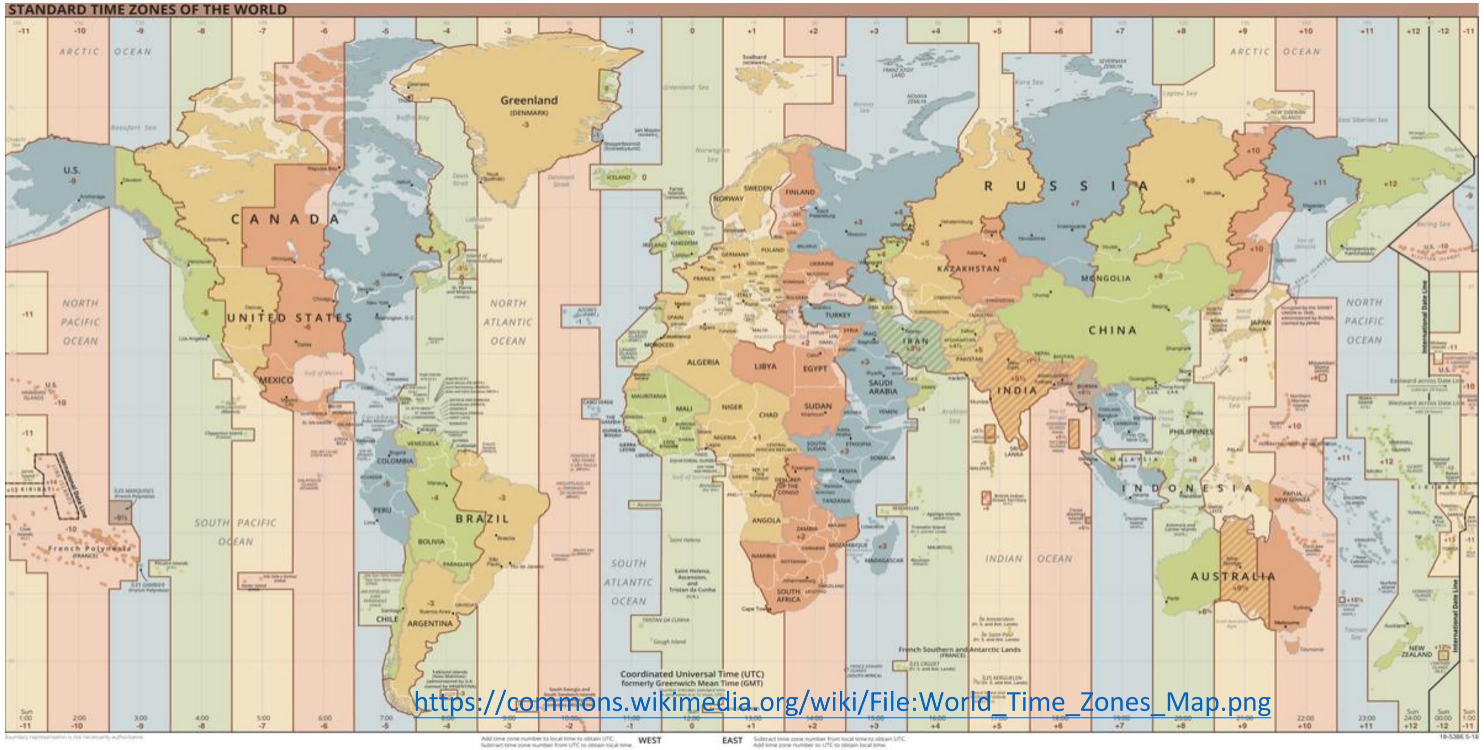
- In many applications, to measure end-to-end latency performance, both ends must have synchronized clocks

NTP: network time protocol

- Goal: minimize both the time difference and frequency difference between UTC and the system clock.”

UTC stands for:

- Coordinated Universal Time
- Temps Universel Coordonné



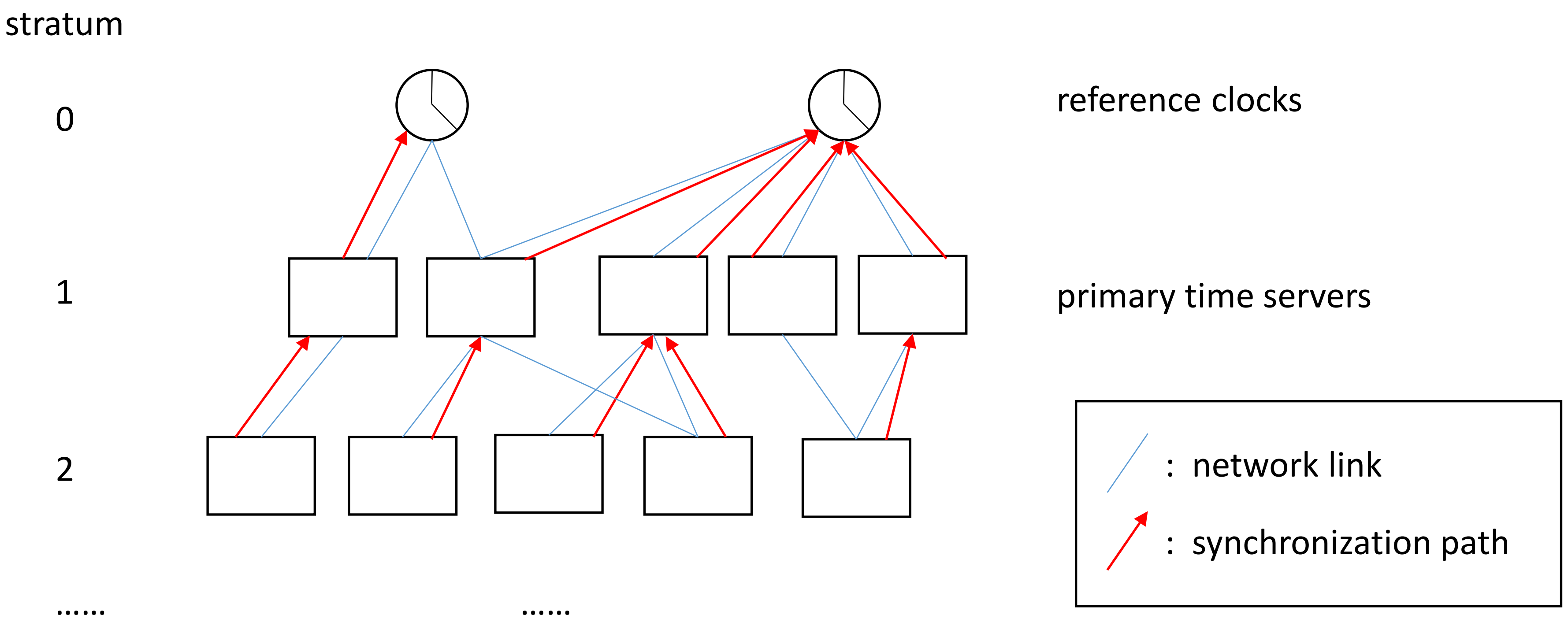
https://commons.wikimedia.org/wiki/File:World_Time_Zones_Map.png

NTP accuracy

- Conventionally, can achieve milliseconds accuracy
  - With improvement, may achieve an accuracy up to tens of microseconds
- Primary time servers
  - Synced to national standards by wire or radio
  - Accuracy: tens of microseconds
- Secondary time servers
  - Synced to primary time servers
  - Accuracy: a few hundred microseconds to a few tens of milliseconds
- Clock strata

NTP topology example

- Synchronizing clocks along the synchronization paths



NTP topology (cont.)

- Master-slave subnetwork with synchronization paths determined by some spanning-tree algorithm
- “As a standard practice, timing network topology should be organized to avoid timing loops and minimize the synchronization distance. In NTP, the subnet topology is determined using a variant of the Bellman-Ford distributed routing algorithm, which computes the shortest-path spanning tree rooted on the primary servers. As a result of this design, the algorithm automatically reorganizes the subnet, so as to produce the most accurate and reliable time, even when there are failures in the timing network.” - RFC 5905

NTP synchronization basics

- Suppose that host A is going to synchronize its clock to that of host B
- Notation:
  - δ : one-way delay
  - θ : offset between two clocks
- The mean offset between two clocks is determined by message exchange between A and B
  - (see the blackboard illustration)
- Synchronization is performed by gradually reducing the mean offset

NTP: network time protocol

- “Reliable message delivery such as TCP can actually make the delivered NTP packet less reliable since retries would increase the delay value and other errors.” - RFC 5905
- If the network is very busy, or the server’s CPU is very busy, would that affect the performance of NTP?
  - Ans: Potentially yes, because synchronization is based on message-exchange, and software timestamping may be delayed due to preemption
  - To address those problems:
    - Network: out-of-band synchronization
    - CPU: pinned tasks to certain core(s)

PTP: precision time protocol

- Designed to achieve *microsecond* to *sub-microsecond* accuracy and precision

- Spatially localized

- Administration free

- Accessible for both high-end devices and low-cost, low-end devices

PTP topology

- Synchronization is performed by syncing each slave clock to its corresponding master clock (master-slave relationship is defined *per link*)

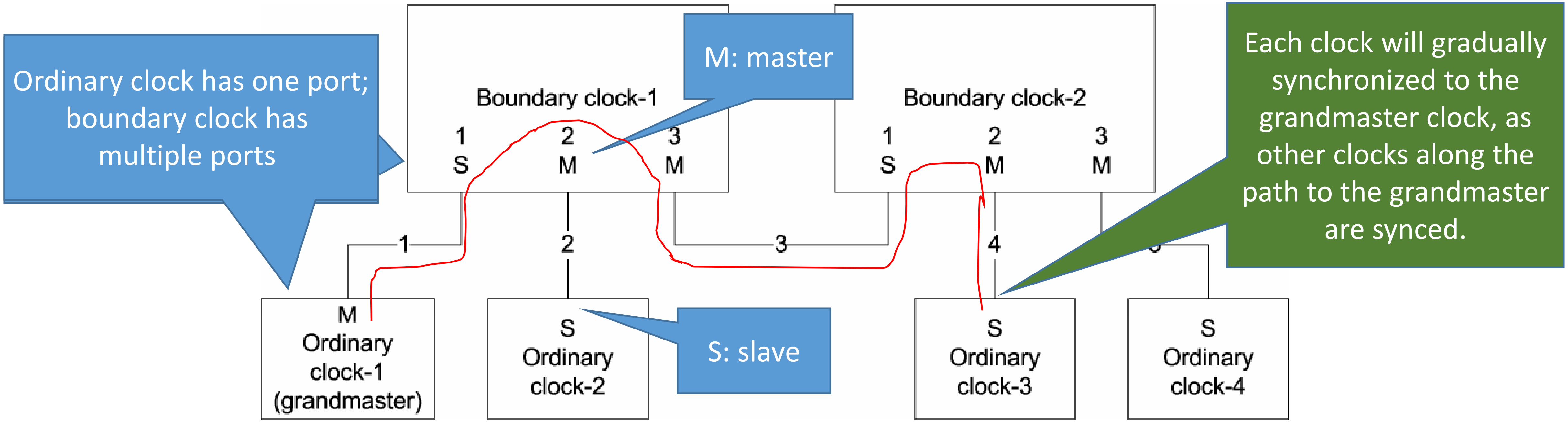
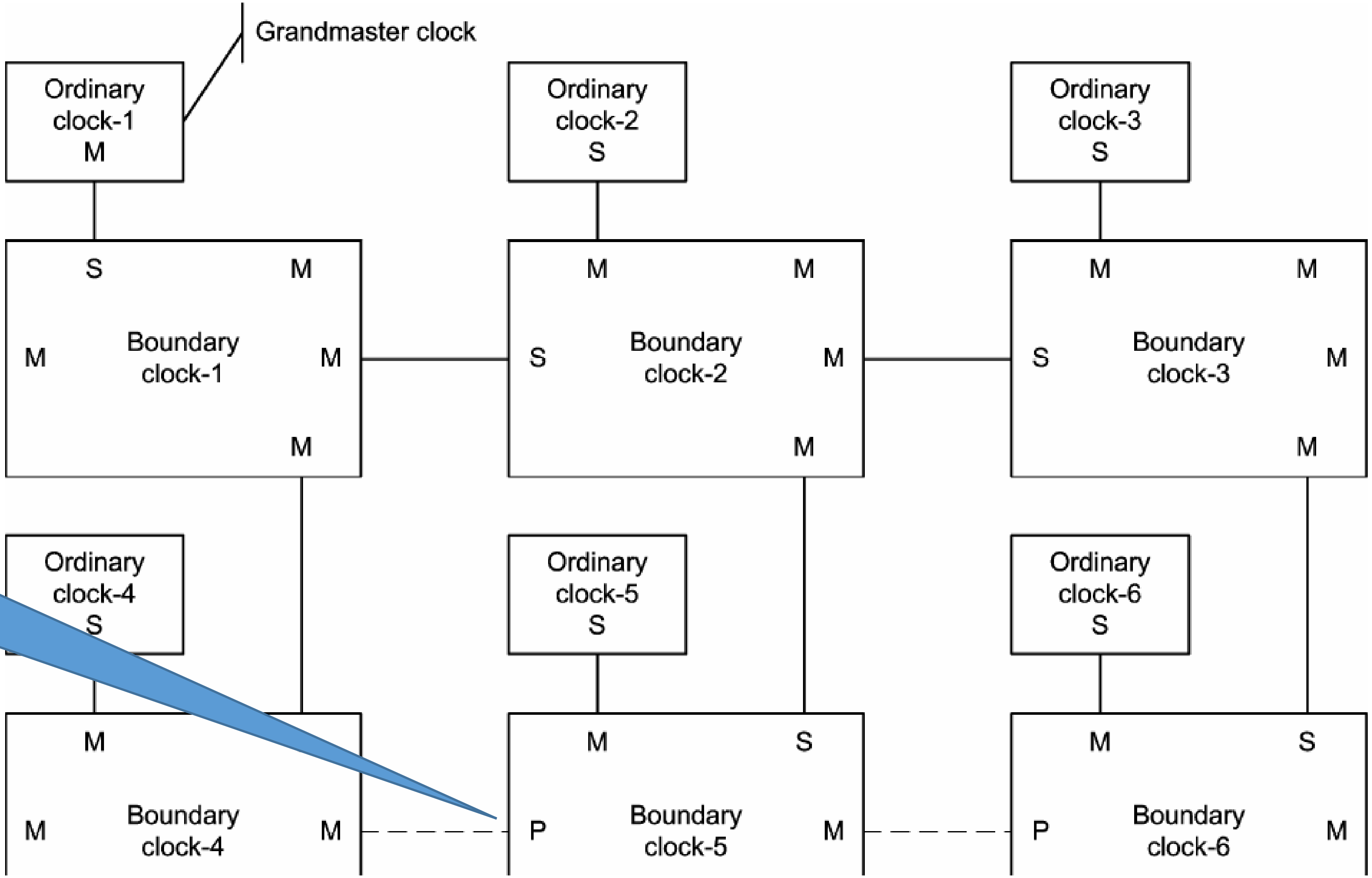


Figure 10—Simple master–slave clock hierarchy

PTP topology (cont.)

Some ports are passive (denoted by S), to prevent loops in the PTP topology



Part of **Figure 11—Pruned mesh topology**

Basic synchronization method

- termed “delay request-response”
- *Mean* path delay

  $= [(t_2 - t_1) + (t_4 - t_3)]/2$
- *Mean* offset from master

  $= [(t_2 - t_1) + (t_3 - t_4)]/2$

used for synchronization

<u>Offset from Master time</u>
= time on the slave clock - time on the master clock
= t_2 - t_1 - mean path delay (- <u>correction)</u>

Optional, by the use of transparent clock
(next slides)

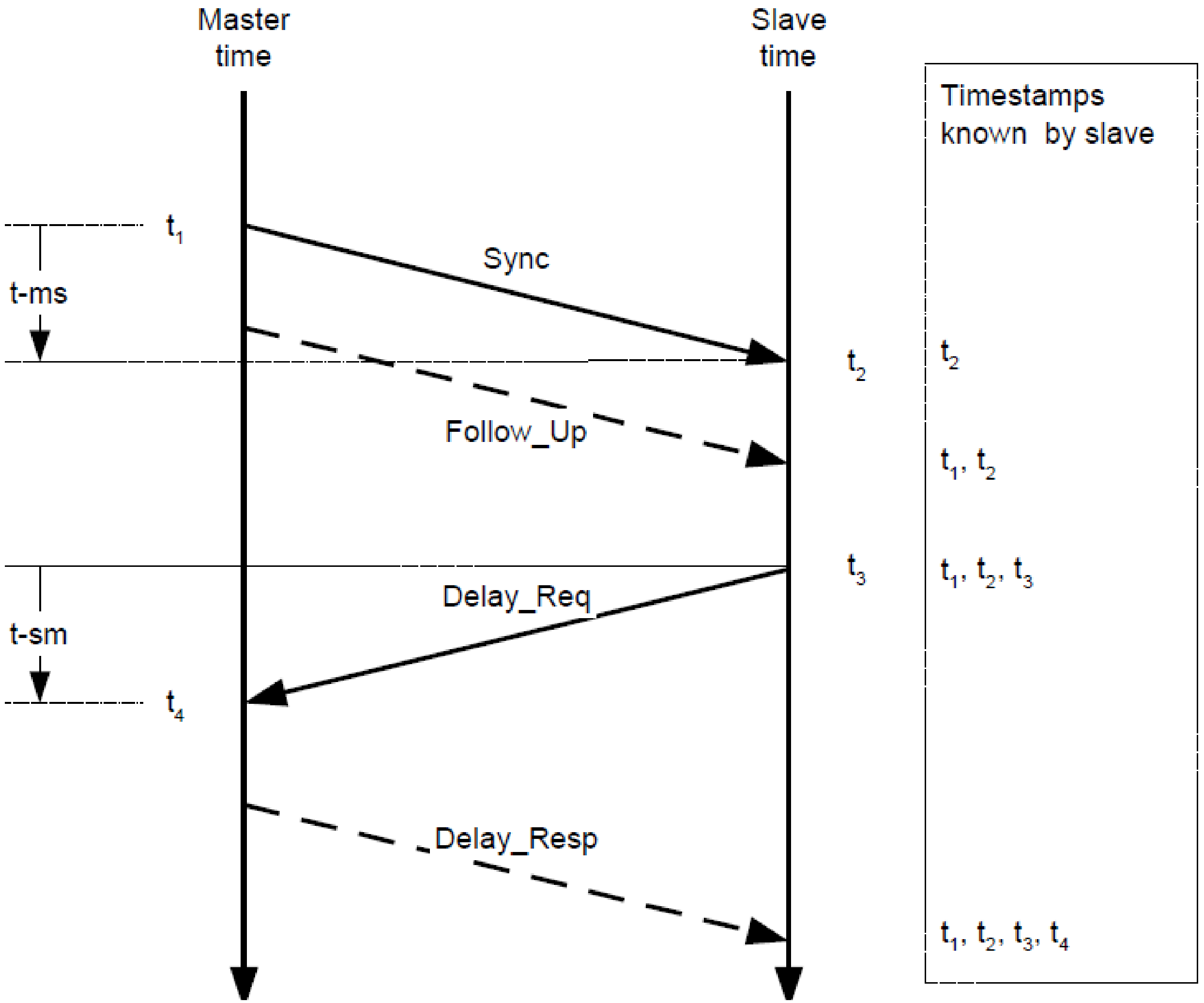
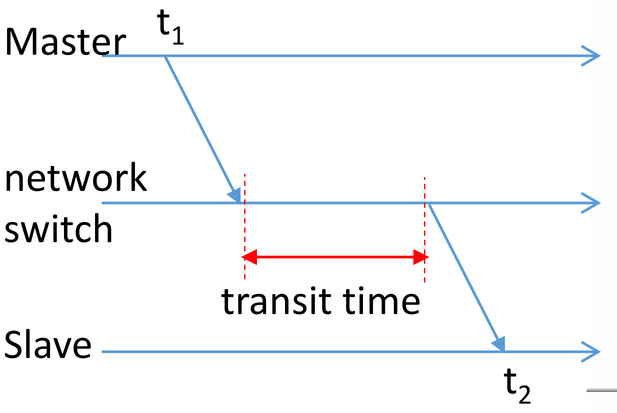


Figure 12—Basic synchronization message exchange

End-to-end transparent clock

for intermediate device, e.g., a network switch

- A transparent clock records the transit time (time interval from ingress to egress), which may be subtracted later to correct the path delay



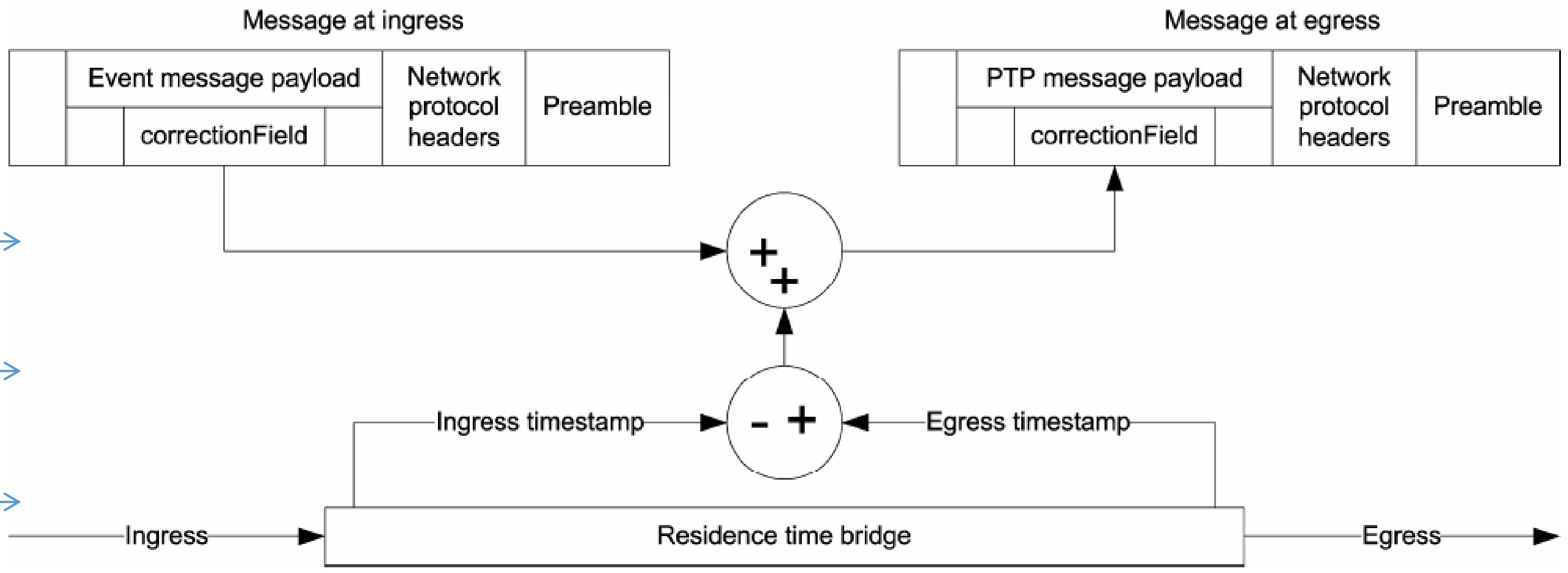
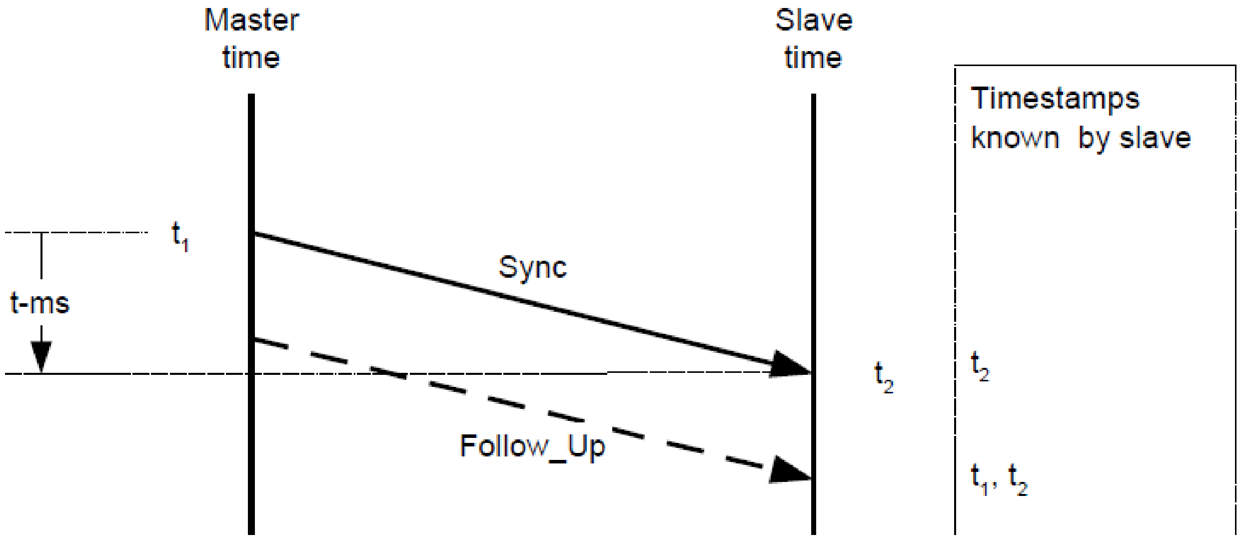


Figure 5—End-to-end residence time correction model

Alternative to delay request-response: Peer delay link measurement

- Idea:
  - Following slides 16, if we can determine the mean path delay, then there is no need for Delay_Req/Delay_Resp message exchange.
  - This may both speed up synchronization process and save some processing load.



Offset from Master time
= time on the slave clock - time on the master clock
= t_2 - t_1 - mean path delay (- correction)

Peer delay link measurement (cont.)

- Mean path delay may be calculate peer-to-peer at each pair of ports
- Mean path delay $= [(t_2 - t_1) + (t_4 - t_3)]/2$

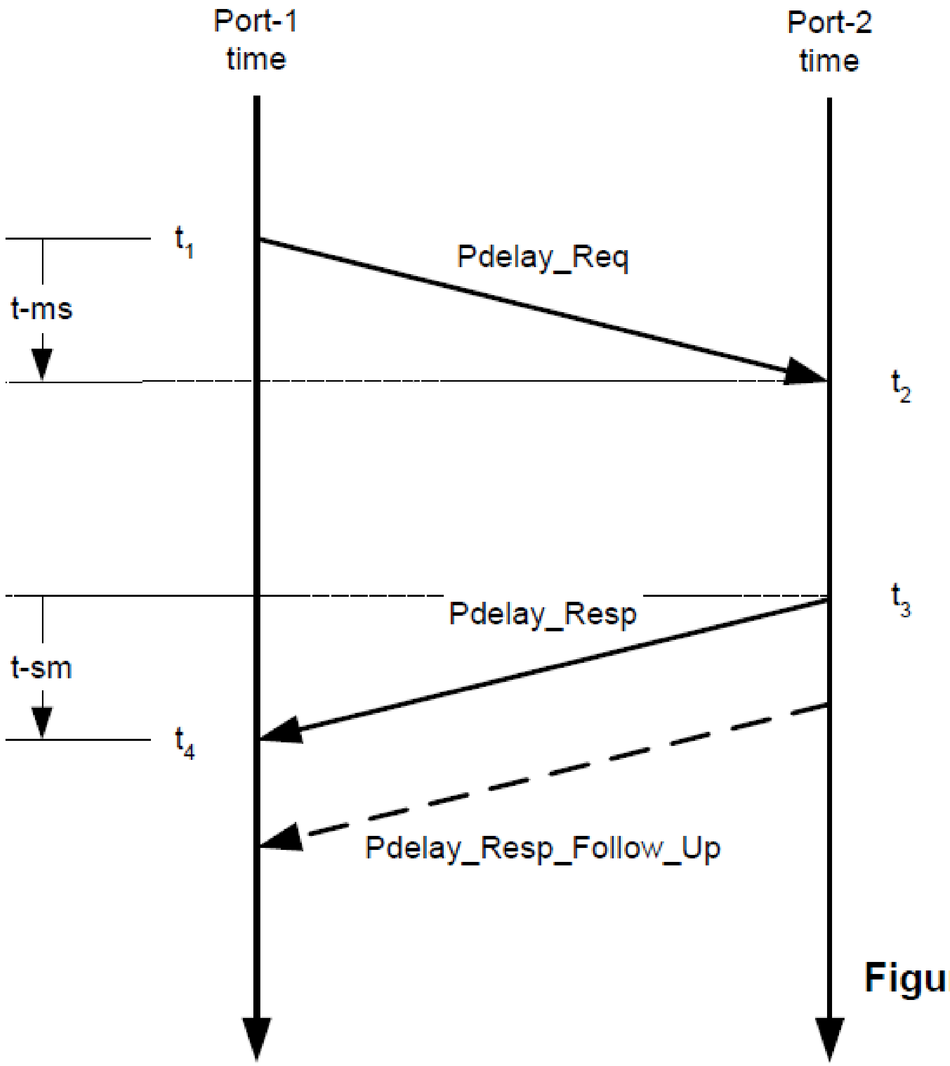
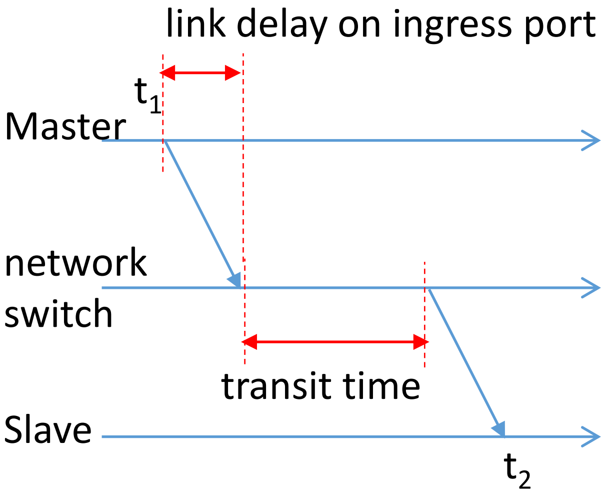


Figure 13—Link delay measurement

Peer-to-peer transparent clock

- Used along with peer delay link measurement



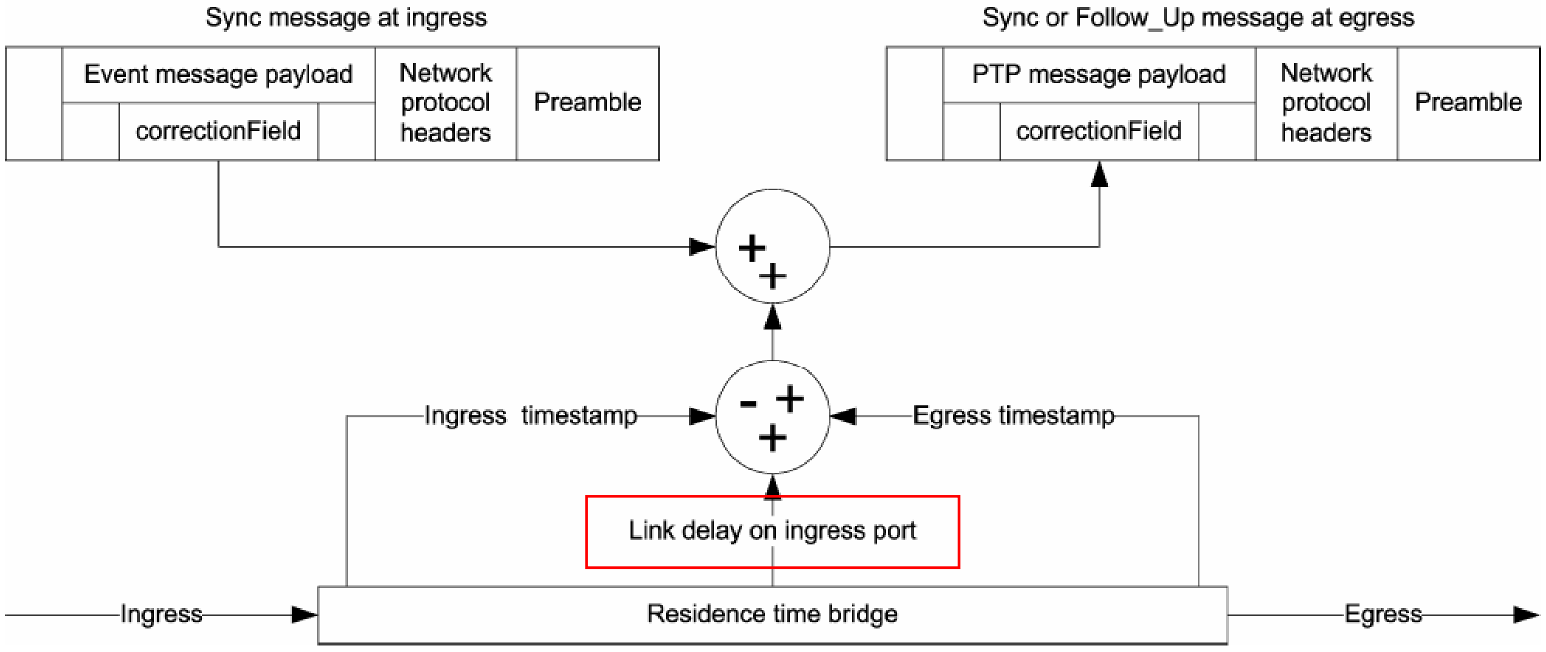


Figure 8—Peer-to-peer residence time and link delay correction model

Timestamp generation

- A timestamp may be taken at point A, B, or C

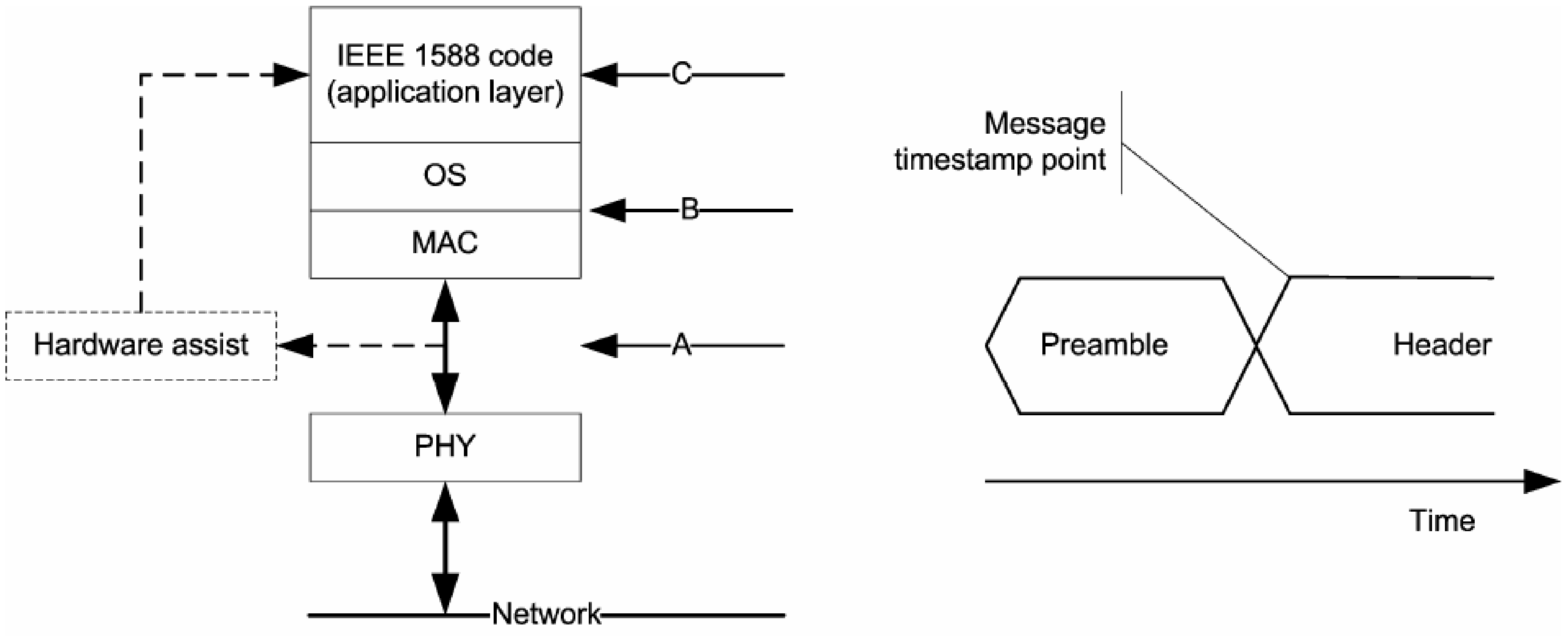


Figure 14—Timestamp generation model

Choosing NTP or PTP?

- It depends.
- With support of hardware timestamping, PTP may provide sub-microsecond accuracy
- There are many public NTP servers and built-in NTP clients for easy time synchronization
  - Amazon EC2 offers one for its VMs
- Example scenario (combine both PTP and NTP):
  - Use PTP to synchronize local devices, and use NTP to synchronize both local master clock and remote devices to a global NTP server

Outline of lecture17

- Leaky bucket flow control (illustrated using the blackboard)
- Time synchronization among communication hosts
  - NTP and PTP
- **Course review**

Course review

- Data communication as a network of flows
- Point-to-point communication vs. end-to-end communication
  - ARQ at the data-link layer vs. Window flow control at the transport layer
- Error detection vs. error correction vs. retransmission
- Data retransmission and/or Passive replication
- Free-for-all multiaccess vs. Perfectly-scheduled multiaccess
  - CAN, Aloha, TDMA and its extension to improve throughput

Course review (cont.)

- Theory and models
  - Data communication as a network of flows (a graph)
  - Network flow algorithm, shortest-path algorithm, etc.
  - Little’s theorem, Queueing theory, Markov chains, Poisson process
  - Scheduling for timely data delivery and/or data-loss tolerance
- Protocols and systems
  - CRC, ARQ, multiplexing, CAN, Aloha, TDMA and its extension
  - Real-time communication
  - Fault-tolerant communication
  - flow controls, time synchronization