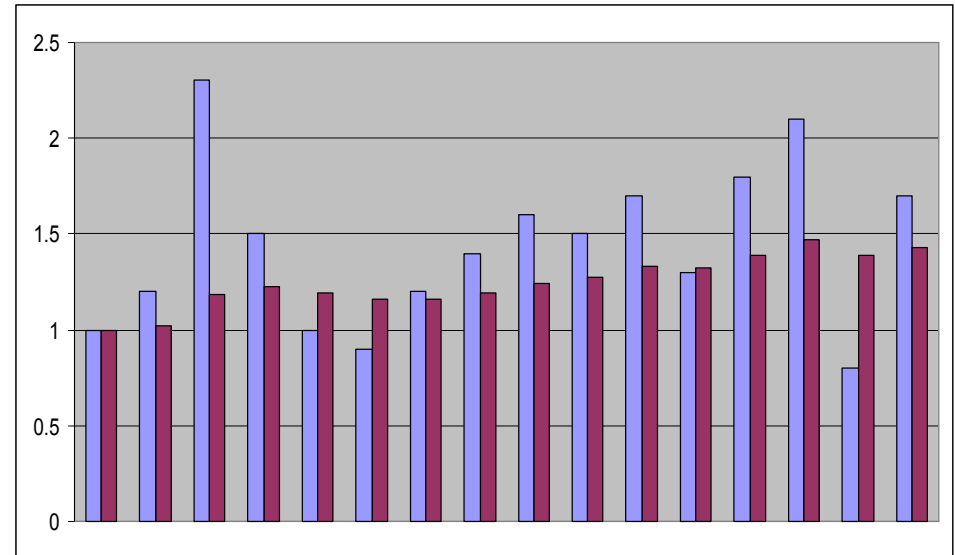


Transport layer: RTT Estimation: Addendum

ECE 50863 – Computer Network Systems

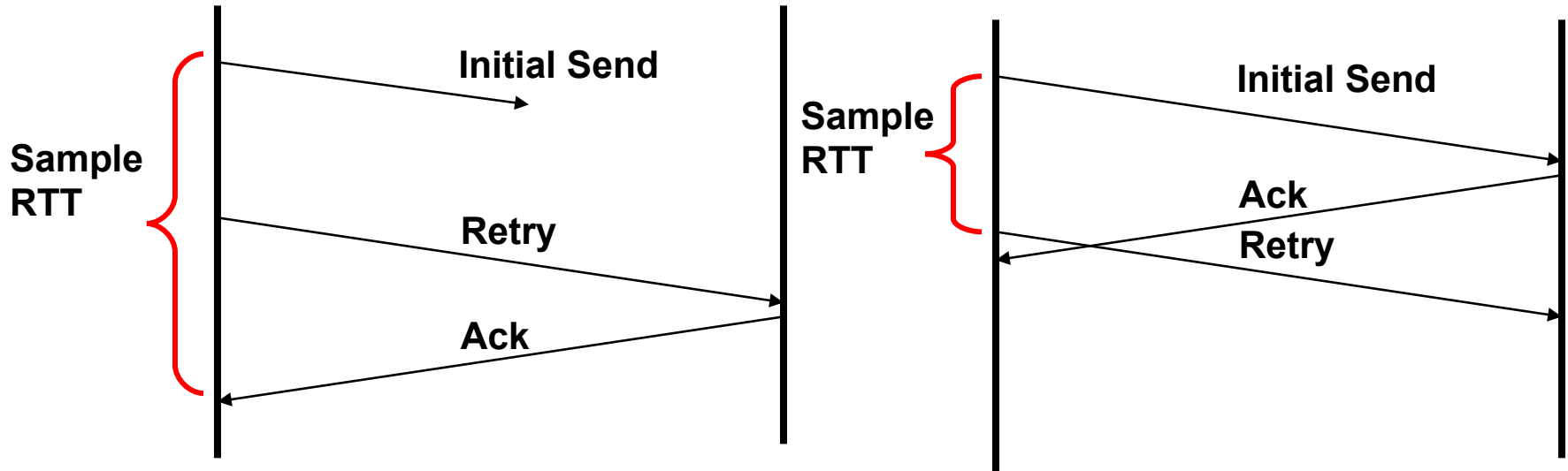
Original TCP Round-trip Estimator

- Round trip times exponentially averaged:
 - New RTT = α (old RTT) + $(1 - \alpha)$ (new sample)
 - Recommended value for α : 0.8 - 0.9
 - 0.875 for most TCP's



- Retransmit timer set to $(b * RTT)$, where $b = 2$
 - Every time timer expires, RTO exponentially backed-off
- Enhanced algorithm used in practice
 - [Jacobson/Karels: see textbook if interested]

Issue



What value of Sample RTT to use?

Solution: TCP simply neglects SampleRTT in such cases.
[Karn/Partridge algorithm]

Jacobson/Karnel's algorithm

- Original: Retransmission Timer $\Rightarrow b * RTT$, $b=2$
- $b=2$ for all paths not a good choice.
- b should depend on variability of RTT.
 - High variability: larger b , low variability: smaller b .
- Measure variability itself dynamically.

More details [Computing TCP's retransmission timer. RFC 6298 (2011)]

- Initialization:
 - RTO initialized to 1 second (prior to 2011, 3 seconds)
- SRTT: Smoothed RTT Estimate, RTTVAR: Estimate of RTT Variance, G: clock granularity

(2.2) When the first RTT measurement R is made, the host MUST set

```
SRTT <- R
RTTVAR <- R/2
RTO <- SRTT + max (G, K*RTTVAR)
```

where $K = 4$.

Updating RTT related variables

When a subsequent RTT measurement R' is made, a host MUST set

```
RTTVAR <- (1 - beta) * RTTVAR + beta * |SRTT - R'|  
SRTT <- (1 - alpha) * SRTT + alpha * R'
```

The value of SRTT used in the update to RTTVAR is its value before updating SRTT itself using the second assignment. That is, updating RTTVAR and SRTT MUST be computed in the above order.

The above SHOULD be computed using $\alpha=1/8$ and $\beta=1/4$ (as suggested in [JK88]).

After the computation, a host MUST update
 $RTO <- SRTT + \max(G, K \cdot RTTVAR)$

When retransmission timer expires

- The host MUST set $RTO \leftarrow RTO * 2$ ("back off the timer").
- Note that after retransmitting, once a new RTT measurement is obtained (which can only happen when new data has been sent and acknowledged), the computations outlined previously are performed, including the computation of RTO, which may result in "collapsing" RTO back down after it has been subject to exponential back off.

RTT Estimation References

- “Improving RTT Estimates in Reliable Transport Protocols”, Karn and Partridge, ACM TOCS, 1991
- Requirements for Internet Hosts (RFC 1122) [Sec 4.2.3] (1989)
- Computing TCP’s retransmission timer. RFC 6298 (2011)
- Congestion Avoidance and Control, Sigcomm 1988.
- Van Jacobson’s emails on TCP
 - <https://ee.lbl.gov/tcp.html>