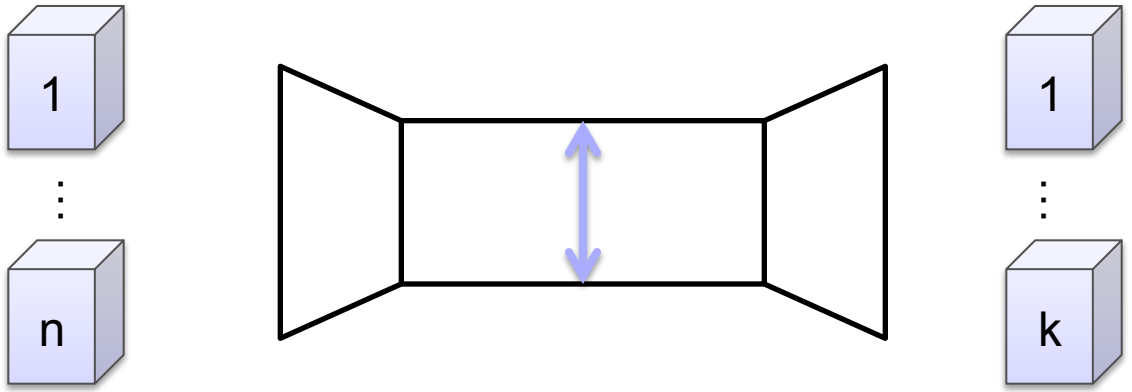
Information Network I
TCP 2/2

Youki Kadobayashi
NAIST

Transport protocol challenges

- With a number of unknown parameters:
  - Number of active communications
  - Bottleneck bandwidth
  - Error rate
- how can we accommodate as much communications as possible, without collapsing network itself?



Key ideas: probing, estimation, self-policing, macro-level stability

Flow control and congestion control:
Competition and Arbitration of Network Resource

- Flow control
  - Absorb difference of transmission rate
  - Recovery from sequence error
  - Recovery from duplication, packets drop and bit error
- Congestion control
  - Sharing the bandwidth while suppressing the congestion
  - Fair sharing of bandwidth

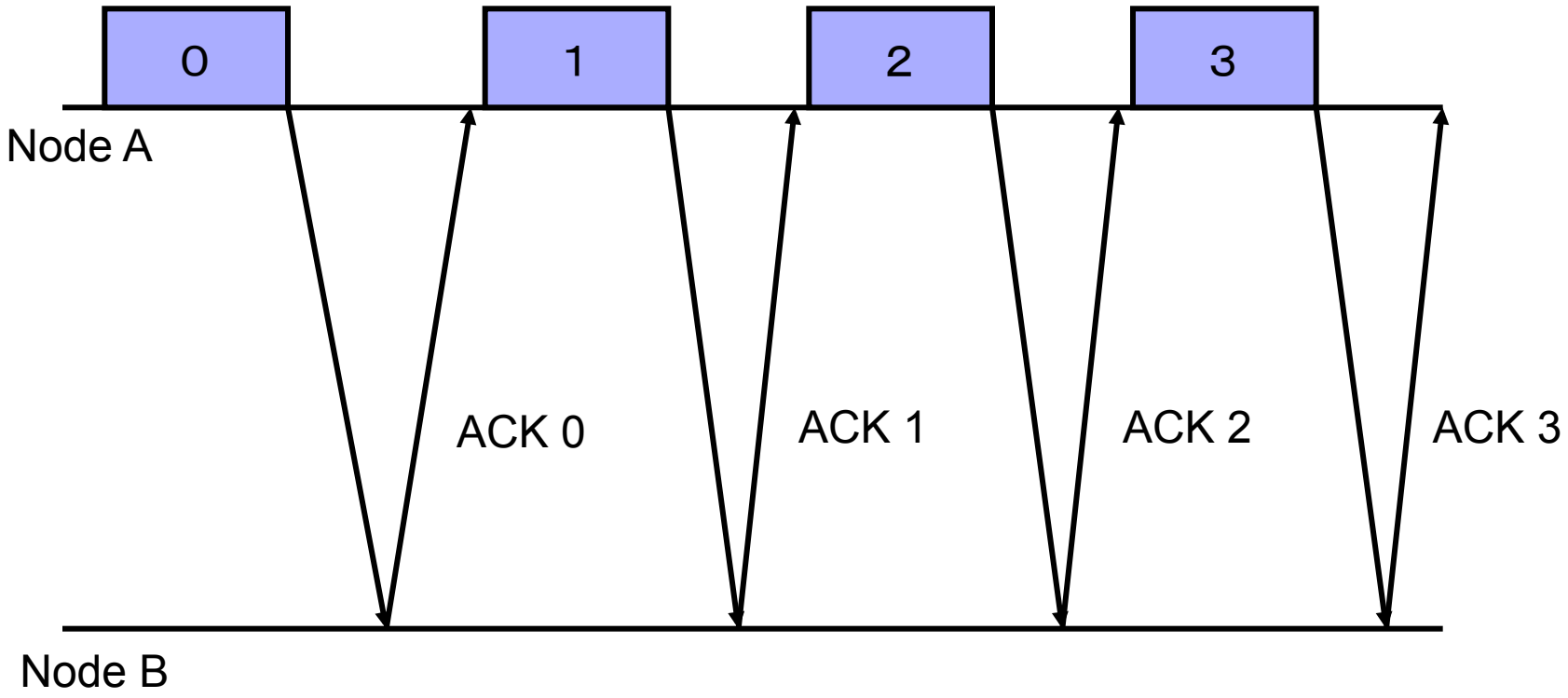
Flow Control

- Stop-and-Wait
- Go Back n

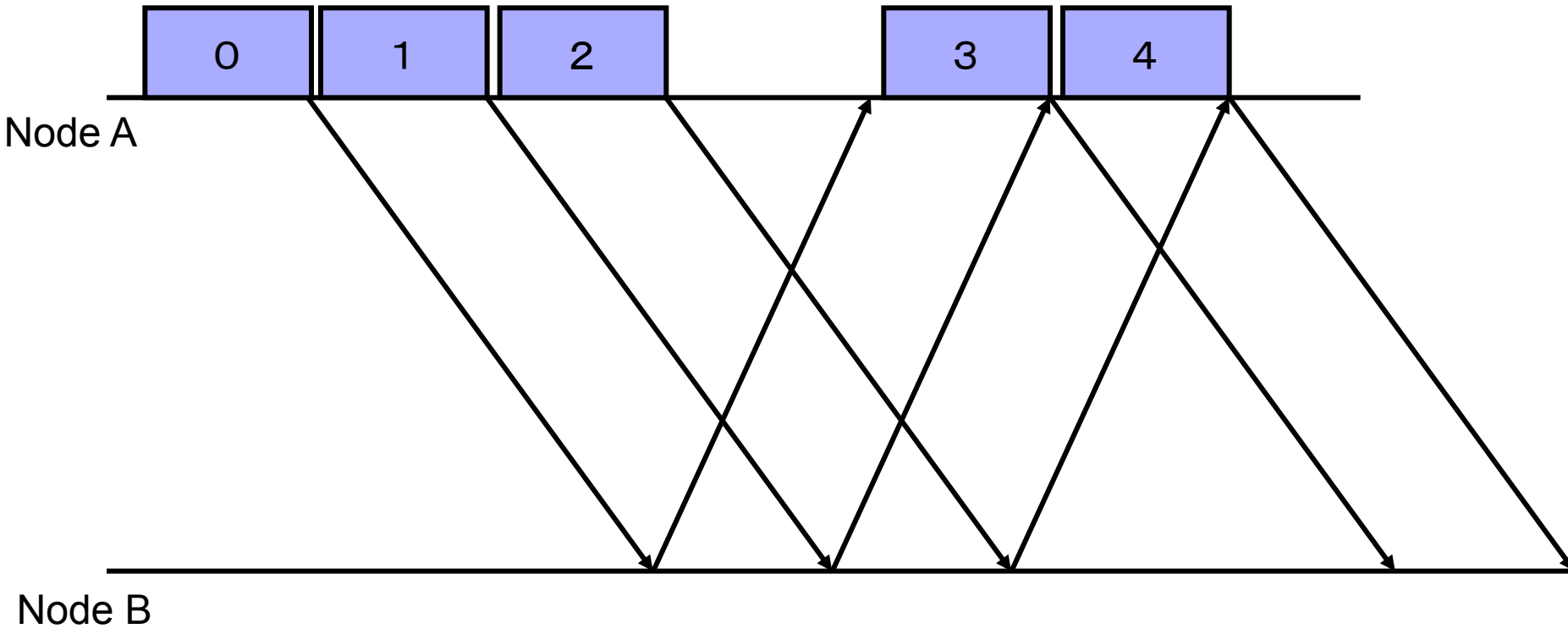
- Various methods exist
  - Bandwidth advertisement from the network
  - Bandwidth estimation at individual hosts

  - ARQ (Adaptive Repeat reQuest)

Stop-and-Wait



Go Back n



Characteristics of TCP Flow Control

- End to end
  - No global assignment of resource
  - Estimate available bandwidth at individual hosts
  - Routers do not explicitly allocate resource
    - Implicit signaling through packet drops
- Scalable
  - Host-based autonomous method has better scalability, because it doesn't need state management inside network.
    - Autonomous → Less state management → Scalable

Questions?

TCP: key characteristics

- Very simple algorithm
  - Macroscopic self-stabilization
- TCP doesn't assume the presence of greedy nodes.
  - Elimination of global control system
  - Reject the idea of intermediate policing system
- Modest performance across almost all data-links
  - As opposed to optimal performance in specific condition
- Stepwise improvements over 30 years

Flow Control on TCP

- Control available bandwidth
  - Sliding window
- Using sequence number, not packet number
  - Window size
- Control transmission interval of packets
  - ACK clocking

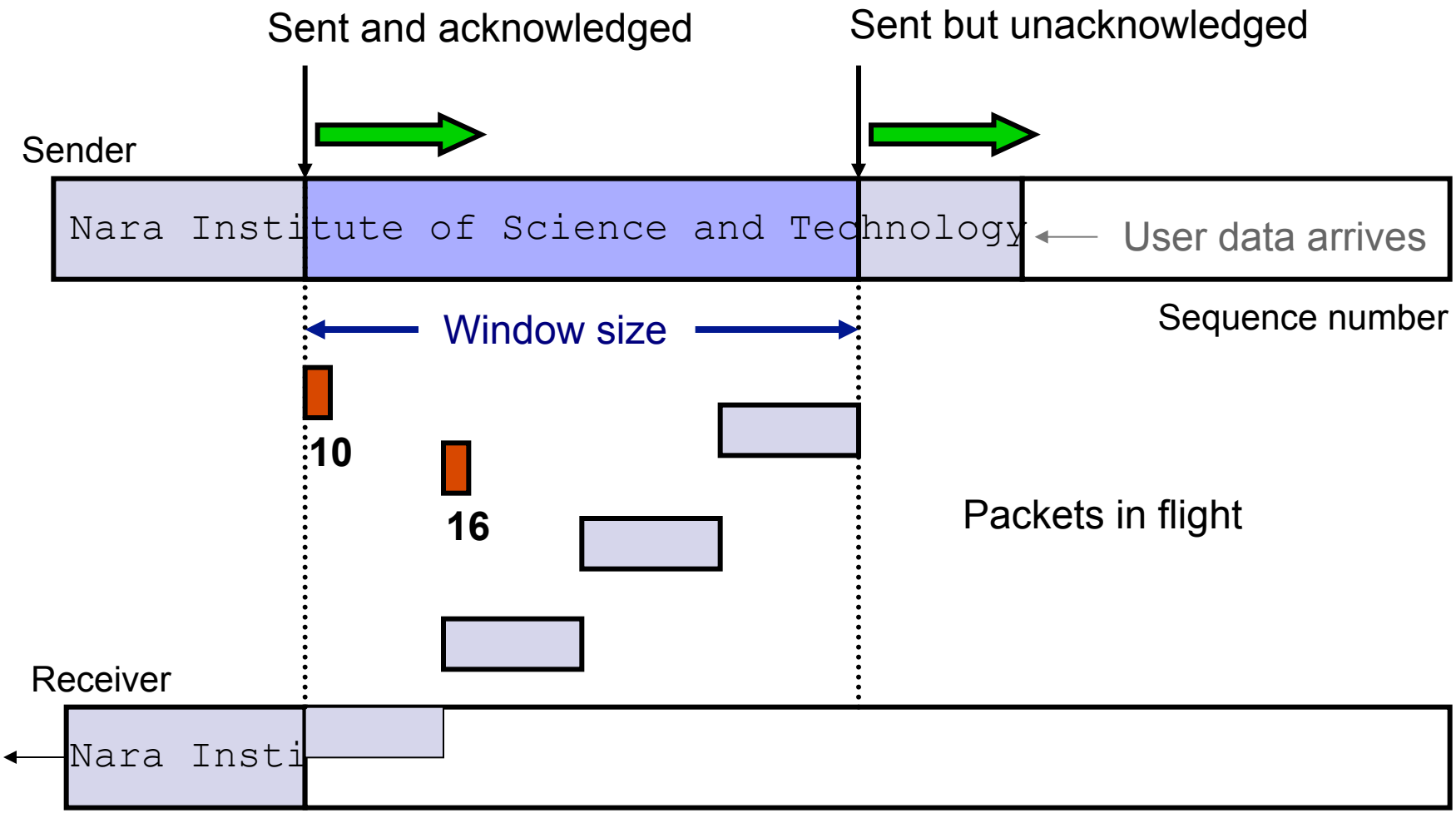
- Miscellaneous
  - Error detection with TCP checksum
  - Packet drop detection with duplicate ACK, timeout

ACK clocking

- Demonstration by nam

- ACK clocking
- Duplicate ACK

Sliding window



Questions?

Congestion Control on TCP

- Fair-share model: fair distribution of bandwidth
  End to end
- Increase and decrease of window size
  - additive increase
  - multiplicative decrease
    - AIMD is known to be self-stabilizing (R. Jain, et al., “Analysis of the increase and decrease algorithms for congestion avoidance in computer networks”, 1989)
- Switch increasing strategy of window size
- Detect congestion through packet drops

Fair-share

- Demo by nam

Increasing Window size

- TCP switches increasing strategy of congestion window (cwnd) by slow start threshold (ssthresh)

(Outline of the algorithm)

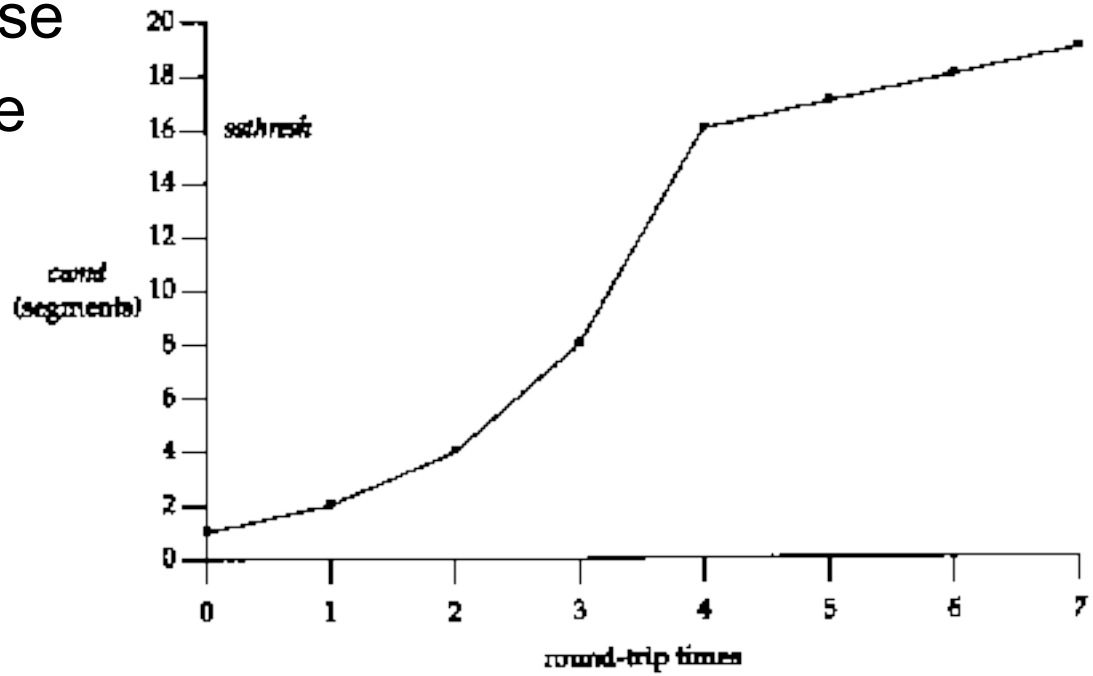
- On receiving an ACK:

```
if (cwnd < ssthresh) {
        /* slow start: exponential increase */
        cwnd += 1;
} else {
        /* congestion avoidance: additive increase */
        cwnd += 1 / cwnd;
}
```

Increasing Window size

- Slow start
  - exponential increase
- Congestion avoidance
  - additive increase

Source: TCP/IP Illustrated, Vol.1



- Effectiveness:

see V. Jacobson, “Congestion Avoidance and Control”, SIGCOMM’88.

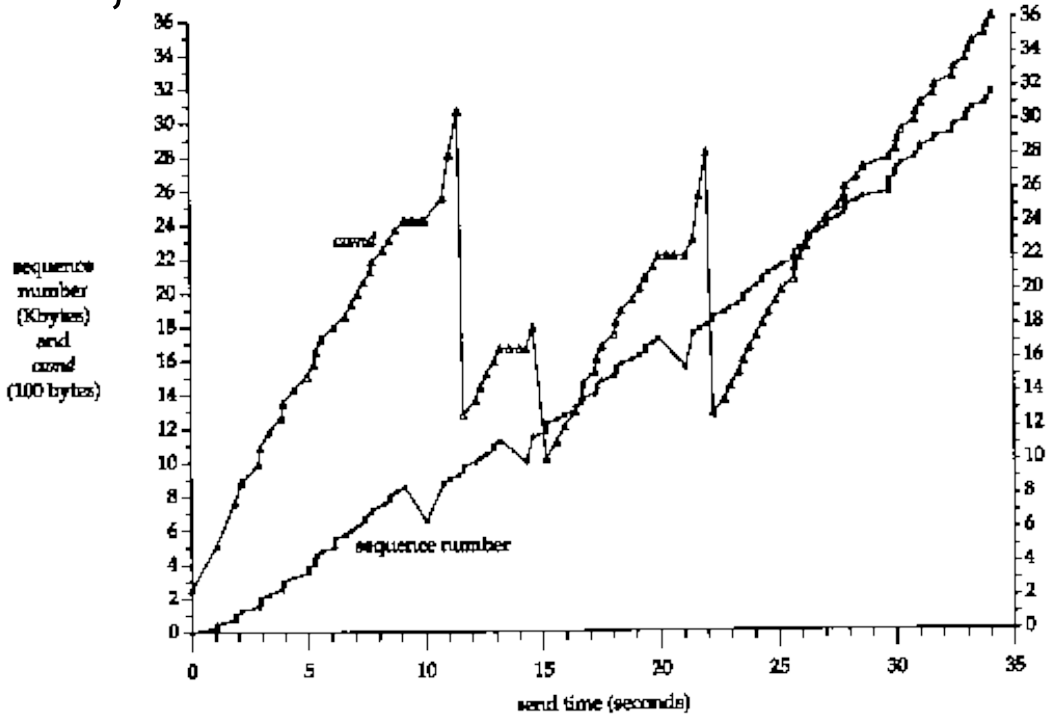
Decreasing Window size

(Outline of algorithm)

- On detecting packet drop:

```
ssthresh = cwnd / 2;
if (timeout) {
        cwnd = 1;
}
```

Source: TCP/IP Illustrated, Vol.1



Questions?

Packet drop

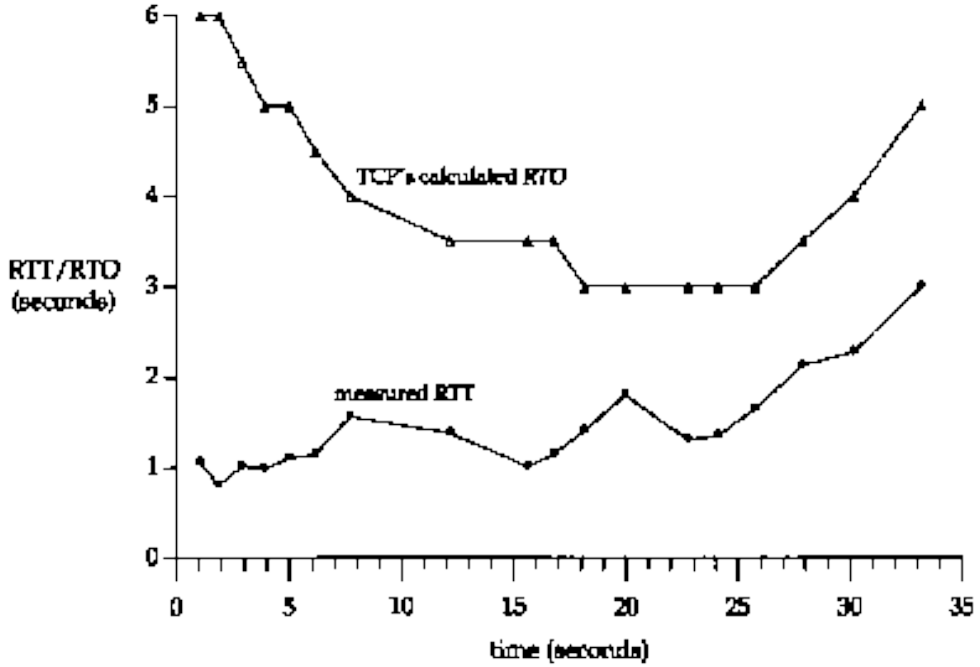
- Detection with 2 methods:
  - Duplicate ACK
  - Retransmission Time Out (RTO)

- How do I judge the time out?

  → RTO ~ RTT Estimator

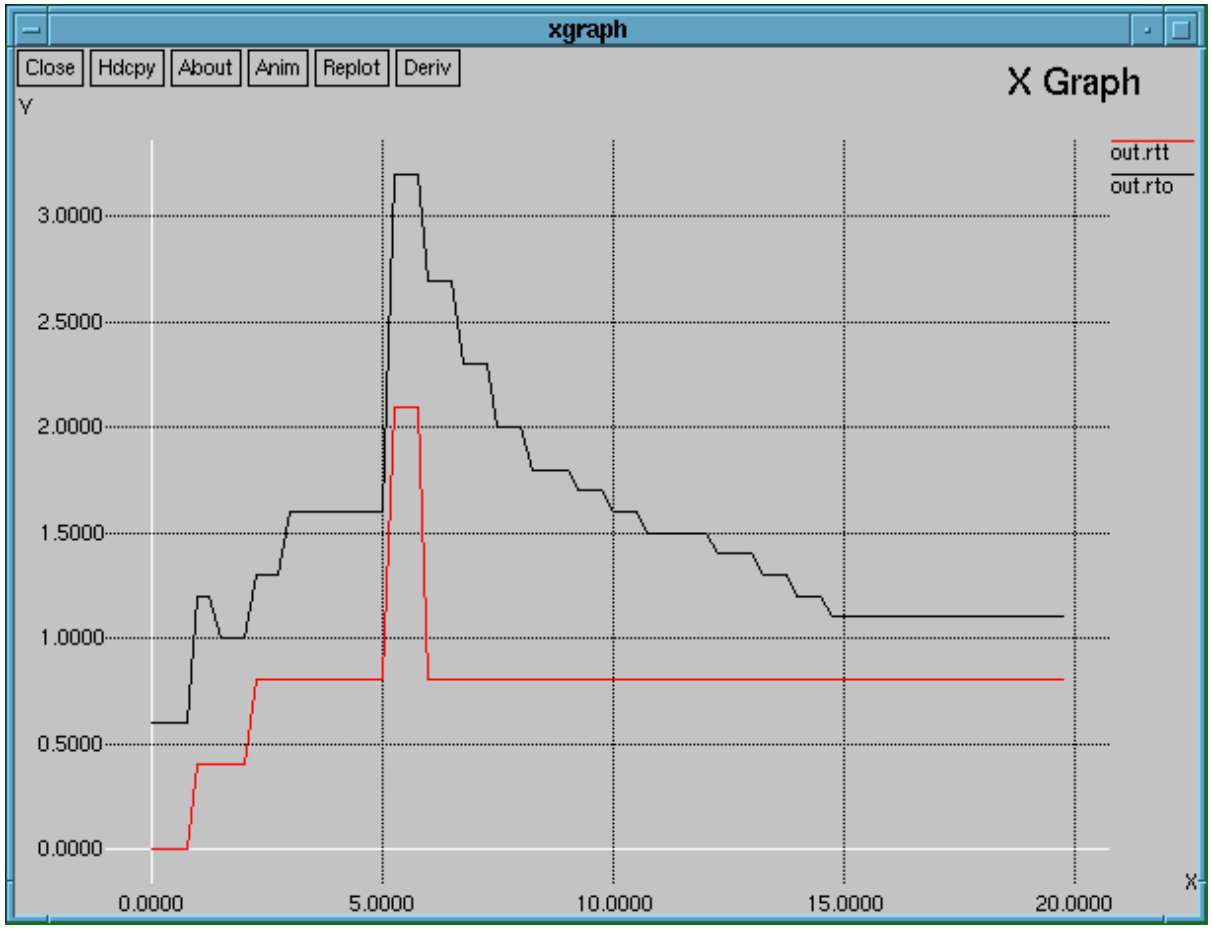
RTO Calculation

Source: TCP/IP Illustrated, Vol.1



- Err = M – A
  A <- A + gErr
  D <- D + h(|Err| - D)
  RTO = A + 4D
  - A: smoothed RTT
  - D: smoothed mean deviation
  - g: gain for the average (1/8)
  - h: gain for the deviation (1/4)

RTO calculation



Source: A Quick Tour Around TCP,
http://web.eecs.utk.edu/~dunigan/tcptour/

Loss sensitivity

- Demo by nam

- Timeout
- Congestion avoidance

More topics on TCP

- Fast retransmit
- Fast recovery (RFC3782)
- Selective Acknowledgment (SACK) (RFC3517)
- TCP Friendly Rate Control (TFRC) (RFC3448)
- Explicit Congestion Notification (ECN) (RFC3168)

- Interaction with RED
- TCP extensions for wireless links
- ...

TCP enhancements

- Improvements of algorithm:
  - Reno
  - NewReno
  - SACK
  - ...
- 2 types of TCP extensions exist:
  - Optimal performance in specific condition
    - → many academic papers, but you don't need to pay attention to these papers
  - Improved performance in various conditions
    - → standardization in IETF
- Vendor-specific tweaks may result in degradation

Questions?

Conclusion

- Flow Control
  - Stop-and-Wait
  - Go back n
- Flown Control on TCP
  - Sliding window
- Congestion Control on TCP
  - Slow start
  - Congestion avoidance

Assignment

- With the provided VM image, pick a sluggish website and analyze its root cause with tcpdump or wireshark.

- Due: 5/23 17:00 JST
- Post to: A3F Internet Engineering Lab