An Empirical Model of TCP Performance

Allen B. Downey

Olin College of Engineering

TCP Performance

Goal: model and predict transfer times.

- Understand TCP performance
  - Improve TCP?
  - Improve the network?
- Interactive applications.
  - Predict duration of downloads.
  - Select mirror site.
- Distributed applications.
  - Resource selection.
  - Scheduling.

Goals

Complementary questions:

- What do we need to know about a network path?
- What can we measure about a network path?

Complementary goals:

- Maximize accuracy.
- Minimize measurement.

Two approaches

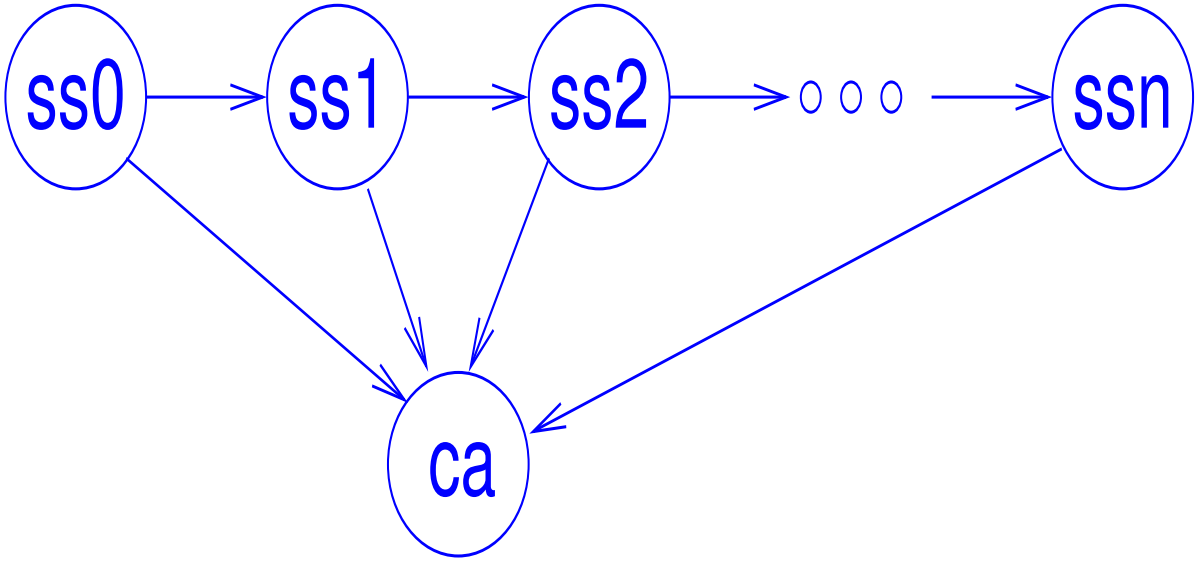
- History-based.
  - Pro: Good statistical description.
  - Con: Availability of data?
    - Passive: Might not have seen what you need.
    - Active: Obtrusive?
- Model-based.
  - Pro: small measurements $\Rightarrow$ parameters $\Rightarrow$ model $\Rightarrow$ predictions
  - Con: single-value predictions.
  - Con: incomplete models, so far.

TCP Models

- Several models for mice (never leave slow start).
- Several models for elephants (ignore slow start).
- For medium size ($bdp < size < 10bdp$), performance depends on slow start and steady state.
- bdp commonly 4 KB – 128 KB.
- Maybe 40% of TCP transfers are in this range.

Hybrid model

- Explicit model of slow start.
- History-based prediction of steady state throughput.



Model parameters

So what do we need to know?

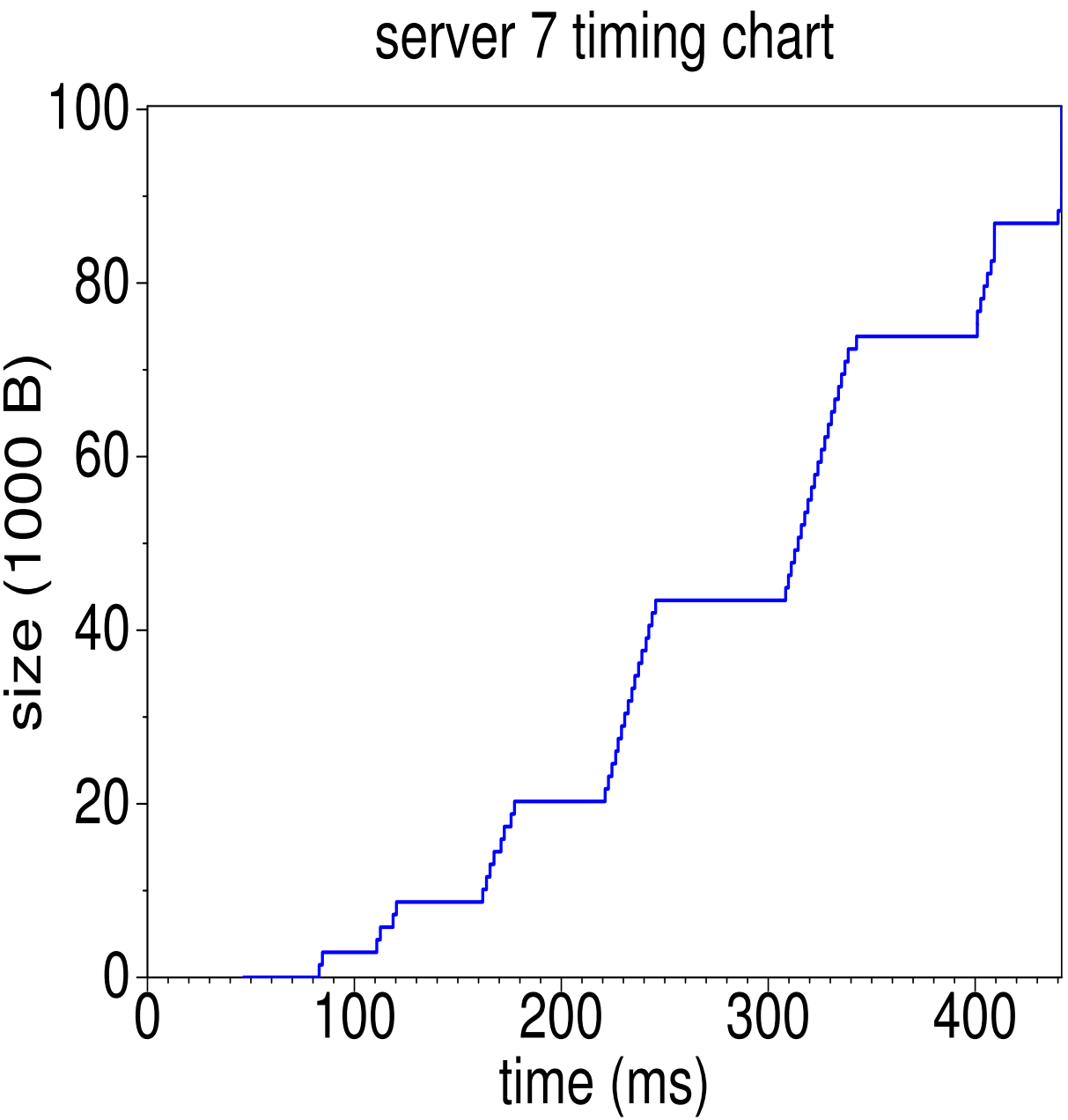
- State transition probabilities.
- Distribution of rtt.
- cw_1, cw_2, cw_3 ...
- Distribution of steady-state throughput.

Can we measure these parameters?

Measurement

- Application-level HTTP timing (instrumented `wget`).
  - Pro: easy to implement.
  - Con: some timing inaccuracy, lost information.

Timing chart



- Plot bytes read vs. time.
- Immediately, we can estimate rtt, cw, bw and bdp.
- And we can infer TCP state.

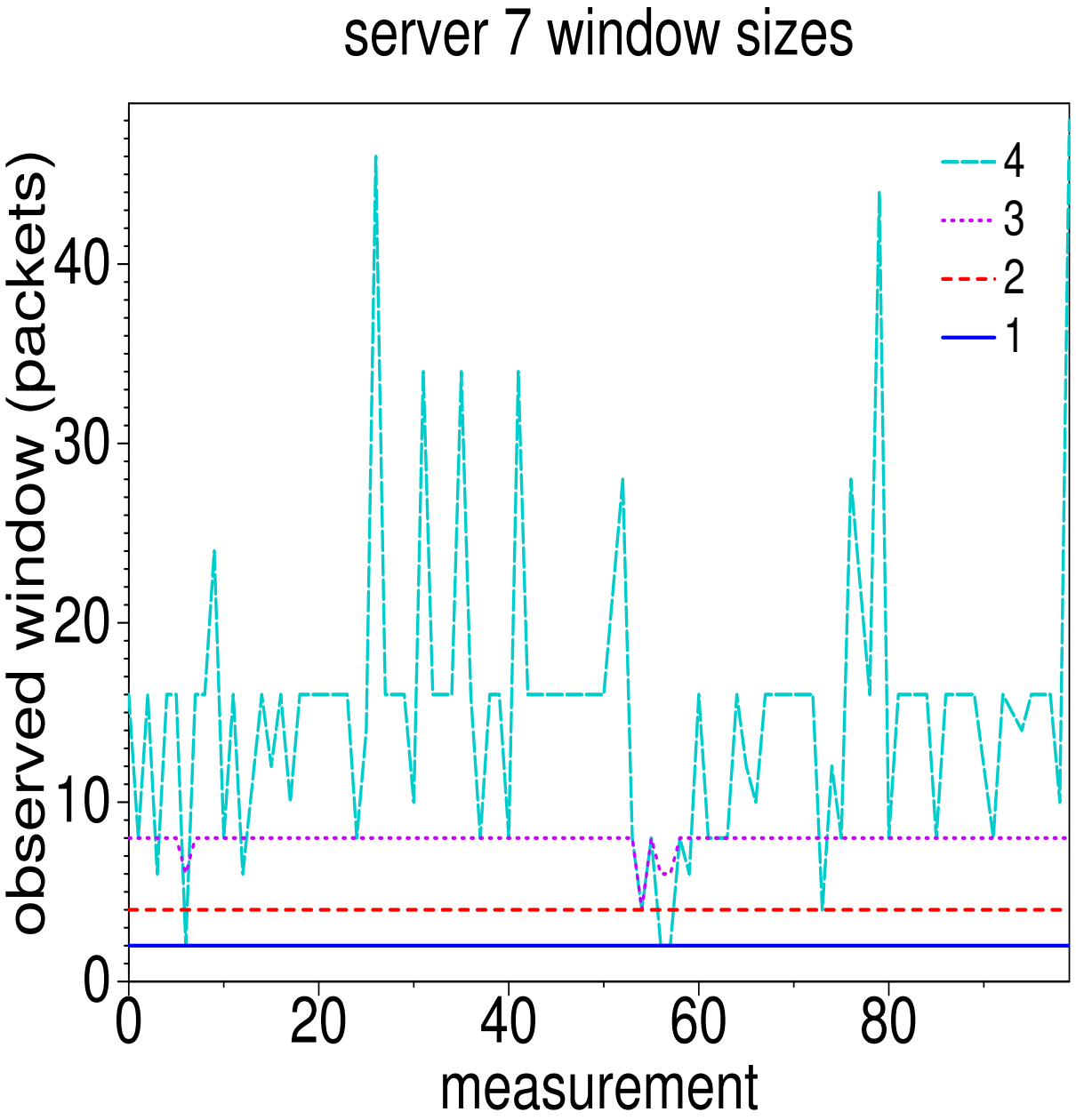
Estimating parameters

- Divide timing chart into rounds.
- Measure window size for each round.
- Pattern match on window sizes:
  - 2, 4, 8, 16, 32 ...
  - 2, 4, 6, 4, 5, 6 ...
  - 2, 4, 6, (long pause) 11, 6 ...
  - 3, 6, 12, 15, 15, 15 ...
  - 3, 6, 12, 51, 17, 63 ...

Measurement

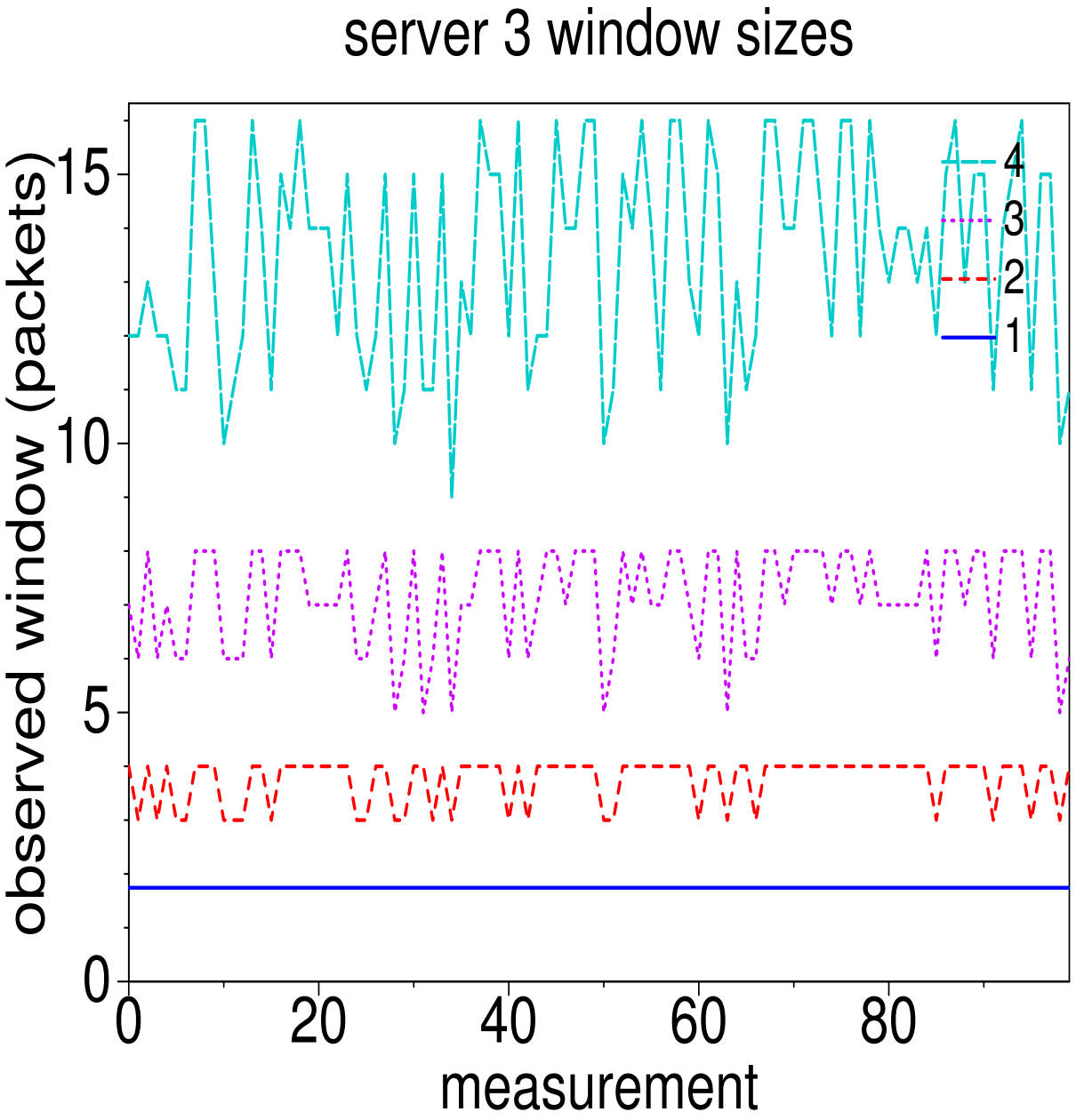
- Measurements:
  - 100,000 byte transfers.
  - 100 transfers, with 100s between.
- HTTP downloads:
  - 2 URLs provided by collaborators.
  - 11 URLs culled from proxy cache logs.
- Diverse network paths:
  - rtt from 7 to 270 ms.
  - bw from 0.350 to 100 Mbps.

Window sizes



- This is the sort of thing I expected.
- Too bad it's the exception.

Window sizes



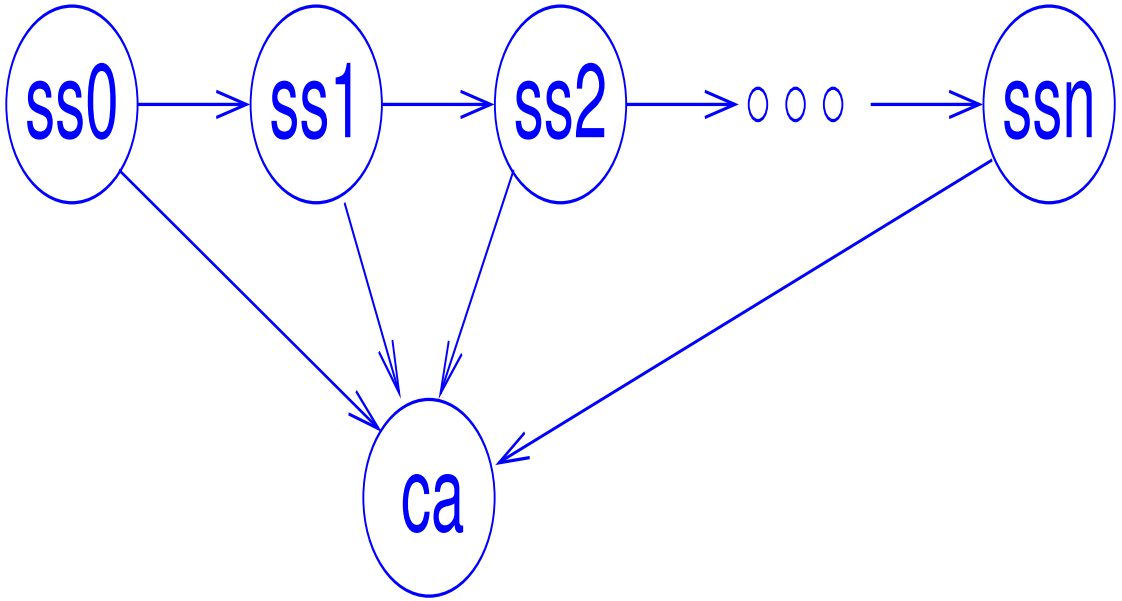
- cw_2 is sometimes 3, sometimes 4.
- cw_{n+1} is $2 \cdot cw_n - m$, where m is $0, 1, 2, ...$
- 10 out of 13 are similar.

Non-deterministic slow start

- Partial explanation in the paper.
- Putting the “empirical” in “empirical model”.
- Models that omit this behavior can’t work for short–medium transfers (2–5 rounds of slow start).

Estimating Parameters

- R: Distribution of rtt.
- W: Distribution of window size.
- T: Distribution of throughput.
- State transition probabilities.



Generating predictions

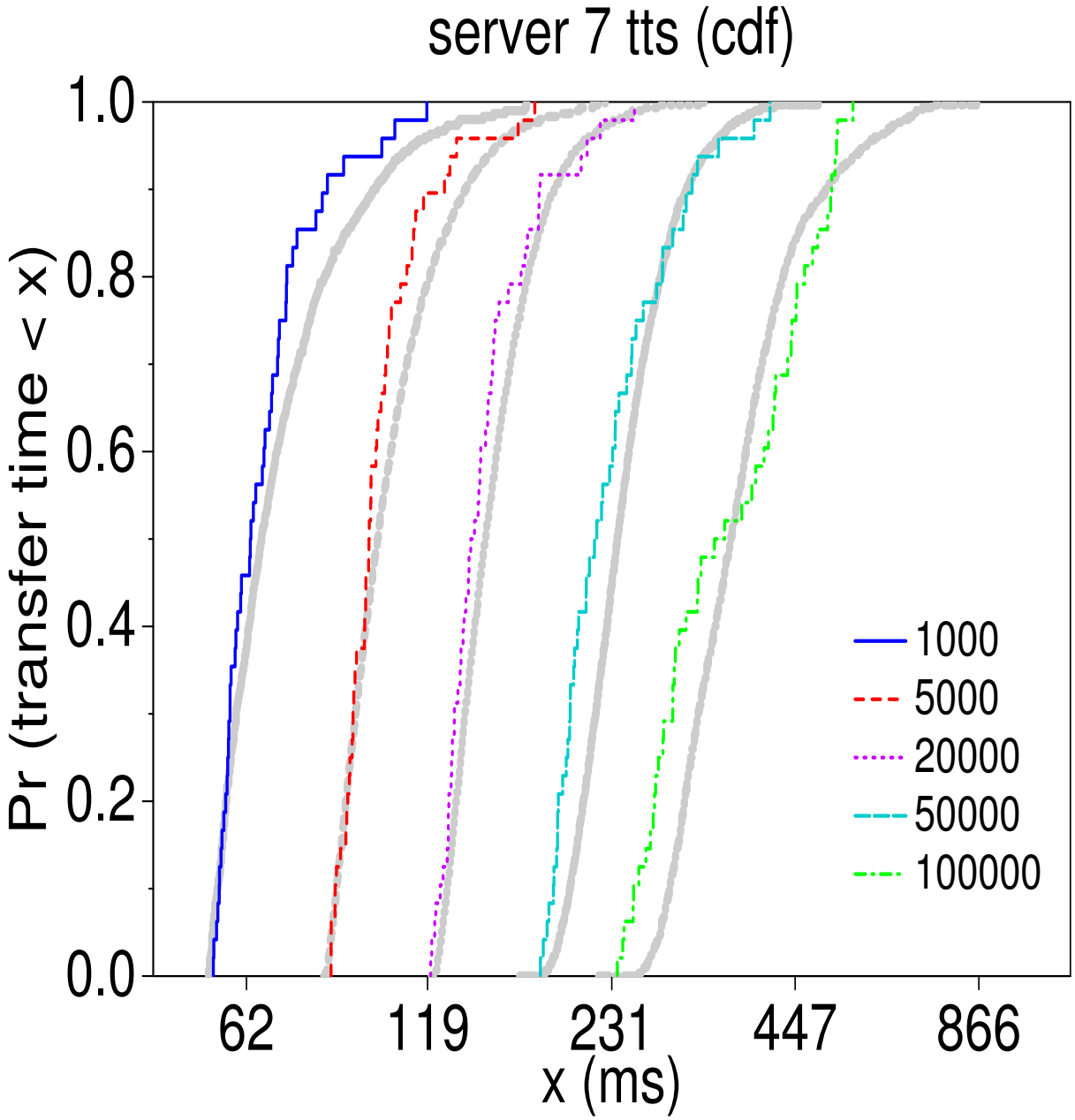
Monte Carlo simulation of finite state model:

1. Start at ss_0.
2. Choose a state transition at random.
3. If in slow start, choose rtt from R and cw from W.
4. If in congestion avoidance, choose $throughput$ from T.
5. Update total time and total data transmitted.
6. If total data $>$ size, return total time.
7. Otherwise go to step 2.

Validation

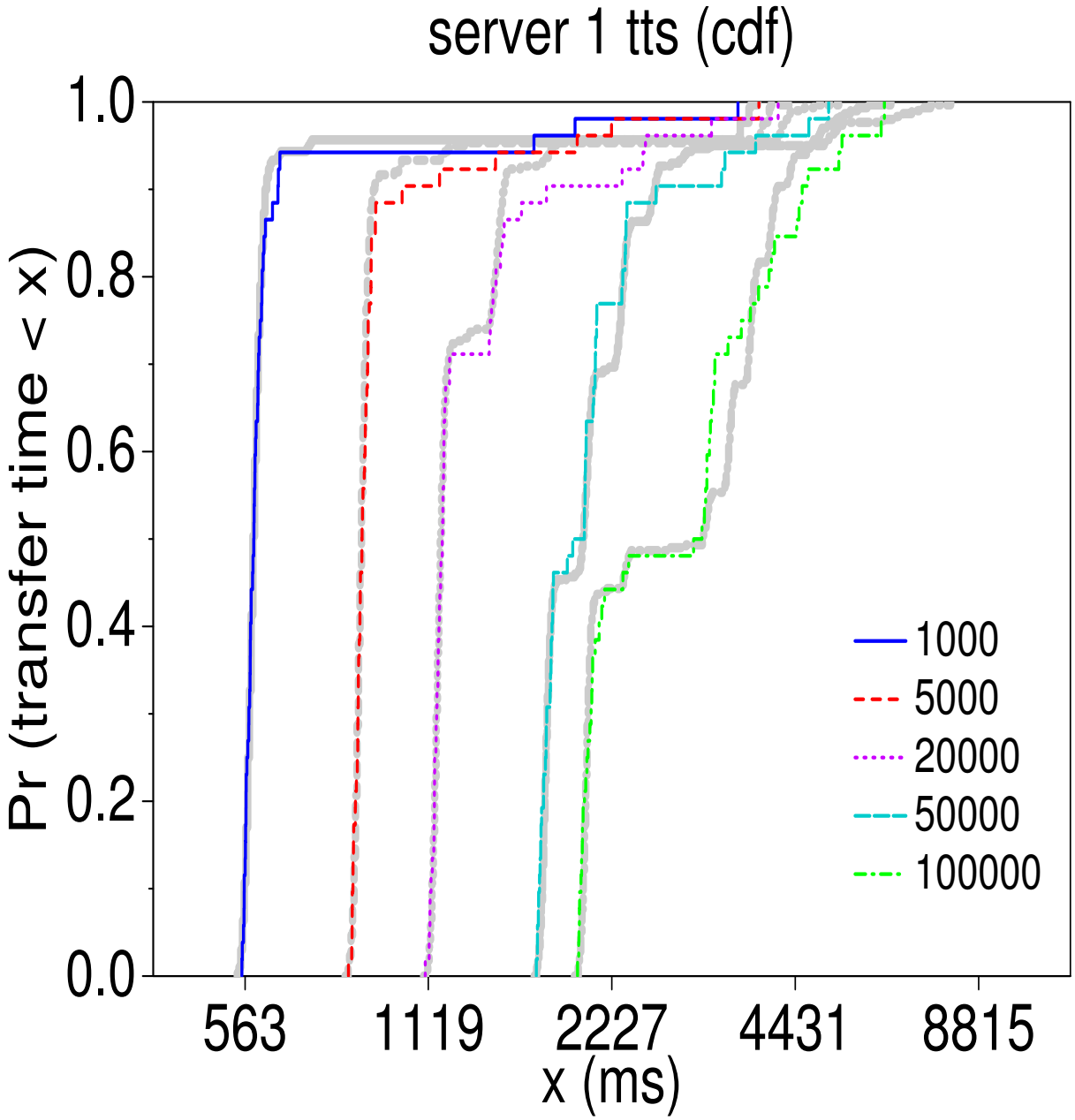
- Randomly partition 2 datasets of 50 measurements.
- Estimate parameters and generate distributions from one subset.
- Compare to actual times from other subset.
- Agreement indicates that the model is sufficiently detailed, and that the estimated parameters are consistent.

Example #1



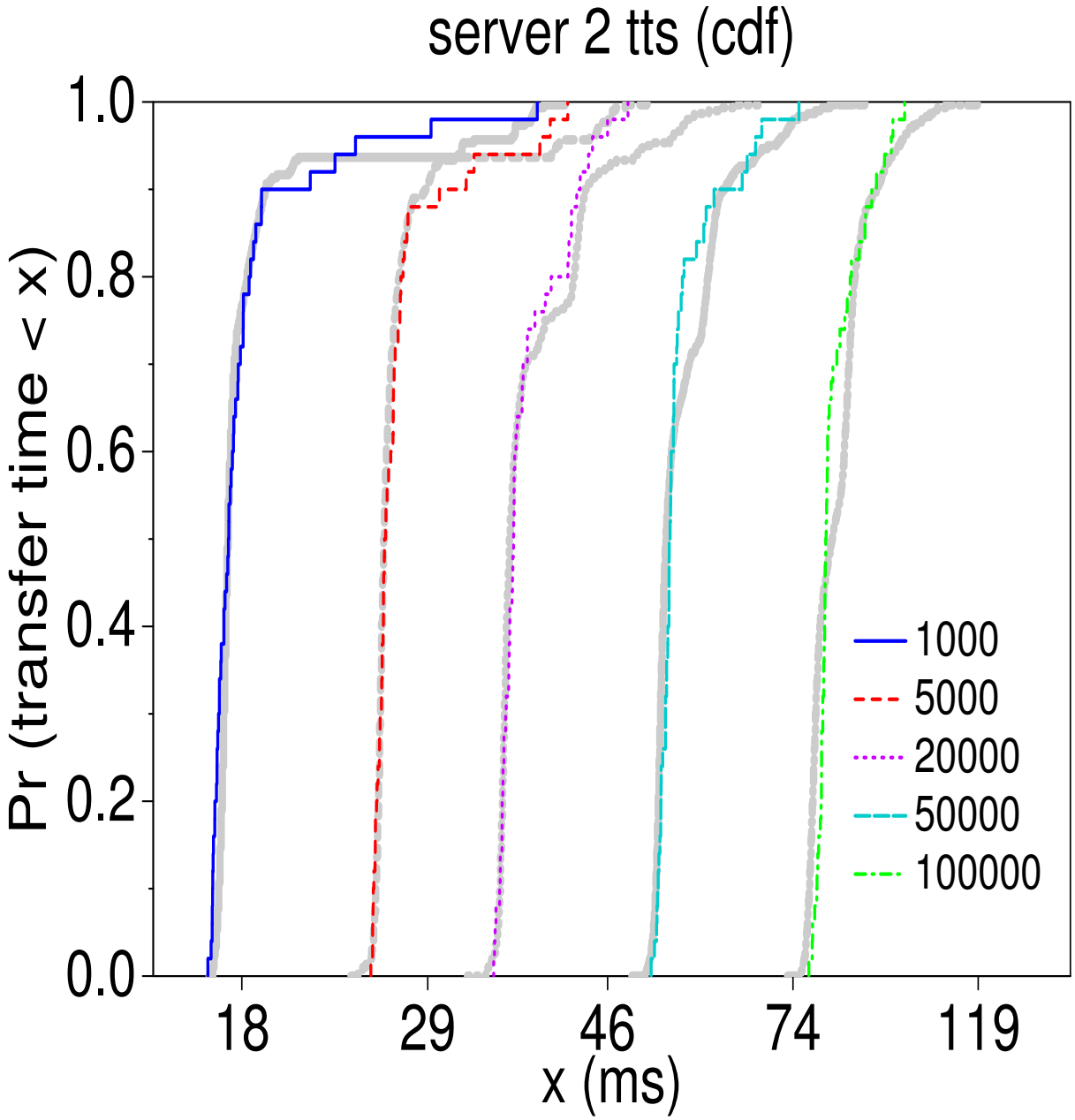
- Deterministic slow start.
- Bandwidth limited.

Example #2



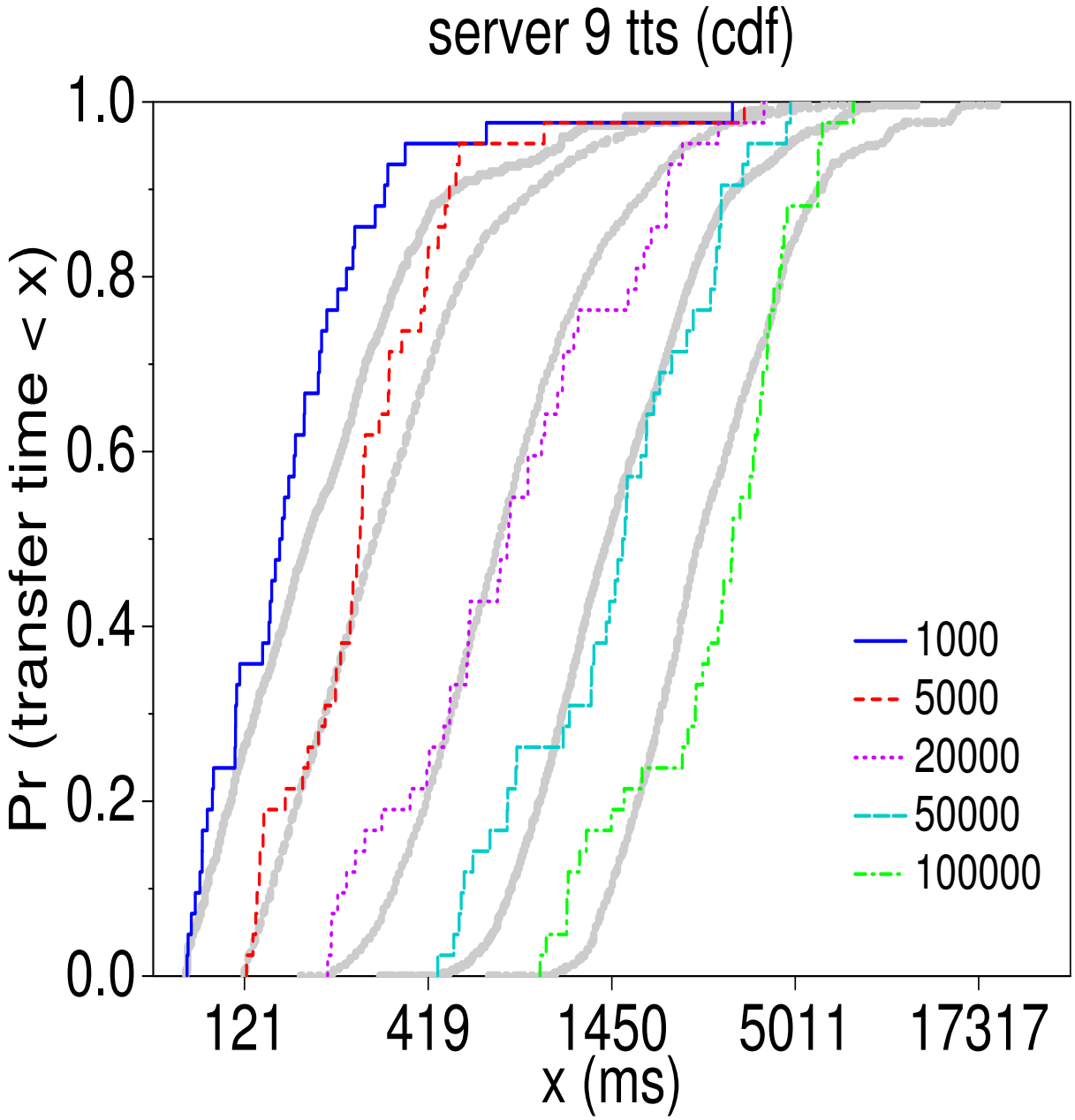
- Nondeterministic slow start $\Rightarrow$ multimodal distributions.
- Modes at multiples of rtt.

Example #3



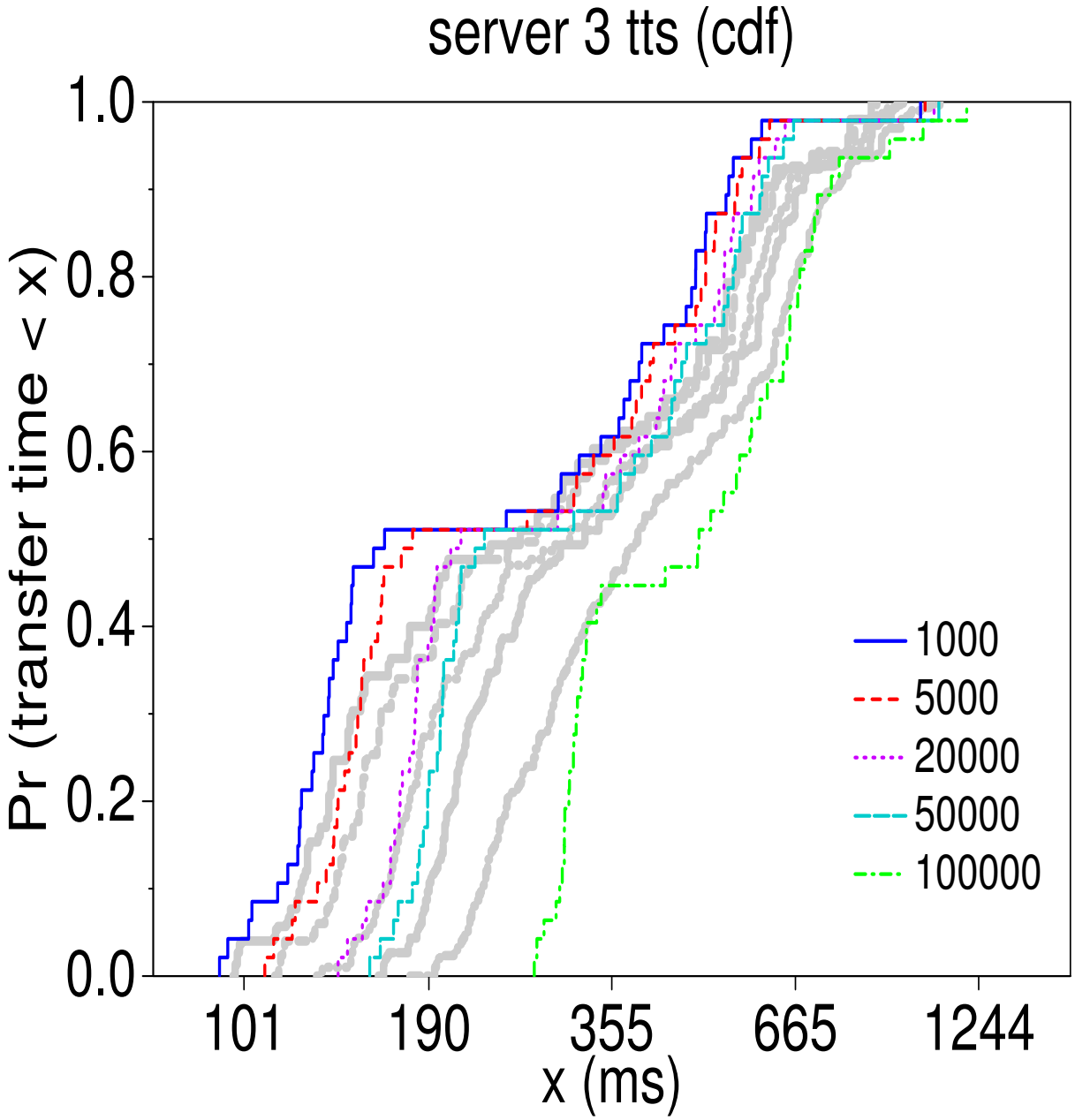
- Buffer-limited.

Example #4



- Congestion limited.
- Underestimating variability?

Evil case #1

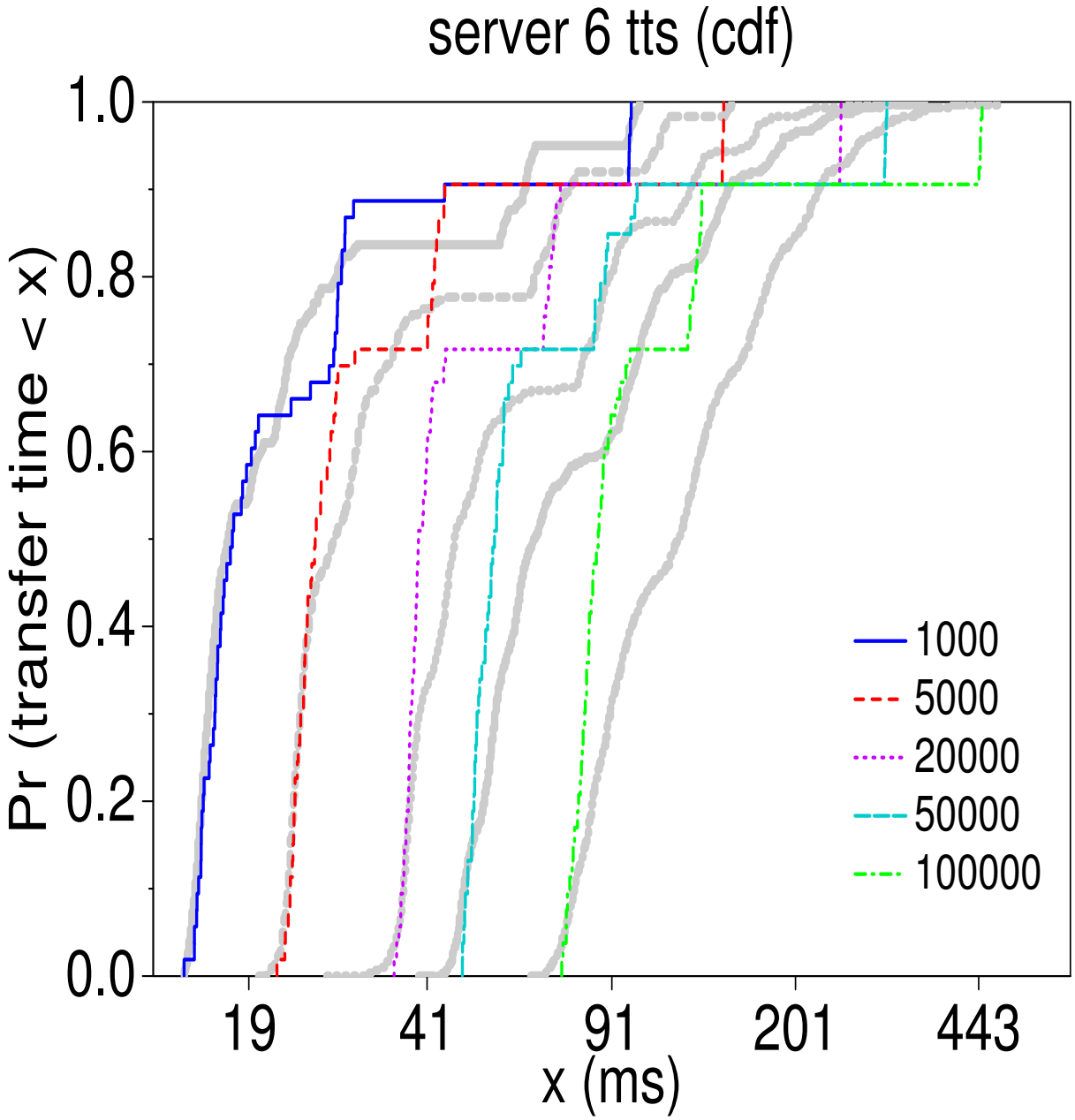


- Up to 50,000 bytes, not bad.
- Anything over 60,000, way off!

Server performance

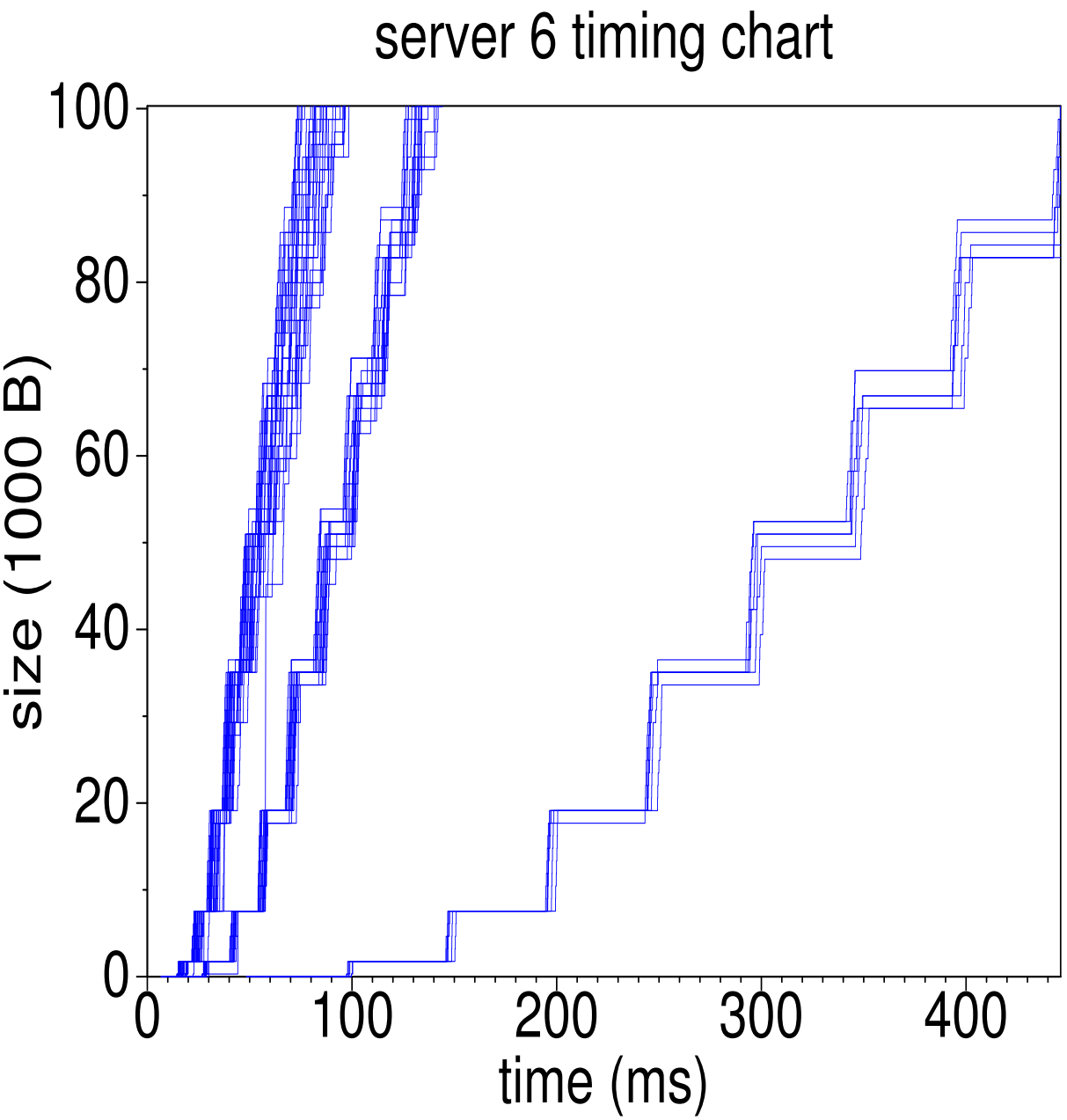
- Server imposes 50 ms delay after 40 packets.
- Model includes processing time in round one, but not afterwards.
- Reminder: real world resists modeling.

Evil case #2



- Actual times are sharply multimodal.
- Model smoothes the modes.

Multiple paths



- Akamai-style content delivery $\Rightarrow$ multimodal rtt.

Headlines

- Hybrid model captures transition from slow start to steady state.
  - Lots of parameters, but easy to estimate.
- Non-deterministic slow start sighted.
  - TCP characteristic or Linux bug?

Ongoing work

- Followup paper:
  “TCP Self-Clocking and Bandwidth Sharing”.
- Network monitoring/predicting tool.
  - Collect active and passive measurements.
  - Maintain database.
  - Generate vizualizations.
  - Answer queries.

Had enough?

- Followup paper and additional data available from

  http://allendowney.com

- Contact me at

  downey@allendowney.com