TCP Self-clocking

Allen B. Downey

Franklin W. Olin College of Engineering

Eight Things

Eight things we know about TCP.

1. During slow start, cw grows exponentially.
2. When cw exceeds bdp, packets get dropped.
3. After a drop, the send rate decreases.
4. After slow start, cw grows linearly between drops.

Eight Things

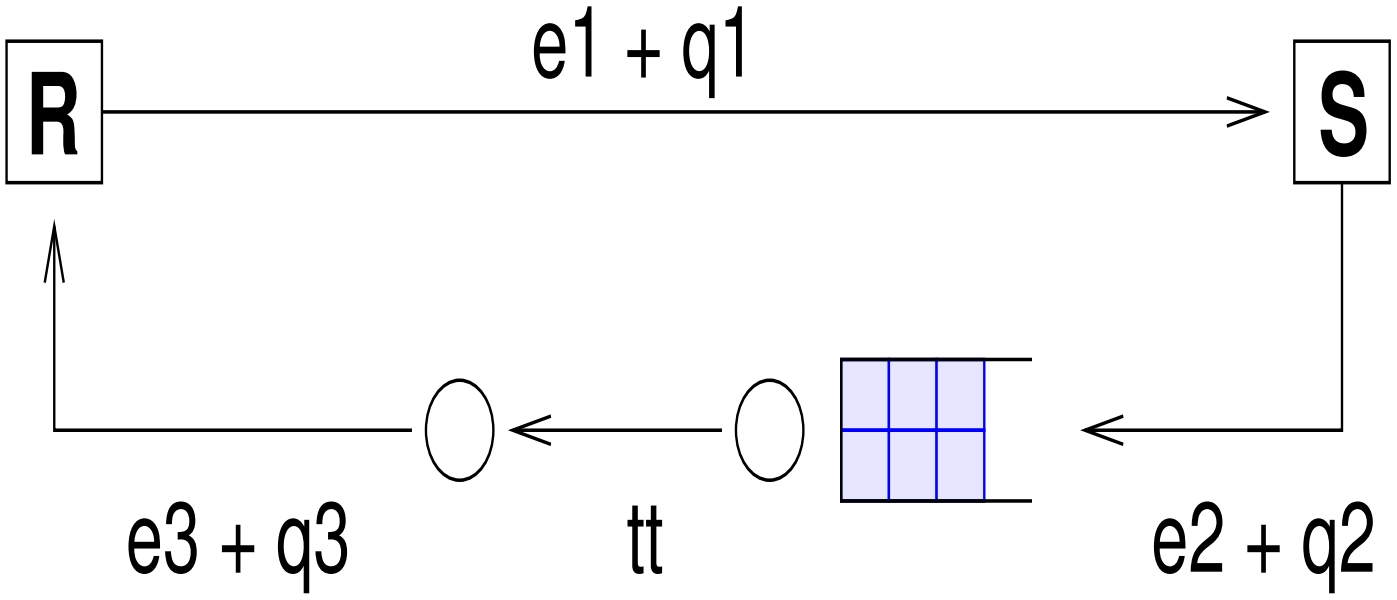
5. Steady-state behavior for long transfers is AIMD.
6. In steady state, $throughput \sim 1/\sqrt{p}$.
7. When connections share a bottleneck, $throughput \sim 1/rtt$.
8. TCP is slow to discover increased available capacity.

False!

- These claims are all false, when the buffer at the bottleneck $> bdp$.
- And it often is.
- Model first, then measurements...

Network Model

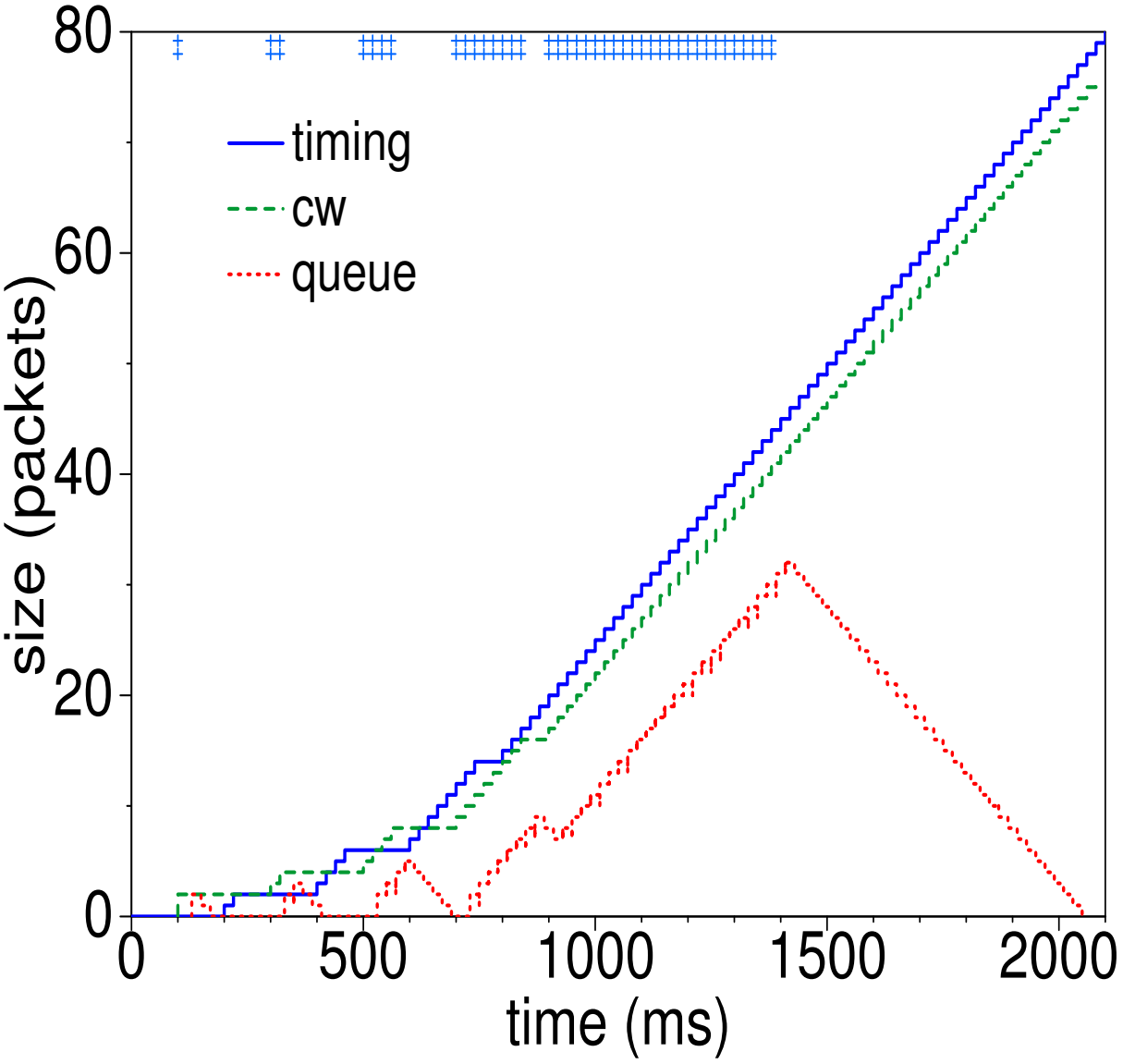
- Receiver sends request, gets reply.
- Then sends ACKs, gets packets.
- Explicit queue at bottleneck.
- Random queue delays before and after bottleneck.



Protocol Model

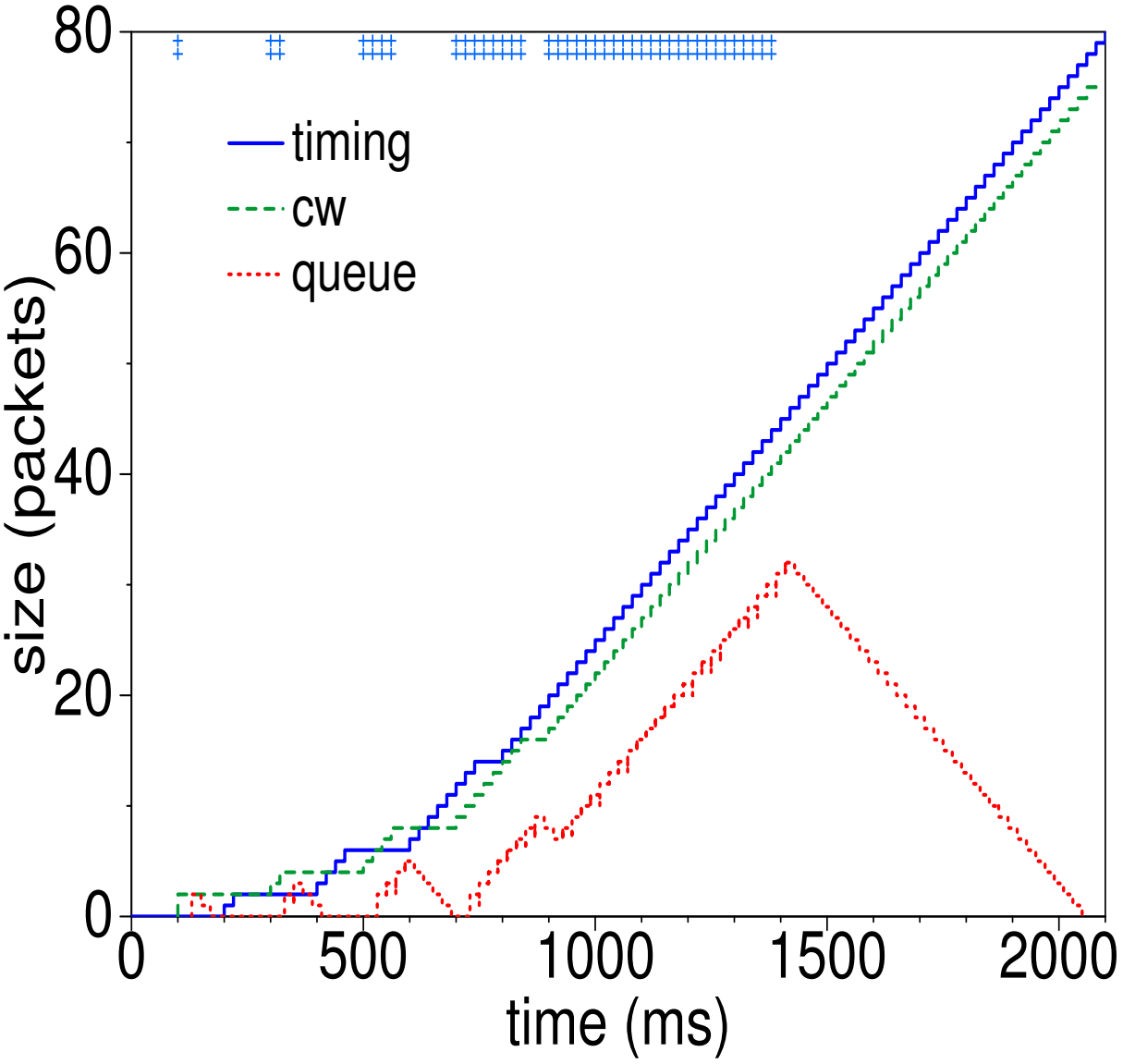
- In slow start, cw grows 1 packet per 1 ACK.
- Switch to CA when $cw > ssthresh$ or receive trip-dup.
- In CA, cw grows 1 packet per cw ACK.
- Model includes fast recovery.
- No timeouts, no delayed ACKs.

Model output



- $tt = 20$ms, $rtt = 200$ms, $bdp = 10$ packets.
- Timing chart shows total data received, cw, queue.
- Expect $cw = 2, 4, 8, 16,$ drop!

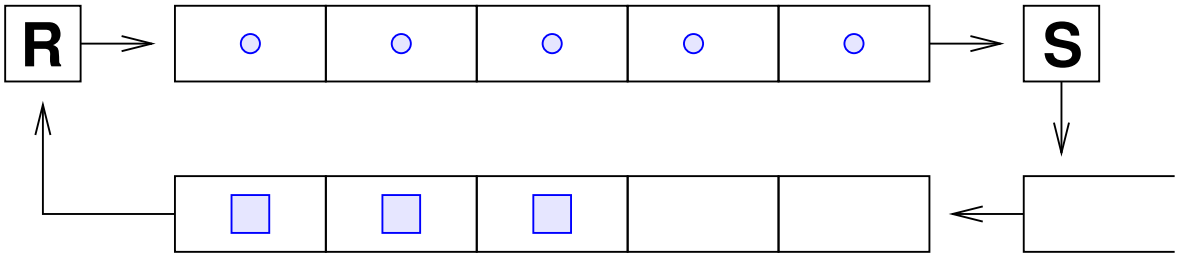
Self-clocking



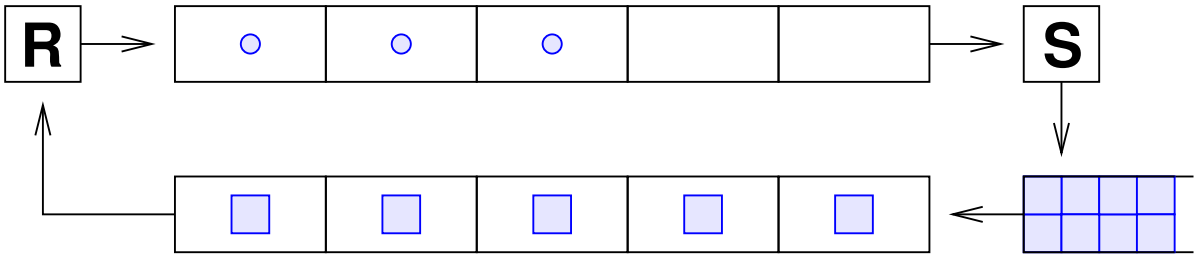
- After $cw > bdp$, cw grows linearly.
- Queue also grows linearly.
- Send rate $= 2bw$.

Transition to SC

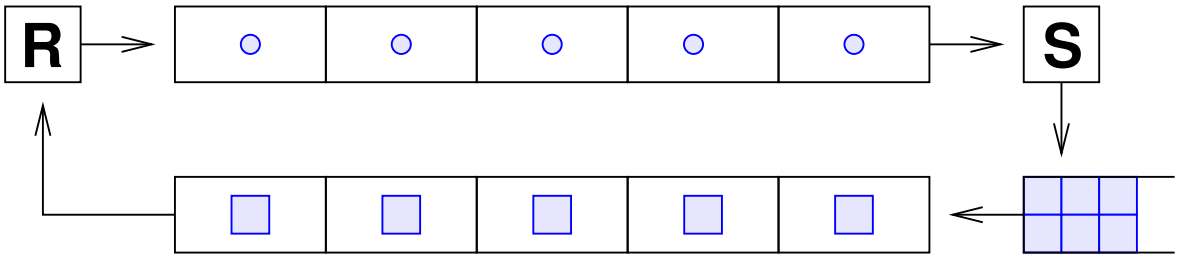
t = 0, cw = 8



t = 8, cw = 16



t = 10, cw = 16



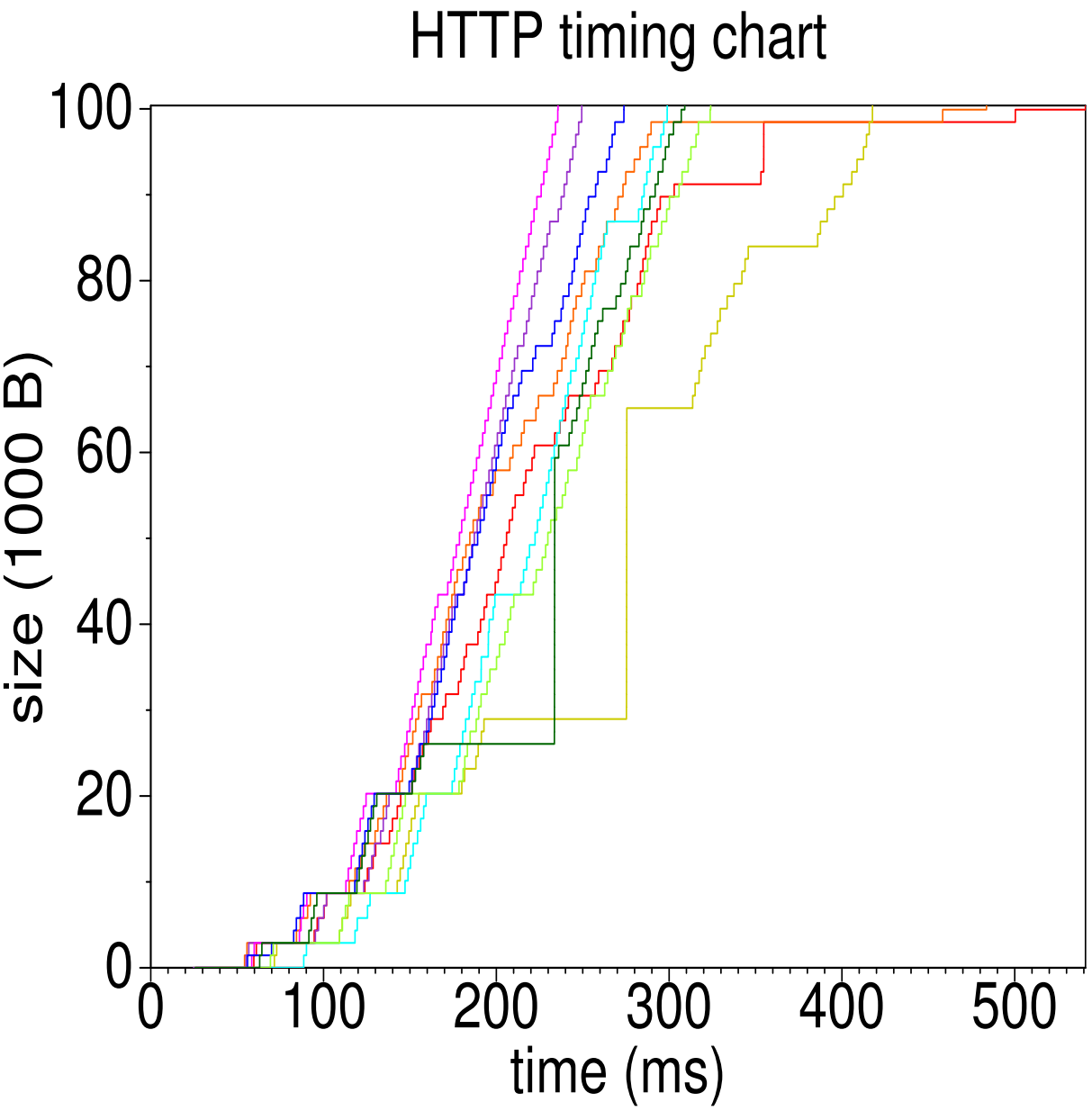
- Define $t = 0$ when $cw = 8$.
- Send 16 packets at $2bw$.
- At $t = 10$, $cw > bdp$, send rate $> bw$, no dropped packets.

Definition of SC

- SC = connection state where:
  - $cw \geq bdp$
  - send rate $\geq bw$
  - no endogenous drops.
- connection state $\neq$ sender state:

  Sender may be in slow start or congestion avoidance while connection is in SC.

Does this really happen?

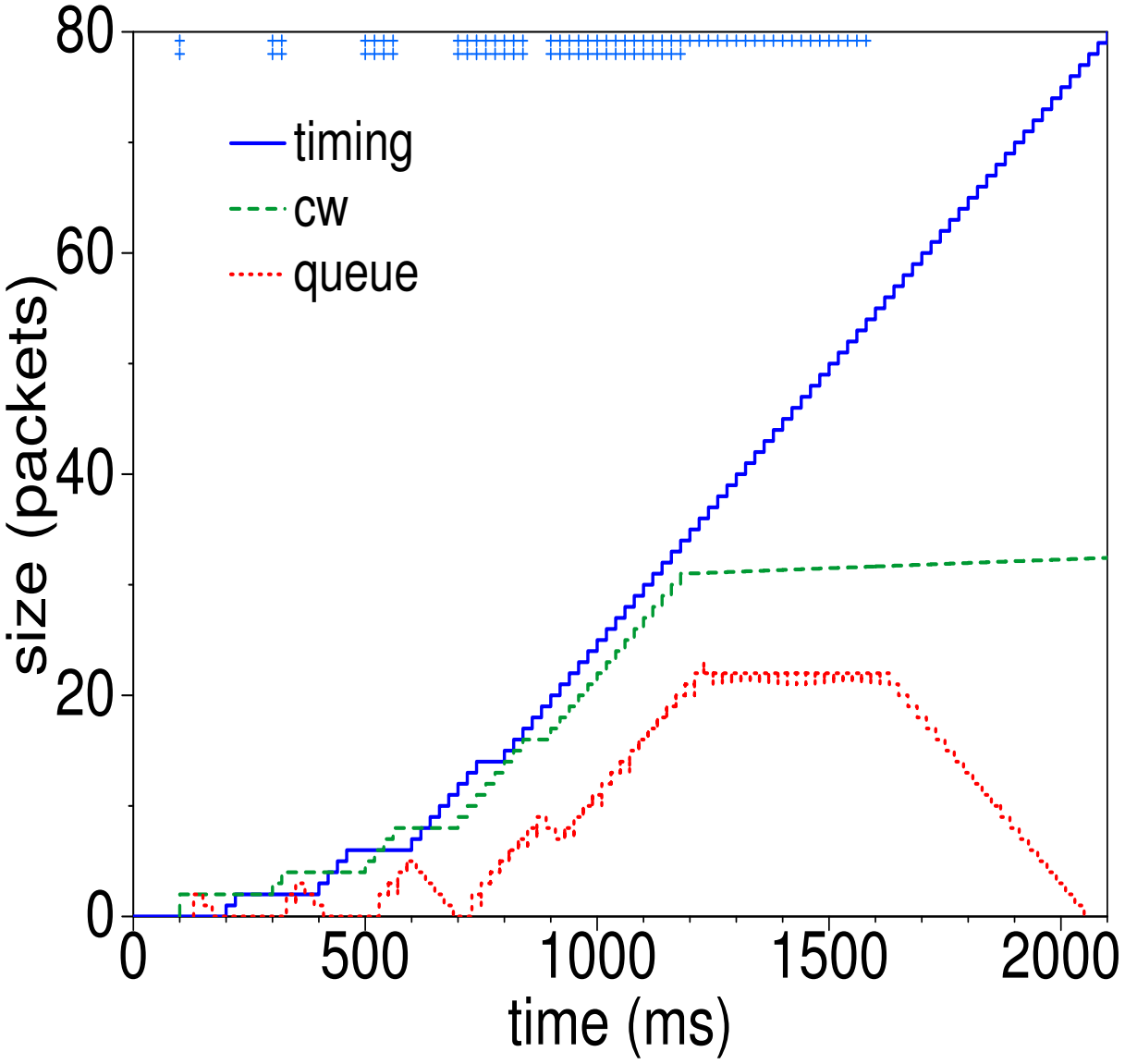


- Initiate long HTTP transfers.
- Record data arrivals.

Where are we?

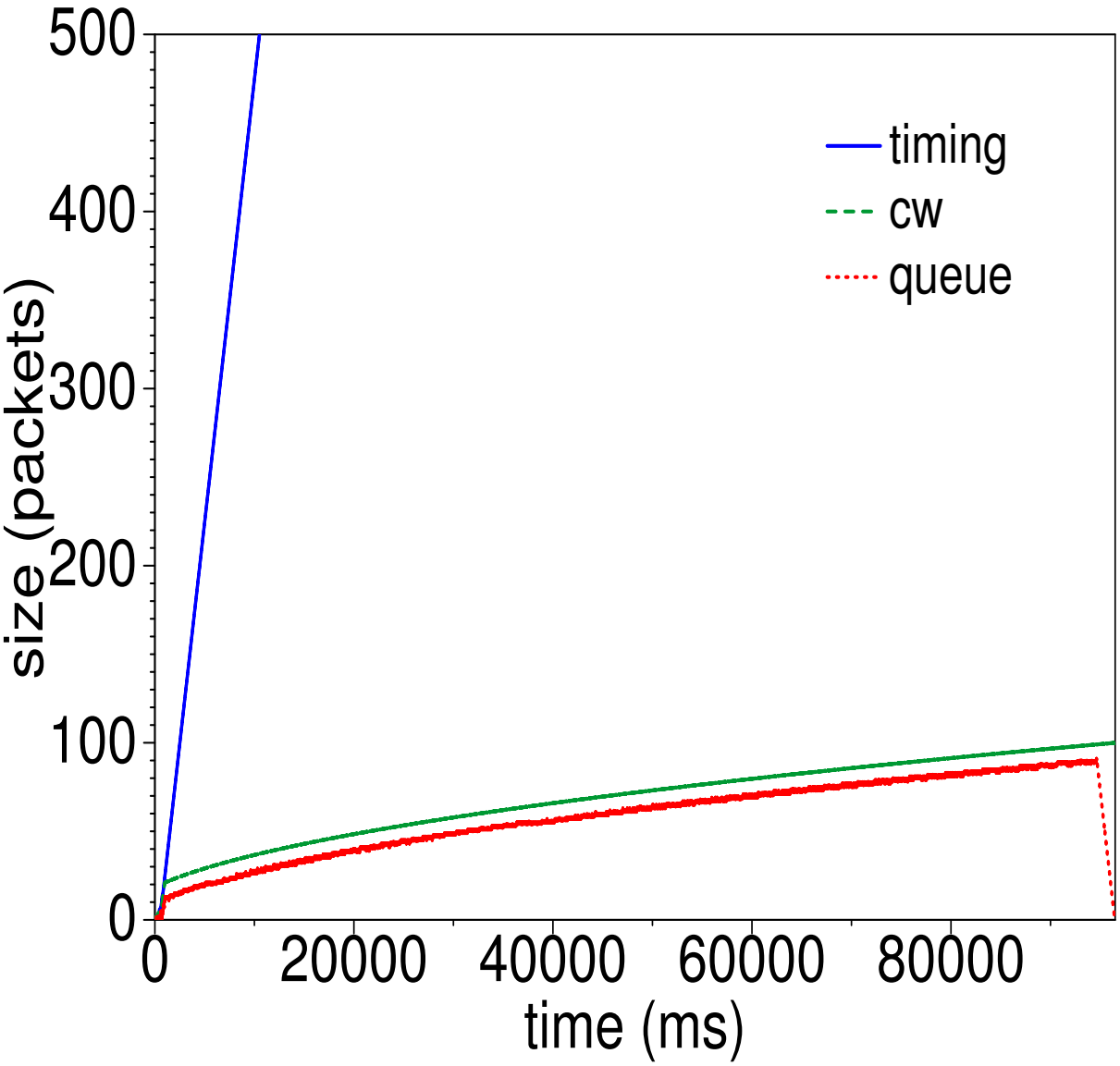
- Model explains transition from SS to SC.
- Measurements demonstrate existence of SC.
- Use model to explore the effect of:
  - $ssthresh$
  - Delays and dropped packets.
  - Bandwidth sharing.
- More measurements to verify results.

Slow start threshold



- When $cw > ssthresh$, sender switches to congestion avoidance.
- SC continues, but cw and queue grow as $\sqrt{t}$.

Slow start threshold

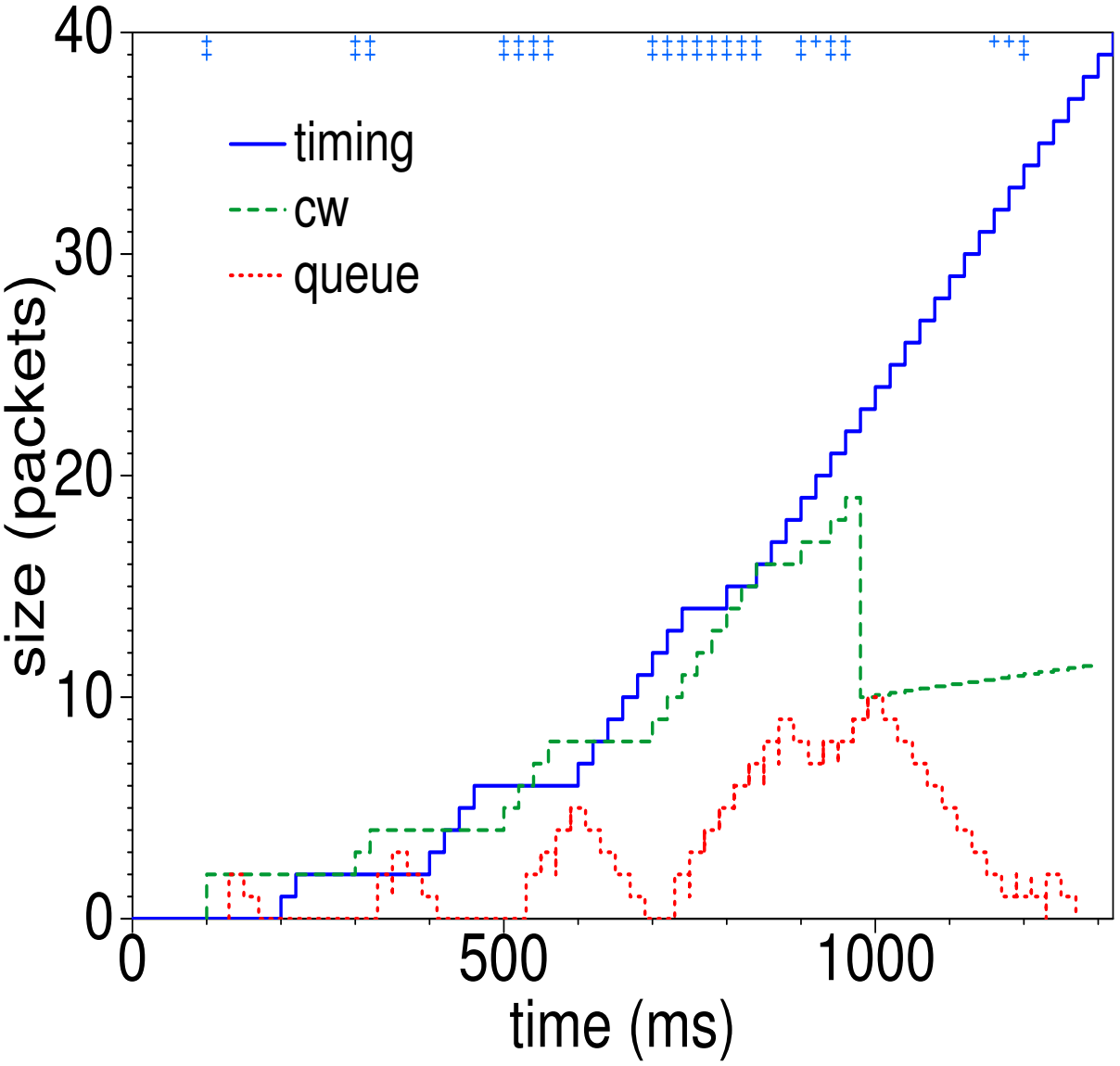


- If $ssthresh = 20$ and buffer $= 100$, we can send 4800 packets before overflow.

Capybara

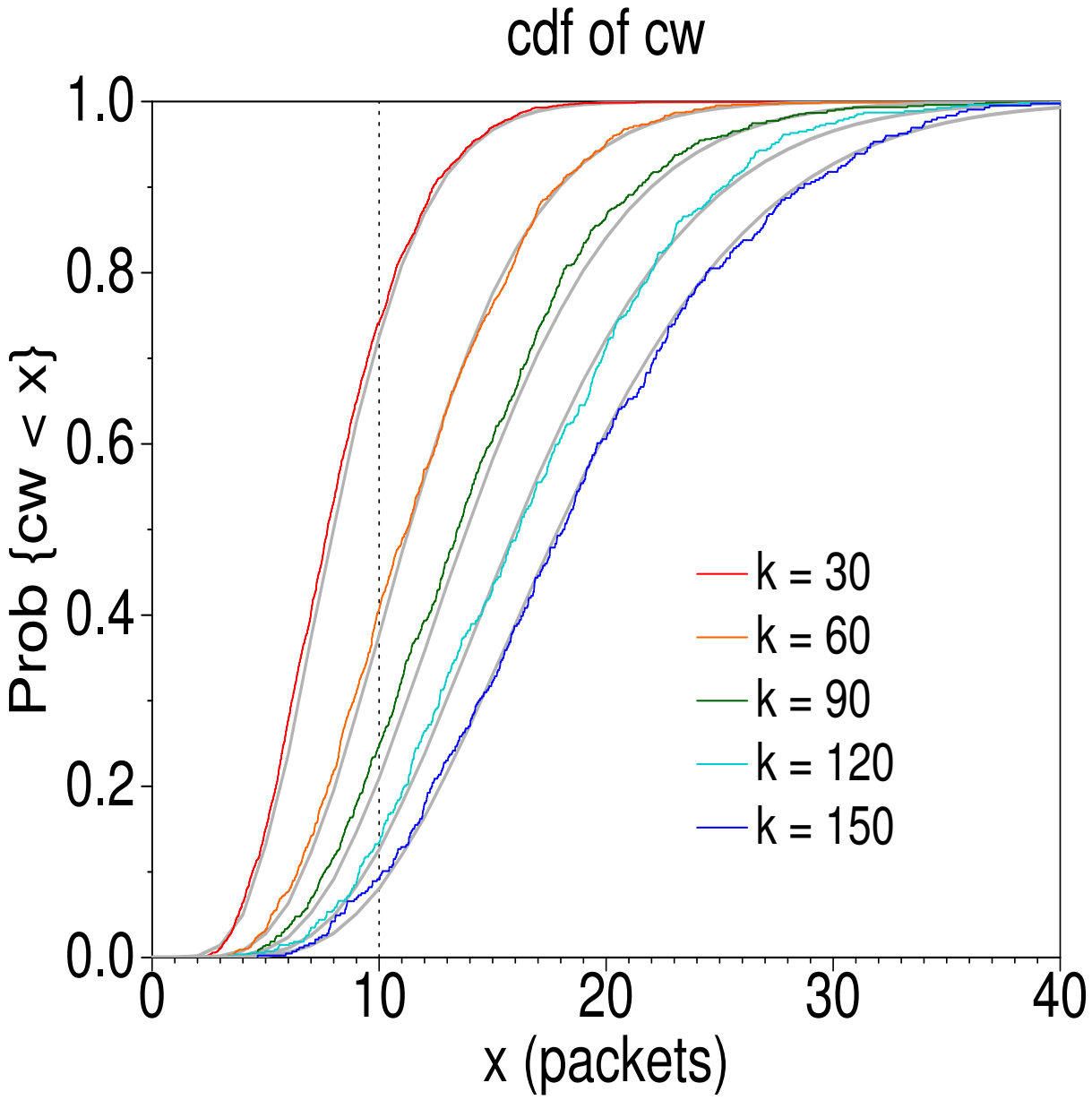
- mouse = short transfer in SS
- elephant = long transfer in AIMD
- capybara = long transfer in SC

Dropped packets



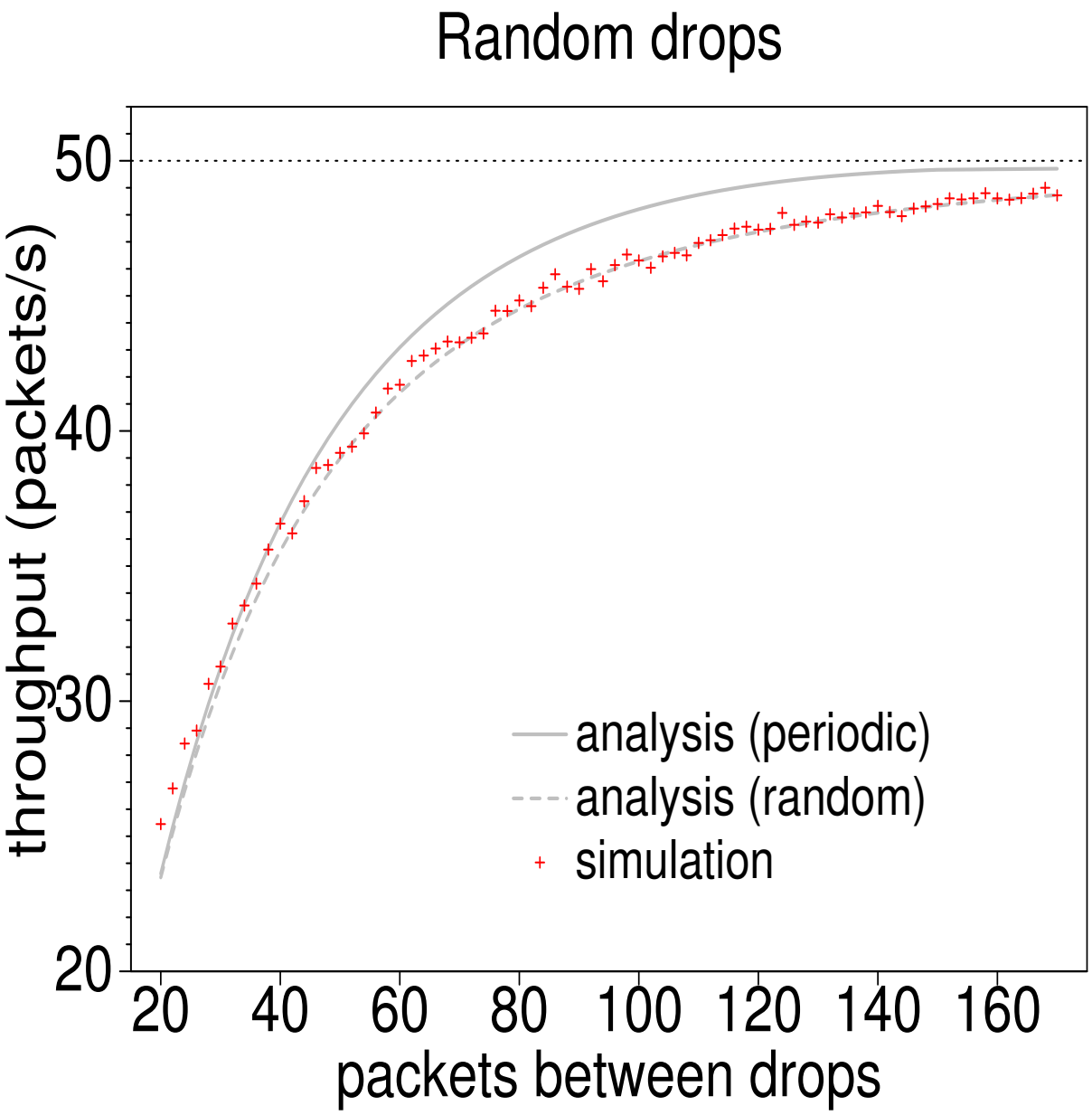
- Can SC survive a drop?
- Yes, if $cw \geq bdp$ before the queue drains.

Exogenous drops



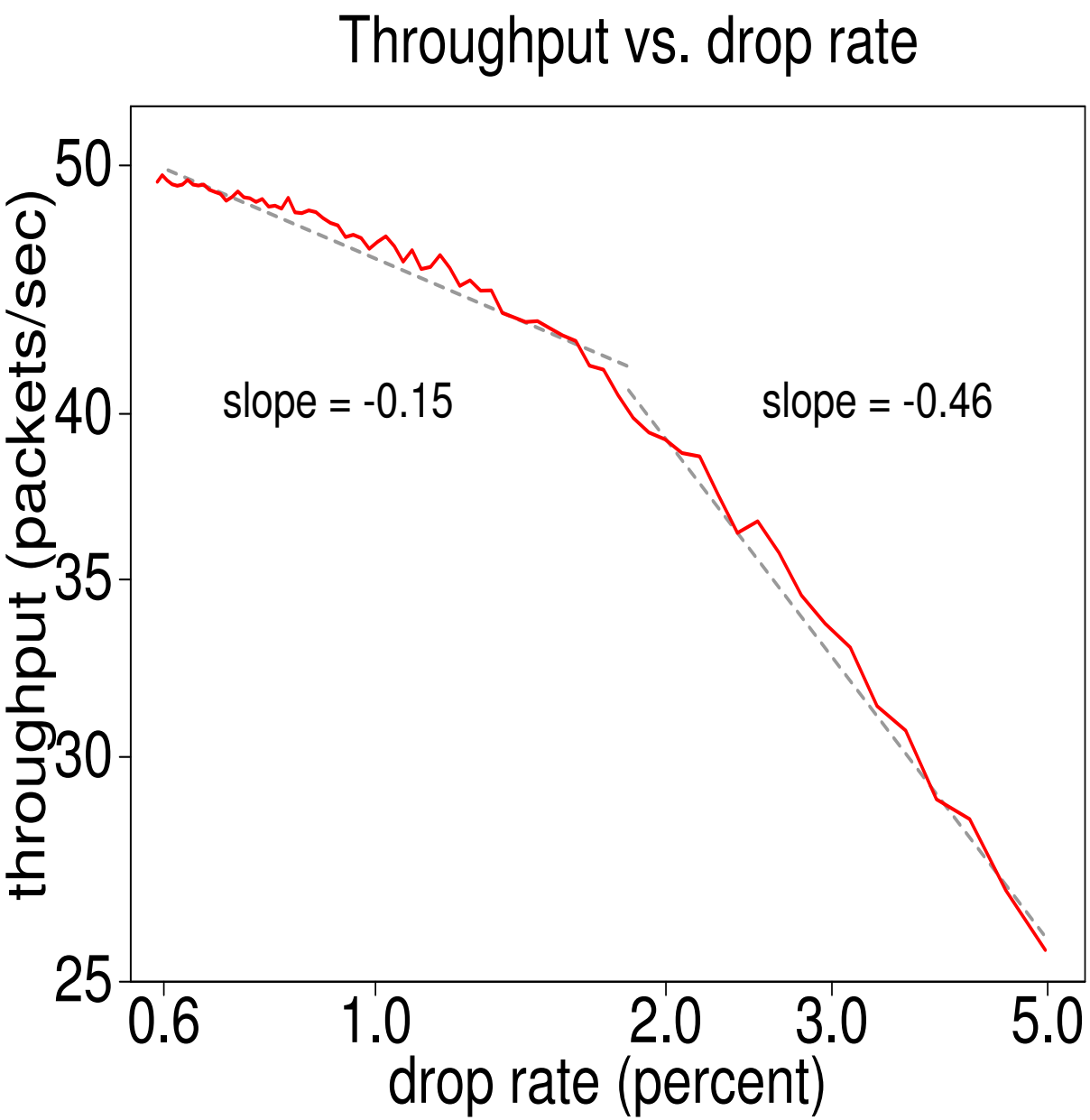
- $k = 1/p =$ average packets between drops.
- Large $k \Rightarrow$ large cw.
- Even with $k = 30$, $cw > bdp$ sometimes.

Throughput versus drop rate



- Low $k \Rightarrow$ high drop rate $\Rightarrow cw$-bound.
- High $k \Rightarrow$ low drop rate $\Rightarrow bw$-bound.

Throughput versus drop rate

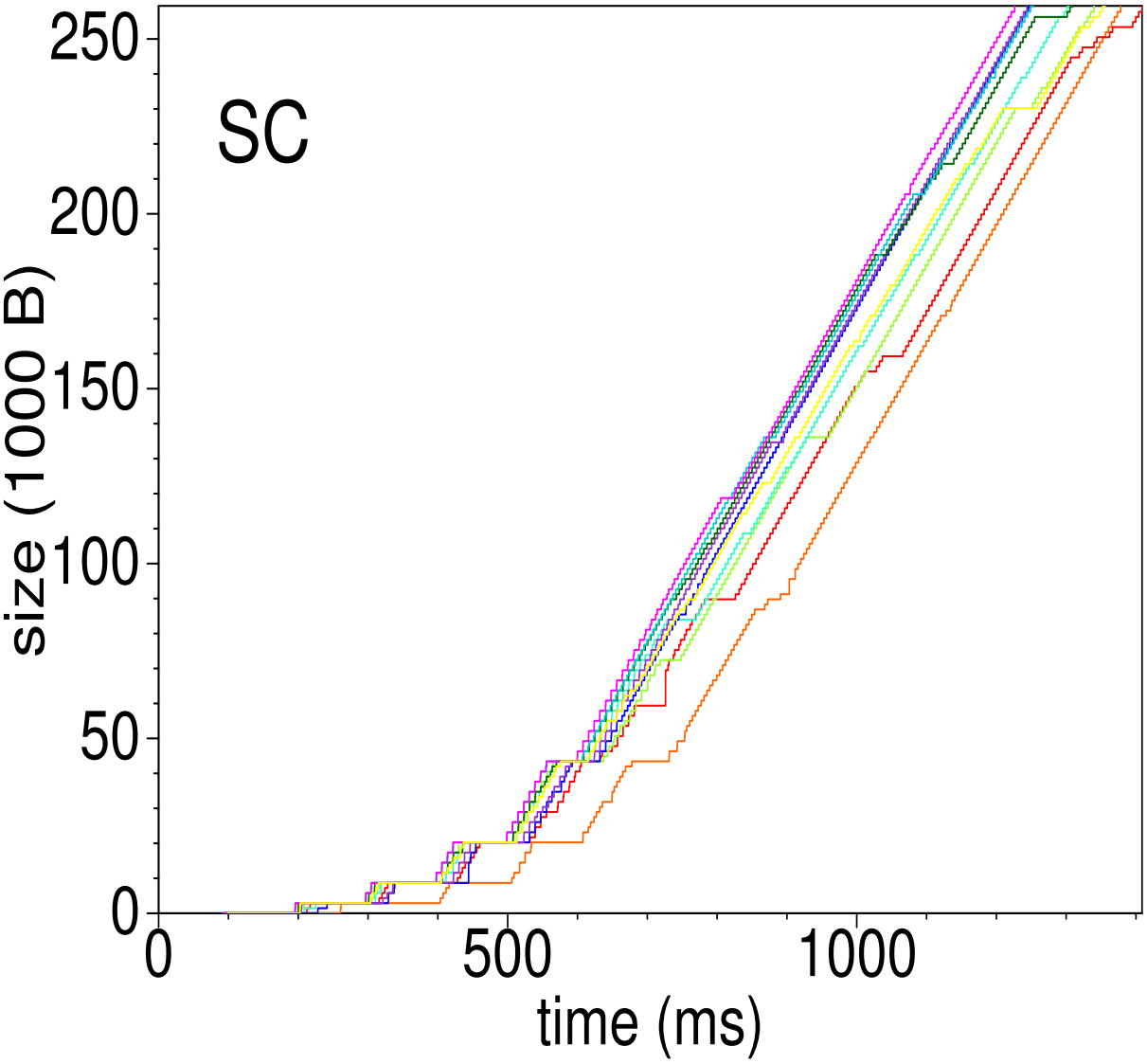


- Log-log scale.
- High drop rate, $th \sim p^{-0.46}$
- Low drop rate, no $1/\sqrt{p}$ rule.

Measurements

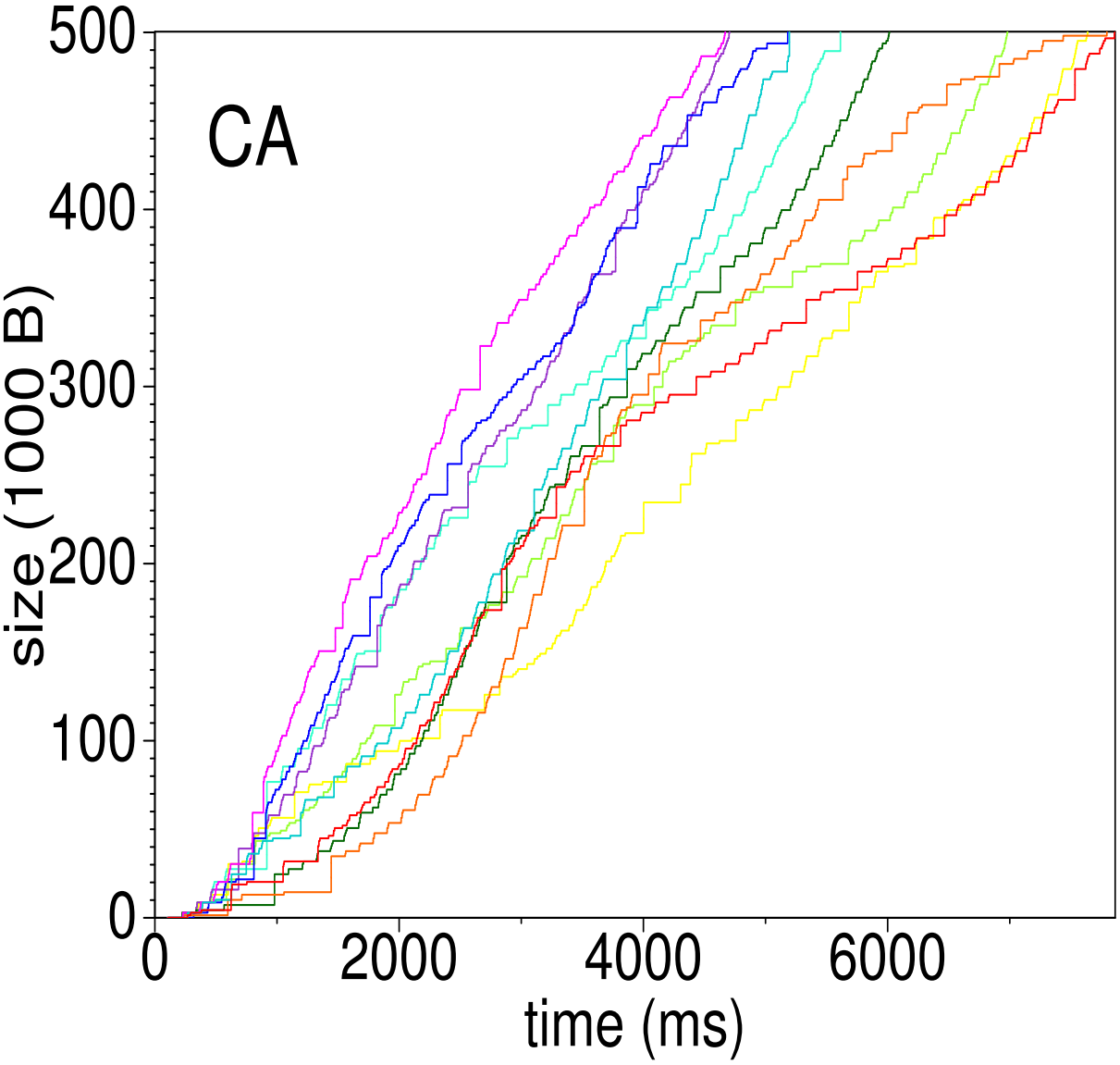
- Find 100 busy Web sites with large files (100–500 KB).
- Download 10 times.
- Classify timing charts (visual and statistical).

Self-clocking



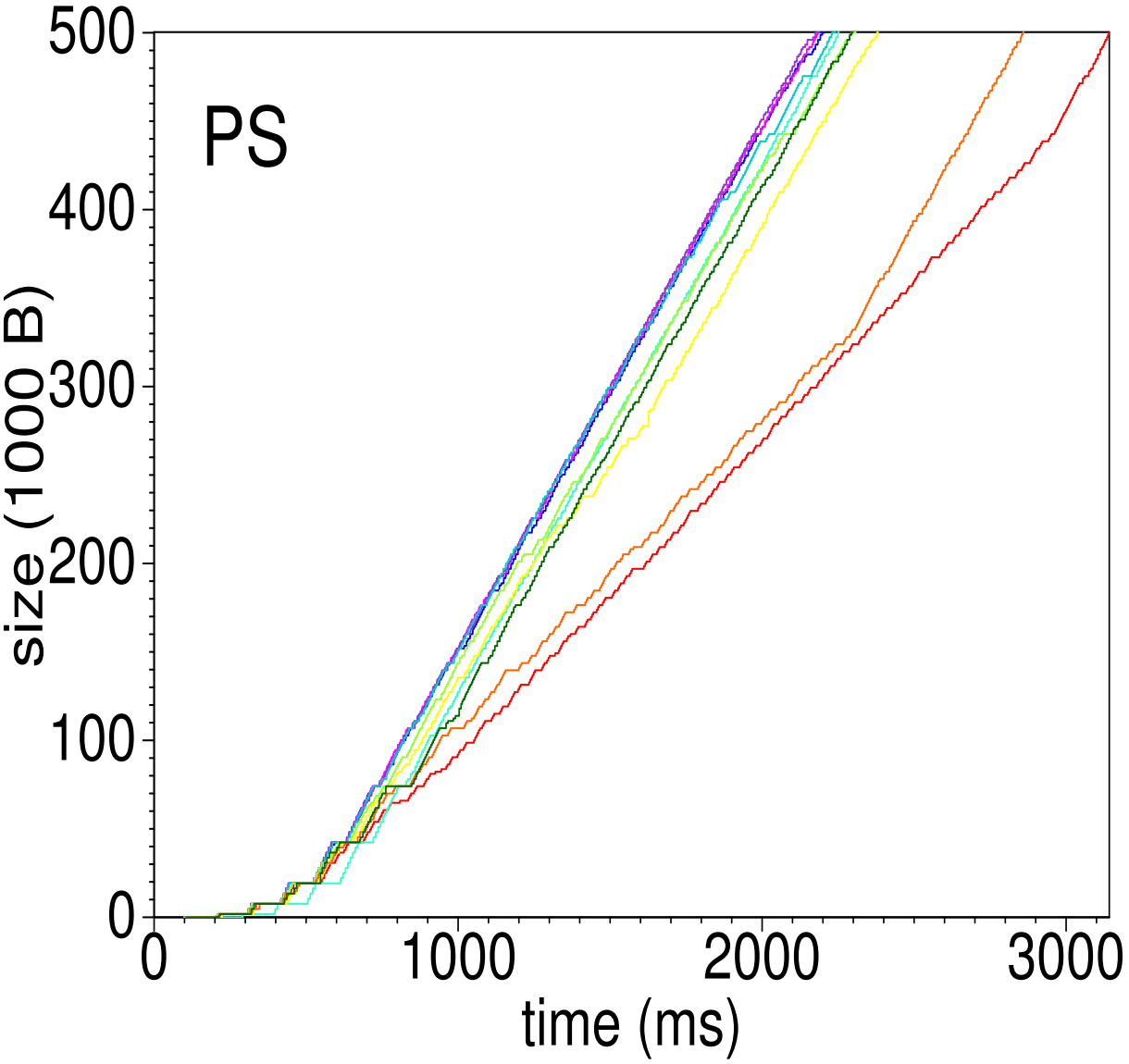
- 61 paths where SC is most common state.

Congestion avoidance



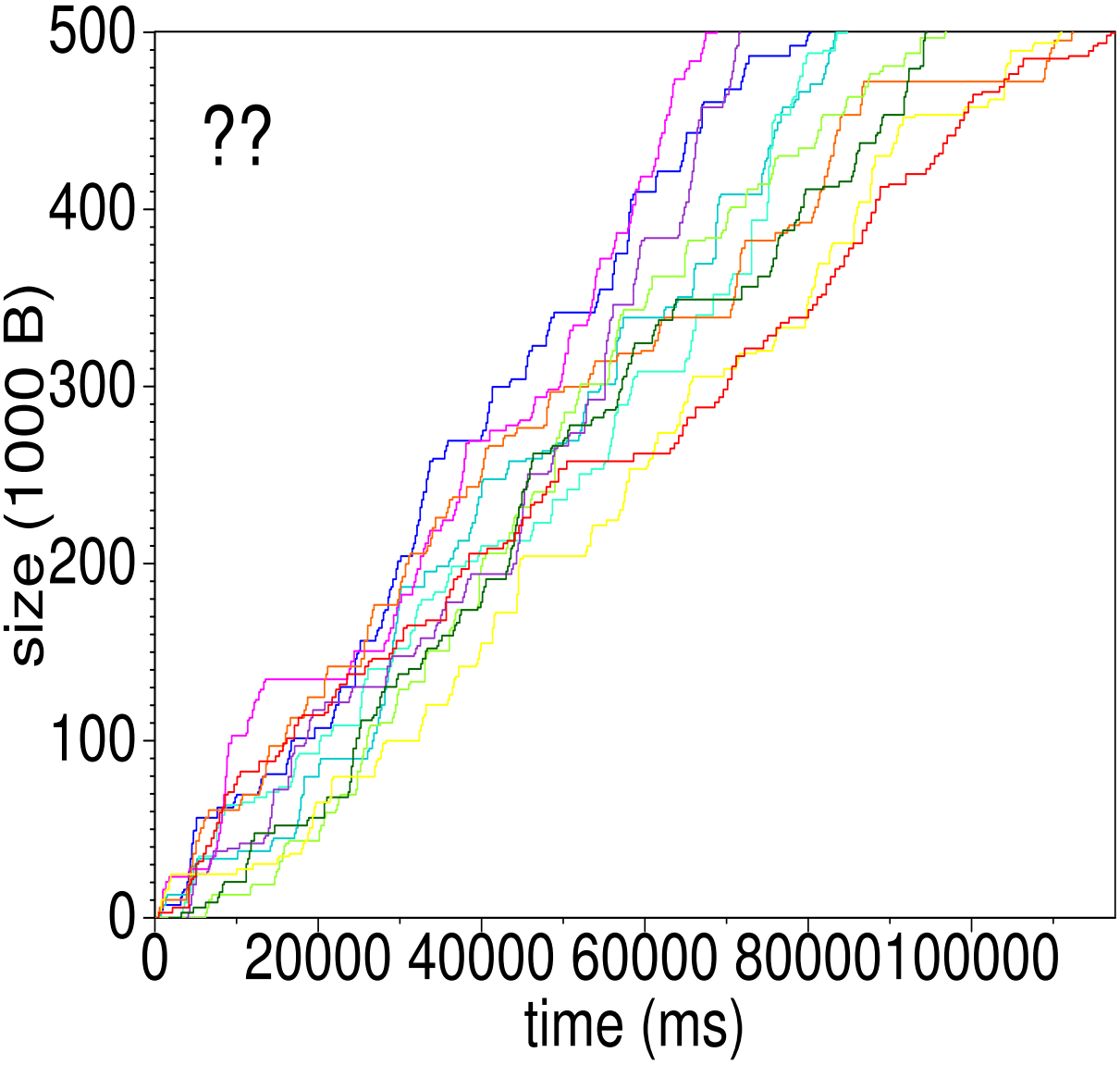
- 7 paths where AIMD is the most common state.

Bandwidth sharing



- Evidence of bandwidth sharing.

Who knows?



- 2 cases of who knows what.
- 10 buffer-limited.
- 2 opportunity-limited.
- 1 application-limited (thttpd).
- 17 file not available.

Summary

- 83 successful downloads.
- 61 paths (73%) where SC is common.
- Client on college campus, behind 2 T1s (2.8 Mbps), tends to see low bdp.

Client	SC	CA	BL	OL	??	AL	Total
1	61	7	10	2	2	1	83

Table 1: Path classifications.

More measurements

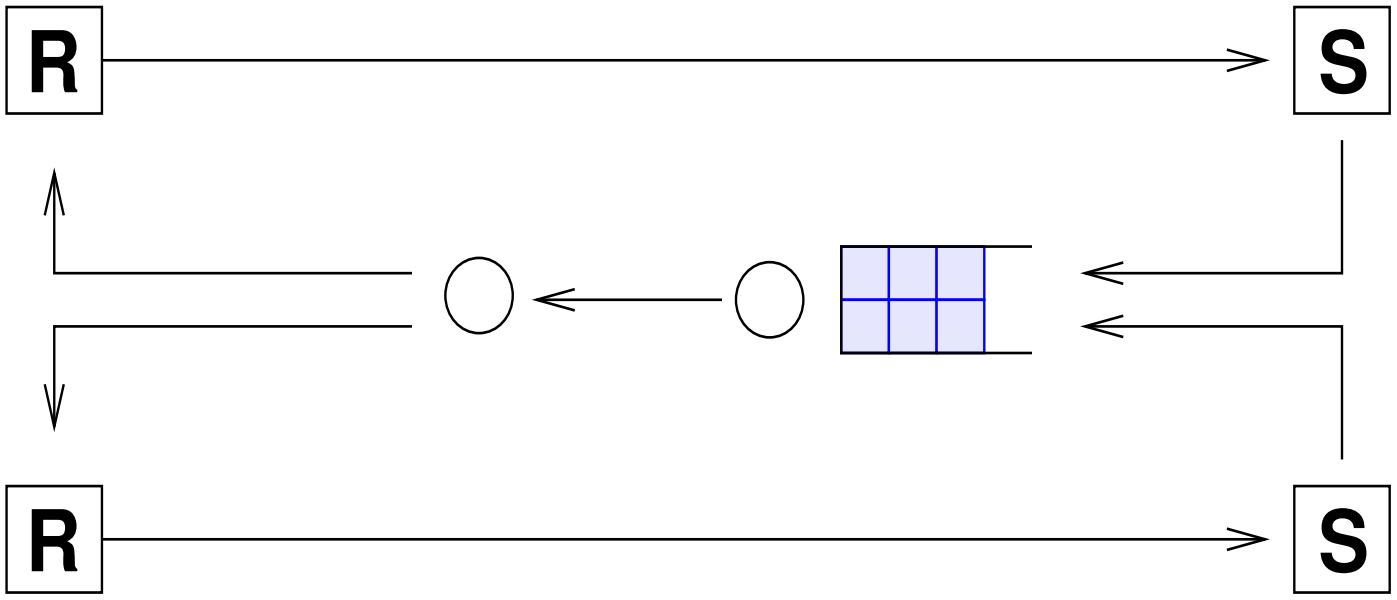
- Client 2, house with cable modem (almost 8 Mbps).
- Client 3, university (45 Mbps and 1 Gbps).

Client	SC	CA	BL	OL	??	AL	Total
1	61	7	10	2	2	1	83
2	54	6	11	9	7	1	88
3	24	24	44	17	24	0	133

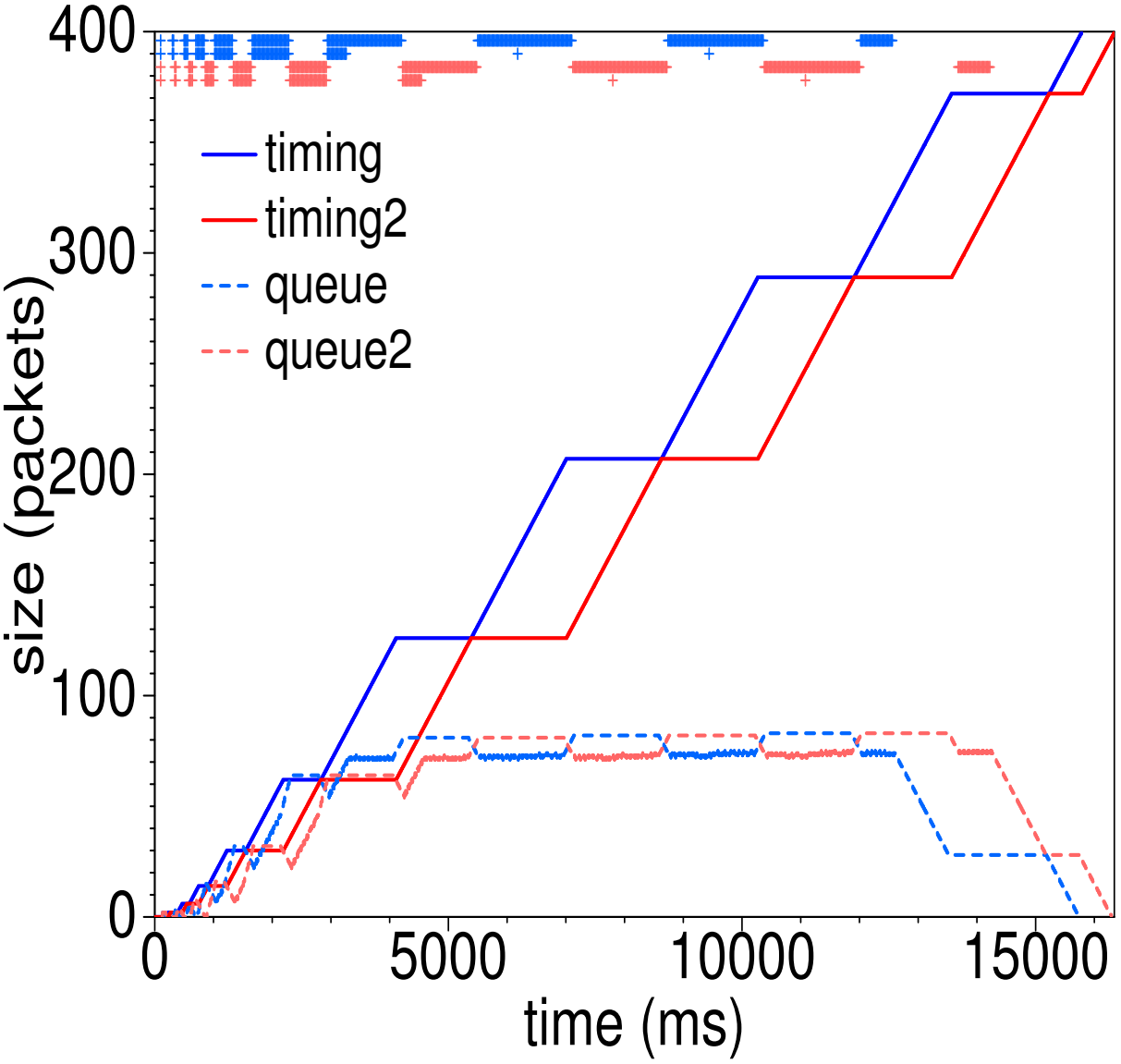
Table 2: Path classifications.

Bandwidth sharing

- Extend the model for n transfers sharing the bottleneck.
- Starting with $n = 2$:
  same rtt and ss, different rtt, different ss.

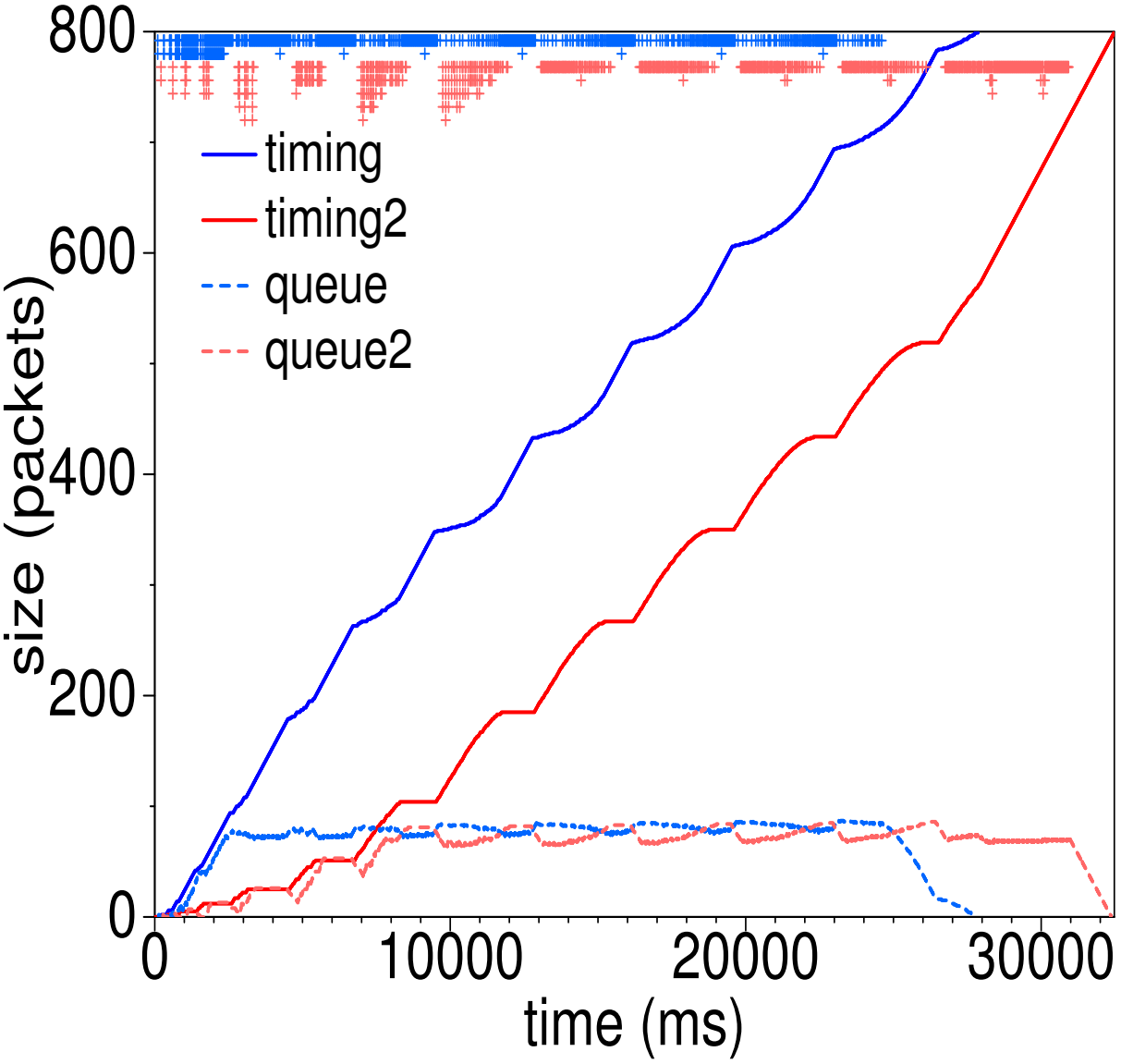


Periodic sharing



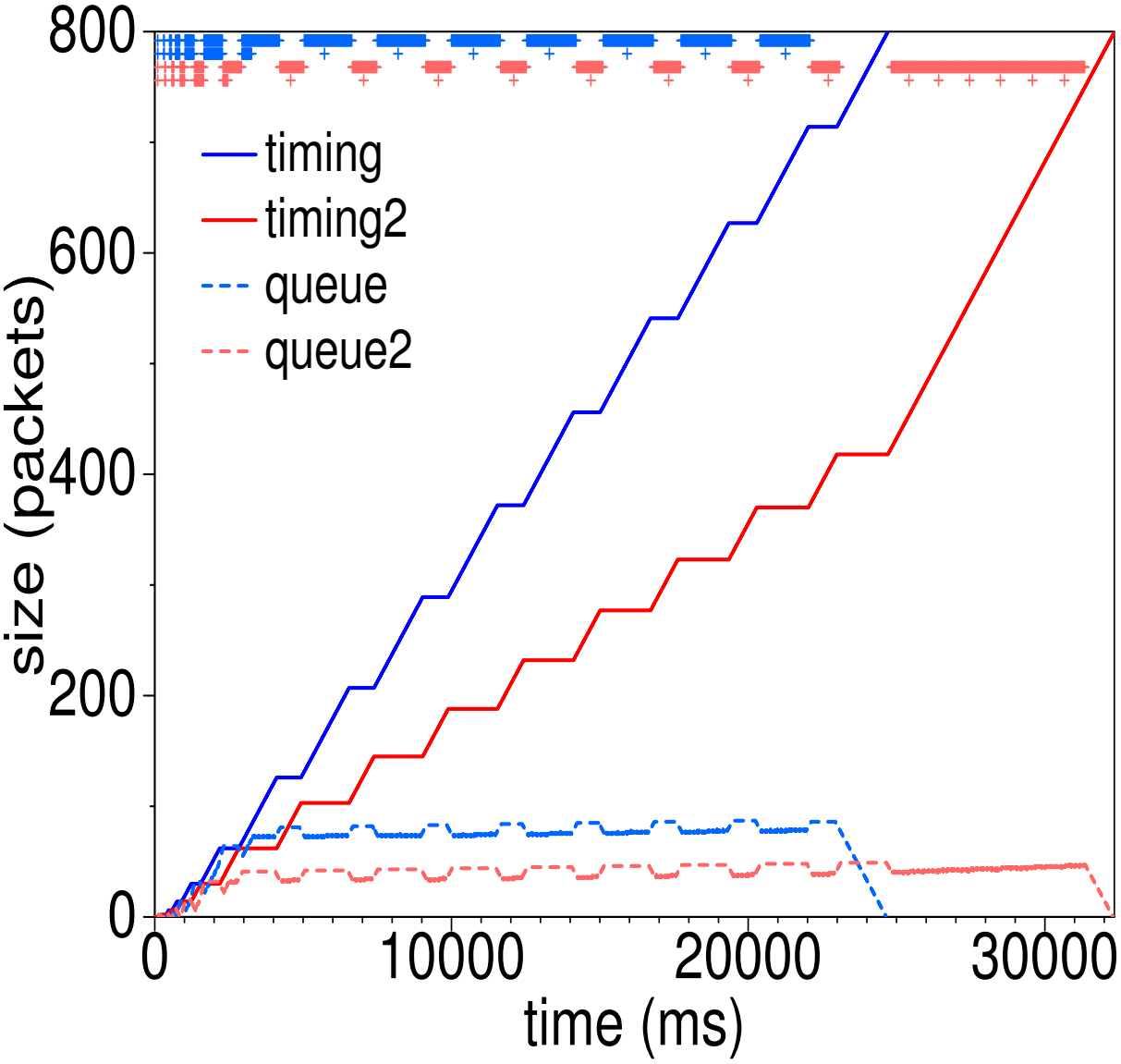
- $rtt_1 = rtt_2 = 200$ ms.
- $ss_1 = ss_2 = 80$ packets.
- Periodic sharing (PS).

Periodic sharing



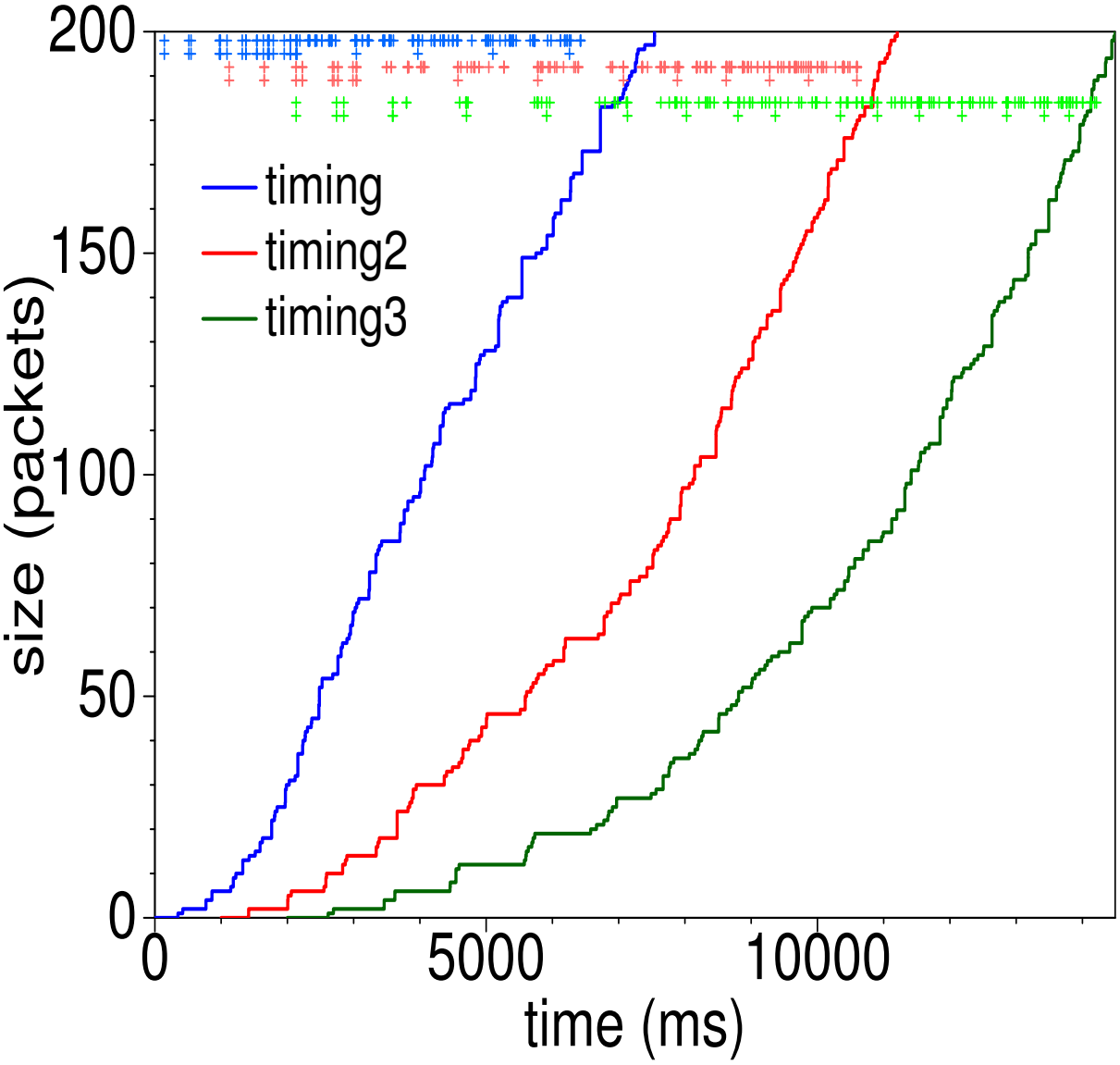
- $rtt_1 = 200$ ms.
- $rtt_2 = 400$ ms.
- $ss_1 = ss_2 = 80$ packets.
- Same throughput!
- Each connection sends cw_i per period.

Periodic sharing



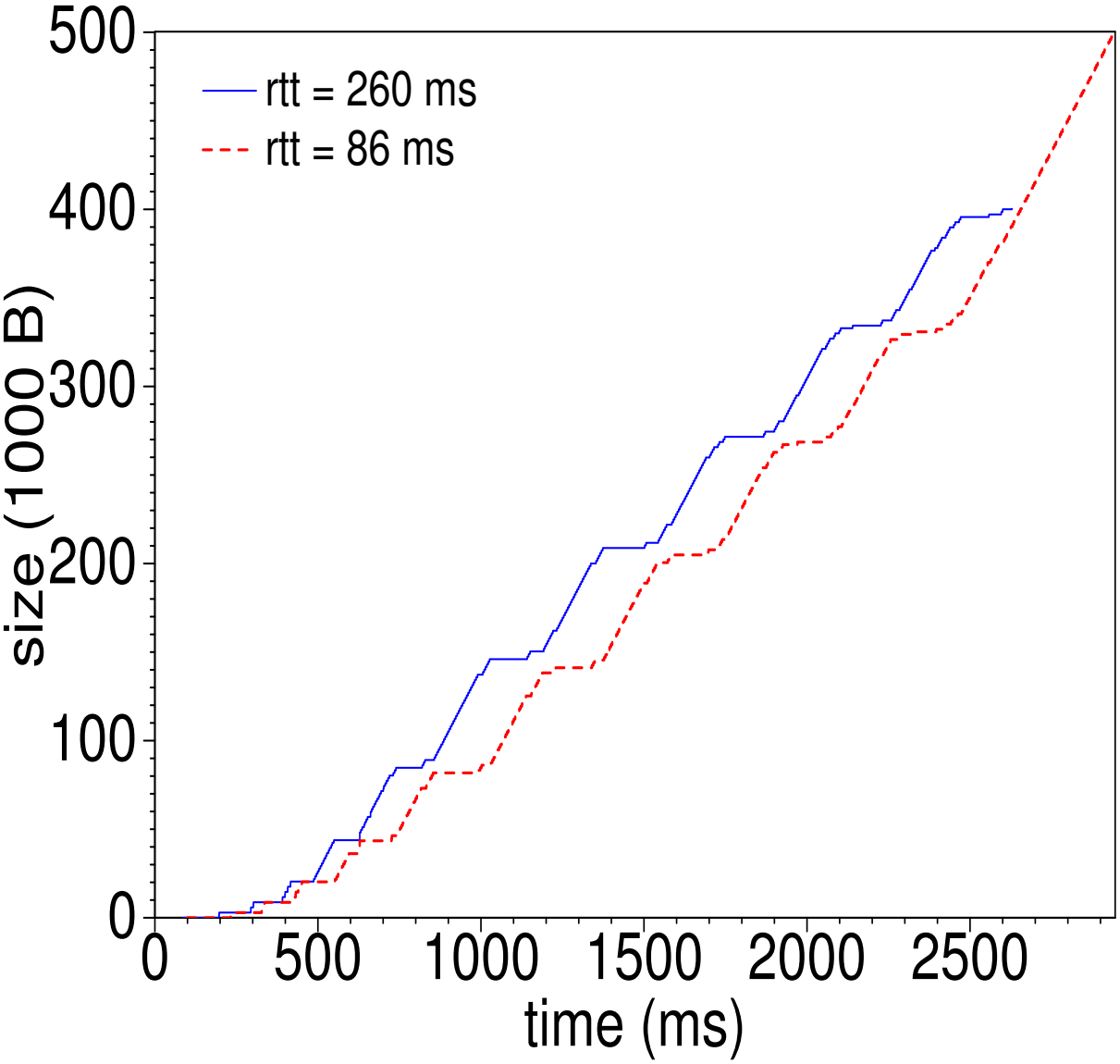
- $rtt_1 = rtt_2 = 200$ ms.
- $ss_1 = 80$.
- $ss_2 = 40$.
- $throughput \sim$ queue residency.
- When one ends, the other takes over.

Periodic sharing



- Random delays obscure periodic behavior.
- Hard to identify PS from measurements.

Periodic sharing

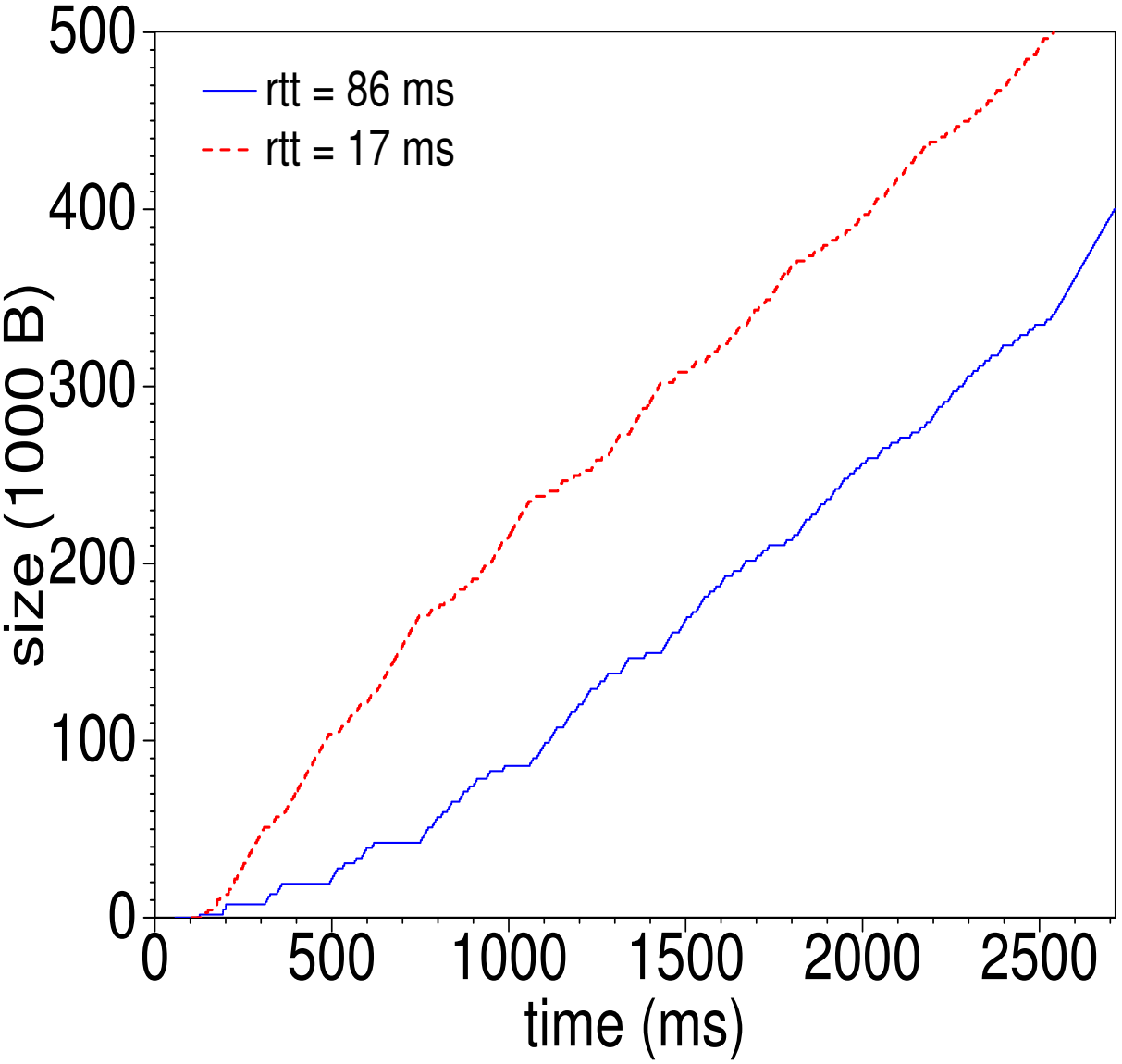


- But some measurements show unambiguous PS.
- Even with different rtt.

Measuring PS

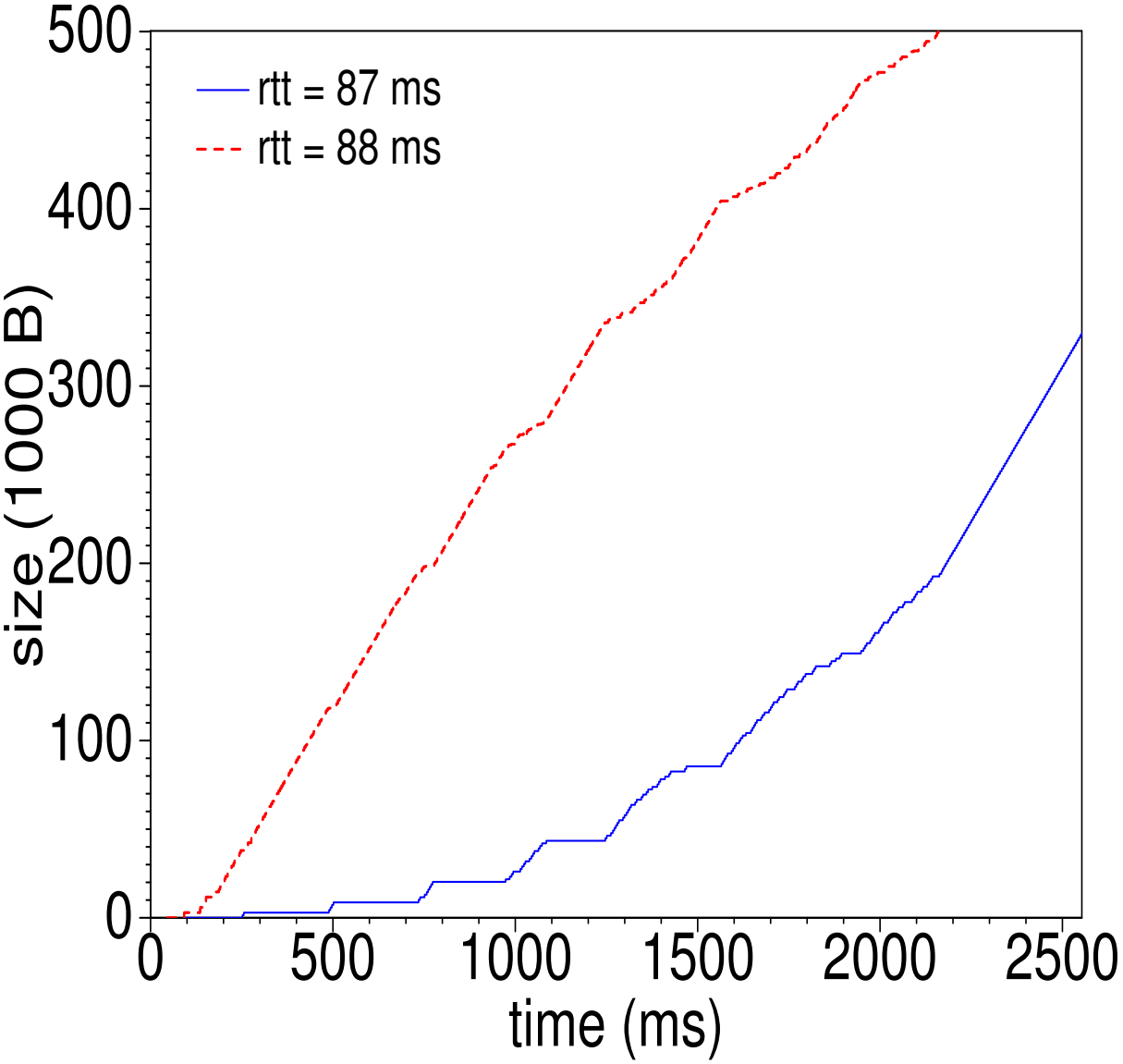
- Find 25 busy Web sites with very large files.
- Choose 2 at random and initiate simultaneous downloads.
- Run 100 pairs.
- Classify timing charts (visual and statistical).

Periodic sharing



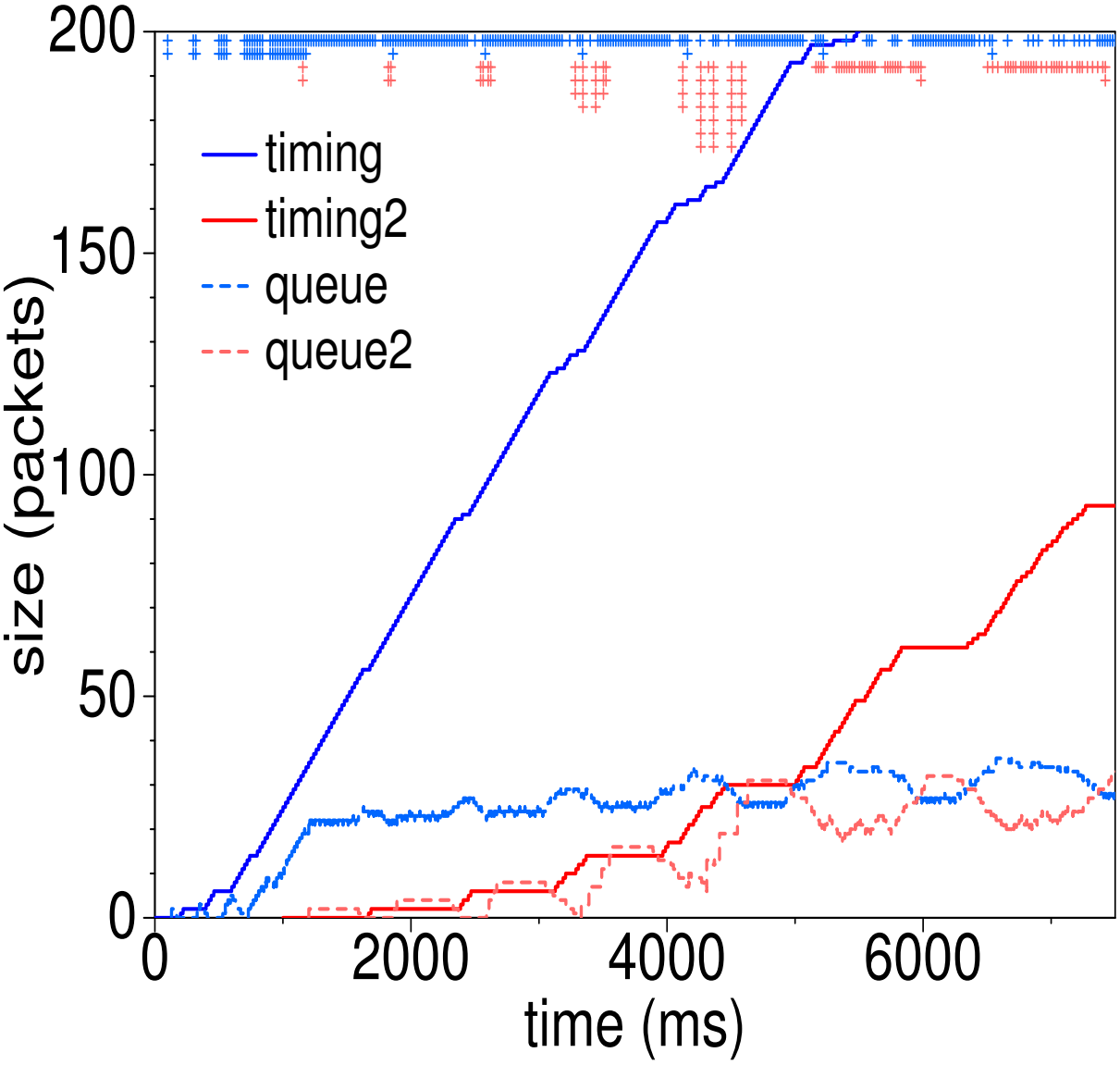
- Very different rtt.
- Same throughput.
- Characteristic scalloping.
- When one ends, the other takes over.

Periodic sharing



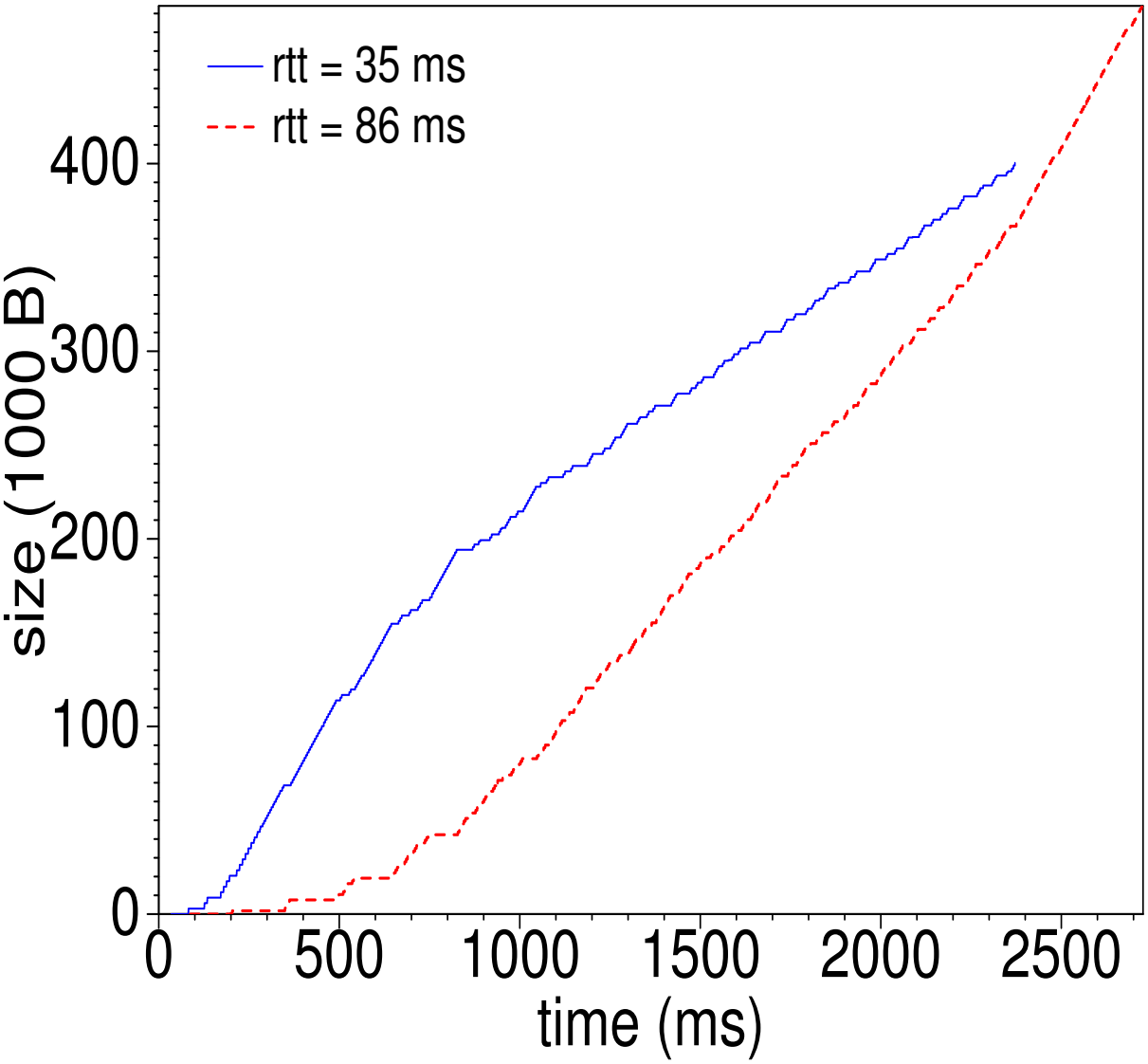
- Same rtt.
- Second transfer to start is delayed by the queue induced by the first...

Periodic sharing



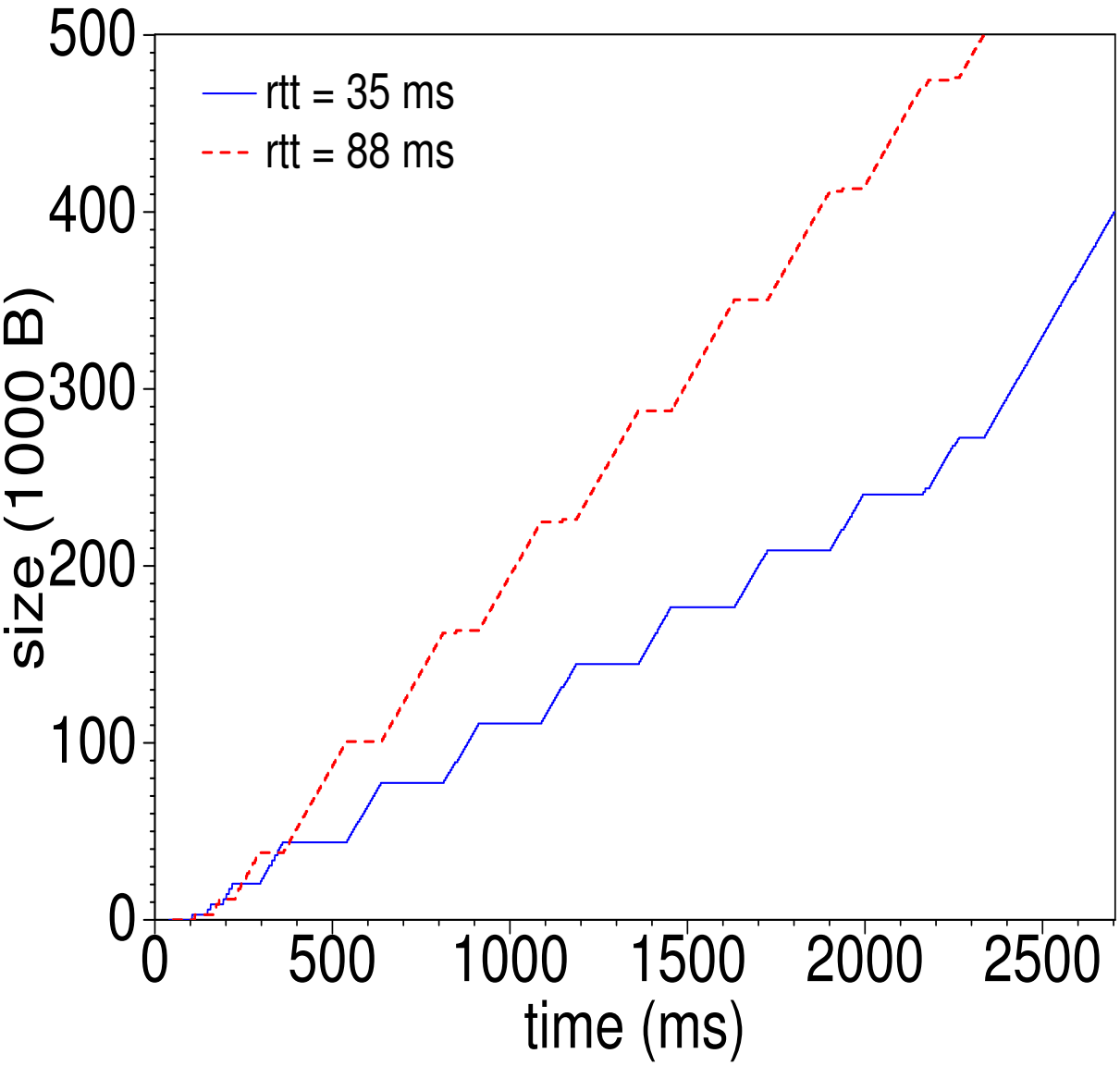
- ... as predicted by the model.
- Connections share the bottleneck bw in proportion to queue residency.

Periodic sharing



- Faster rtt, earlier start, lower throughput?

Periodic sharing



- Alone, both servers reach SC.
- In PS, their queue residency is limited by the send buffer.

Periodic sharing

- Of 100 pairs, 3 don't overlap and 6 don't share a bottleneck.
- 68 of the remaining 93 cases (75%) show clear PS.
- Another 11 cases show aperiodic queue sharing.
- In 12 cases, one of the connections is in CA.

Results

- For hosts behind slow links, SC is common.
- Even for well-connected hosts, SC happens.
- PS is a common mode of bandwidth sharing (at least for low bdp paths).

Eight Things

Eight things we know about TCP.

1. During slow start, cw grows exponentially.
   cw grows exponentially until $cw > bdp$,
   then it grows linearly until $cw > ssthresh$.

2. In steady state, cw grows linearly between drops.
   In SC, cw grows as $\sqrt{t}$.

Eight Things

3. When cw exceeds bdp, packets get dropped.
   cw can exceed bdp for a long time without overflowing the buffer at the bottleneck.

4. After a drop, TCP decreases the send rate.
   In SC the sender might pause, then resume at higher rate.

Eight Things

5. Steady-state behavior for long transfers is AIMD.
   On many paths, the most common steady-state behavior is SC, which is neither AI nor MD.

6. In steady state, $throughput \sim 1/\sqrt{p}$.
   For low drop rates, throughput is less sensitive to p.

Eight Things

7. When connections share a bottleneck, $throughput \sim 1/rtt$.
   In PS, connections share in proportion to queue residency, which depends on start time, $ssthresh$ and send/receive buffers.

8. TCP is slow to discover increased available capacity.
   By keeping excess packets in queue, connections absorb capacity instantly.

Discussion

Is self-clocking a good thing?

- SC does most of what TCP Vegas tries to do.
- Persistent queues make long transfers robust...
- ... but they are a disaster for latency.

Discussion

- $ssthresh$ affects bandwidth sharing!
- If $ssthresh = bdp$, then $throughput \sim rtt$.
- $ssthresh$ arms race?

Discussion

- Send and receive buffers affect bandwidth sharing!
- Buffer arms race?

Thanks

- Most of this work done at BU.
- Thanks to John Byers, Dhiman Barman and others.
- More at `allendowney.com/research/tcp`