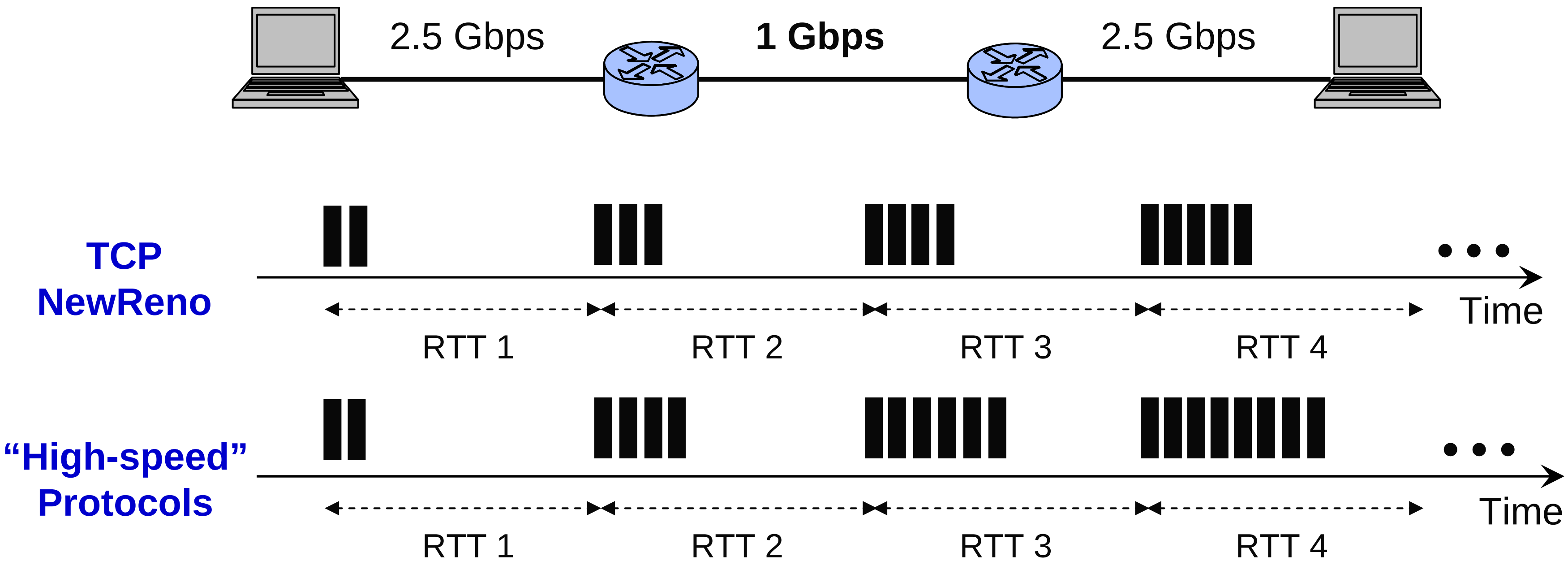
Rethinking the Timescales at Which Congestion-control Operates

Vishnu Konda

Jasleen Kaur

University of North Carolina at Chapel Hill, USA

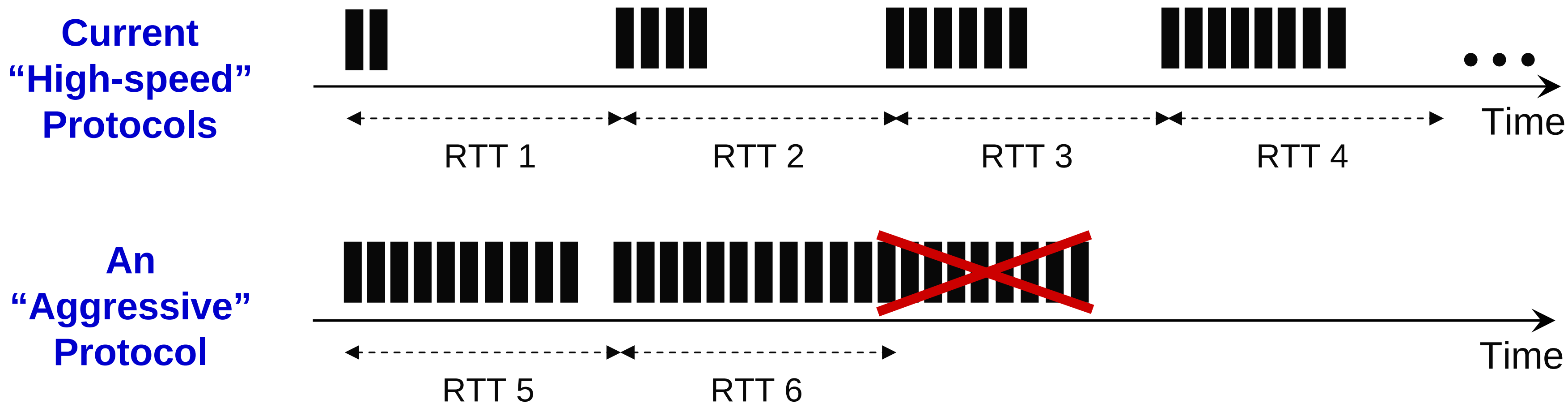
Loong Congestion-control Timescales



TCP NewReno	~ 70,000 RTTs

Even "high-speed" protocols can take hundreds of RTTs for acquiring spare bandwidth!

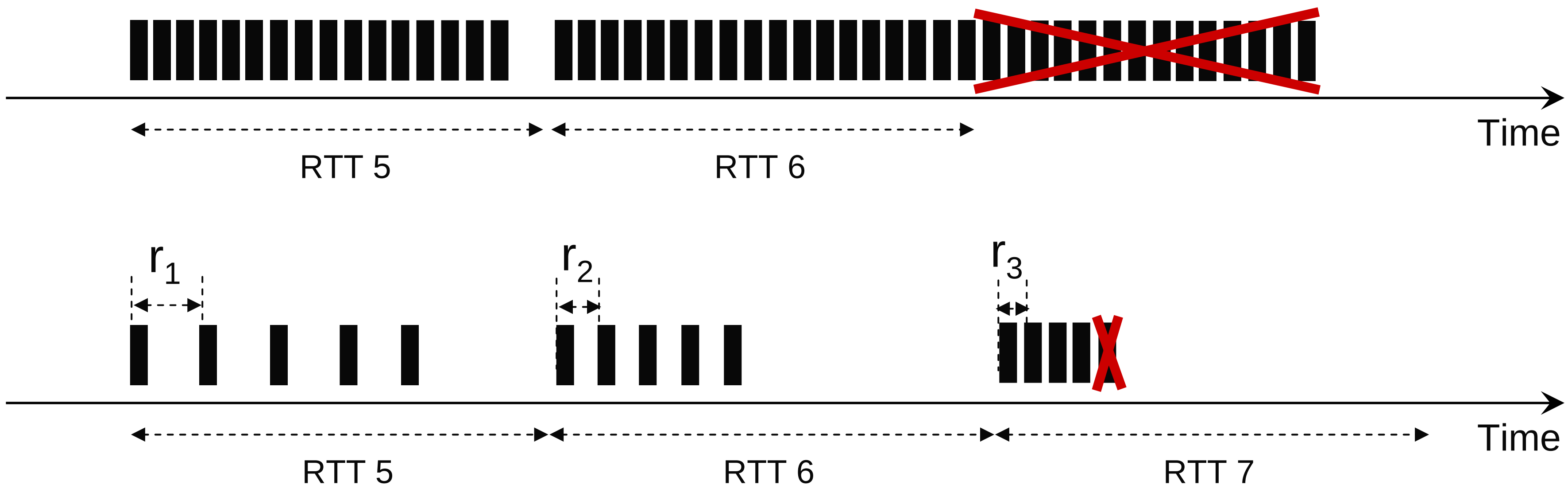
Fear of Overloading the Network



- Even "high-speed" protocols are cautious in increasing their rates
  - nextRate/prevRate $\cong$ 1.1
- A protocol that aggressively increases its sending rate:
  - Can significantly overload router queues
  - Can cause heavy losses in other transfers

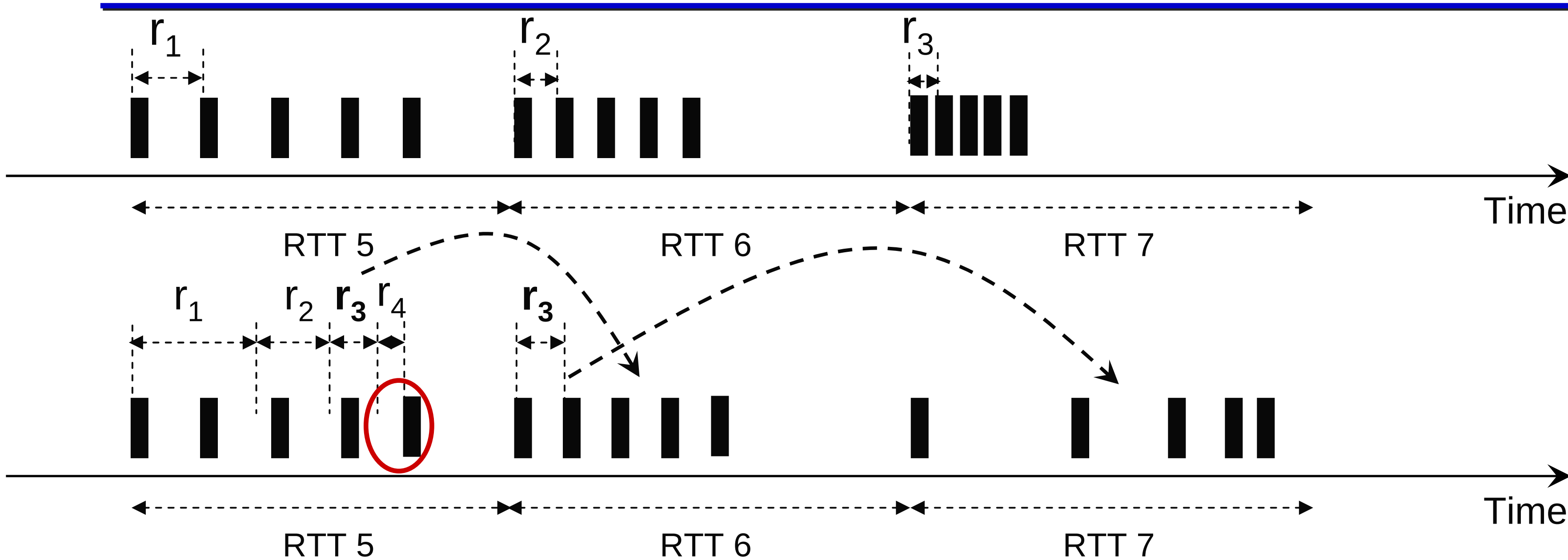
To avoid serious overload, even "high-speed" protocols ensure that next probeRate is not much larger than prevRate !

Key Insight 1: Limit Probe Volume



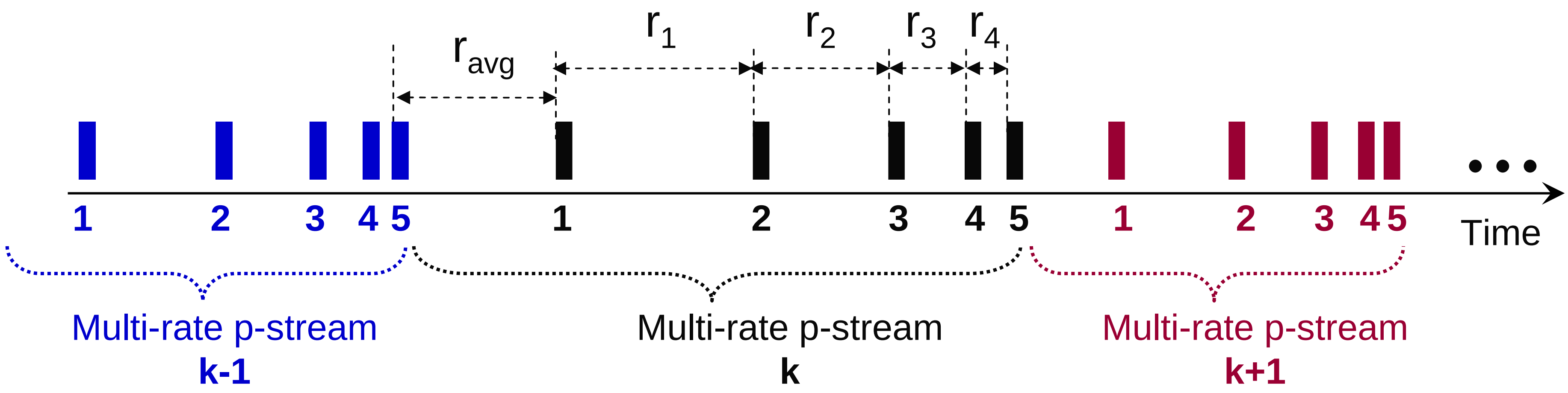
- Limit volume of probes
  - Send *smaller* probe streams *at target rate*
- Rely on *increase in packet gaps* for estimating avail bandwidth
  - Gaps increase if sending rate is larger than available bandwidth
- ★ Can aggressively probe for much larger probeRates
  - Without significantly overloading routers

Key Insight 2: Probe Exponentially in 1 RTT



- Why wait till next RTT for probing higher rate?
  - Probe for an exponentially-wide range of rates in single RTT !
  - Estimate AB based on smallest rate at which gaps start increasing
- Set *average* rate of p-stream equal to most recent AB estimate
  - Avoid persistent overload (at worst, only transient queuing)
- Simultaneously probe for both increase and decrease in AB
  - Highly adaptive !

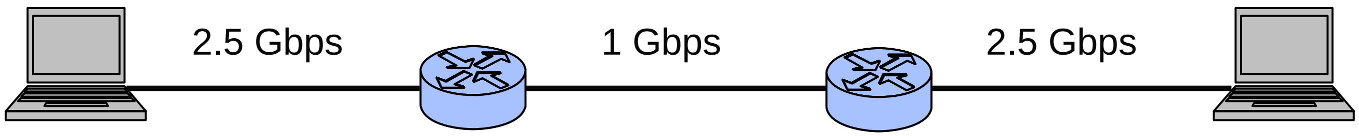
RAPID Congestion Control

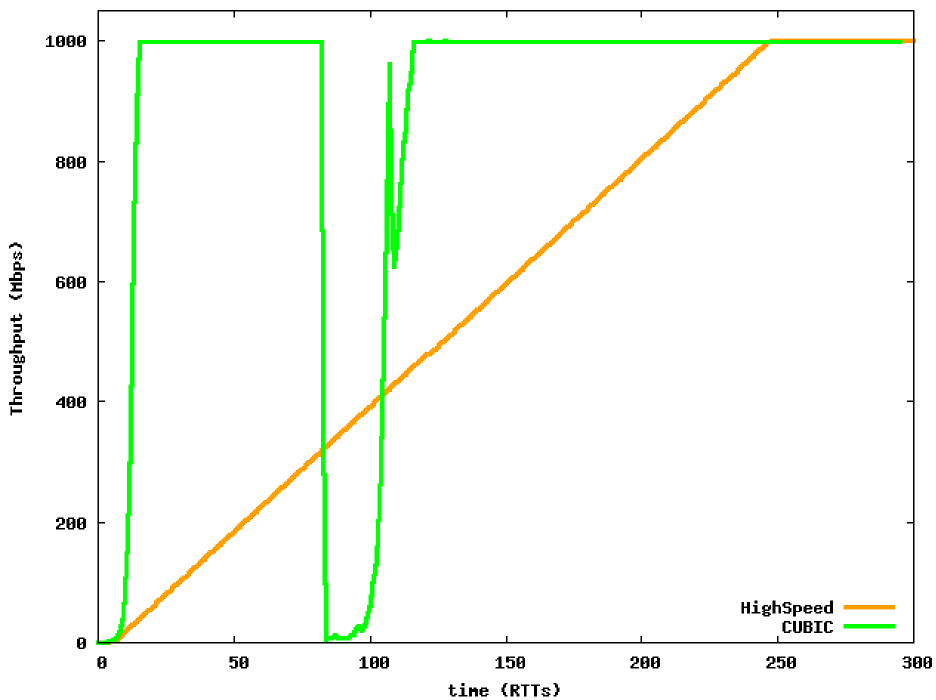


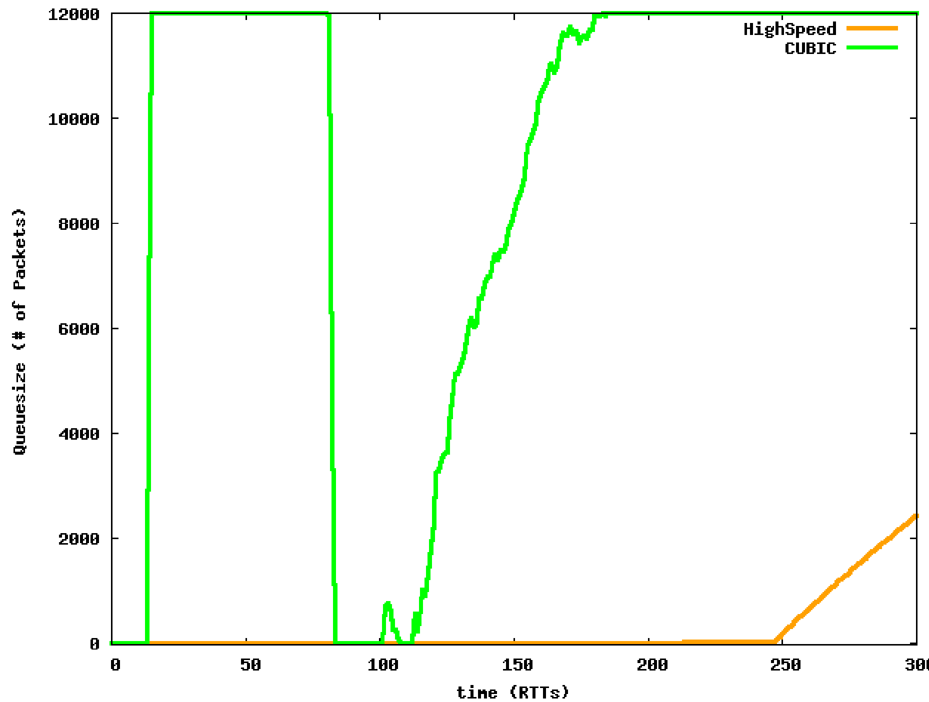
RAPID Feedback Loop:

1. Sender continuously sends multi-rate probe-streams:
   - Controls packet-gaps to probe for an exponentially-wide range
2. Receiver estimates available bandwidth for each p-stream:
   - By observing for increasing trends in inter-packet gaps
   - Sends AB estimate back to sender
3. Sender "acquires" estimated AB:
   - By setting the average rate of next p-stream equal to AB estimate

Speed of Acquiring AB: HighSpeed, CUBIC

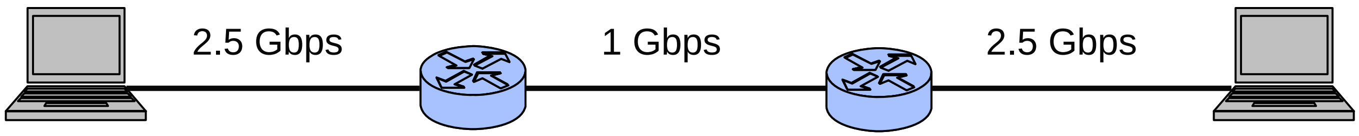


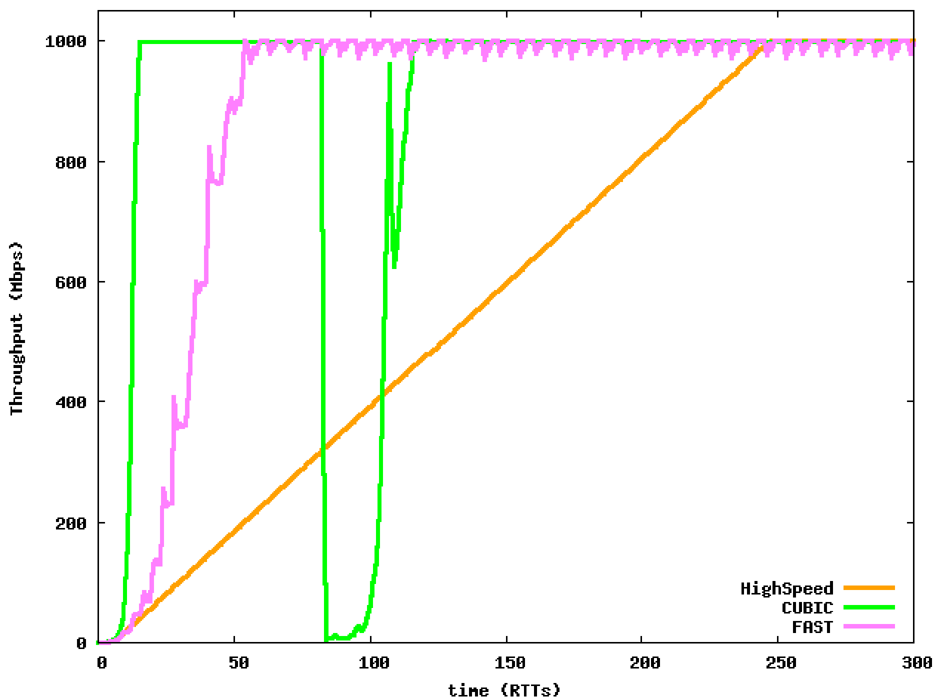


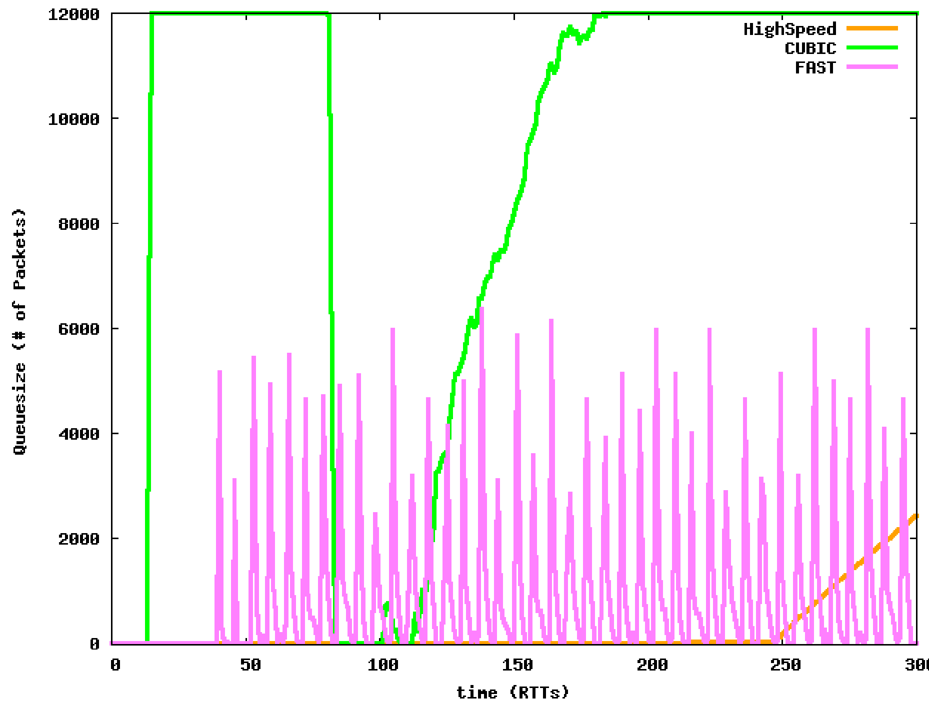


TCP NewReno	~ 70,000 RTTs

Speed of Acquiring AB: FAST TCP

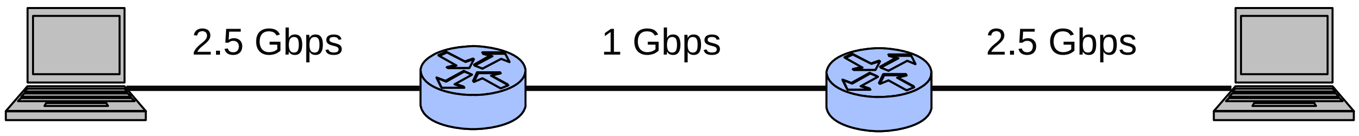


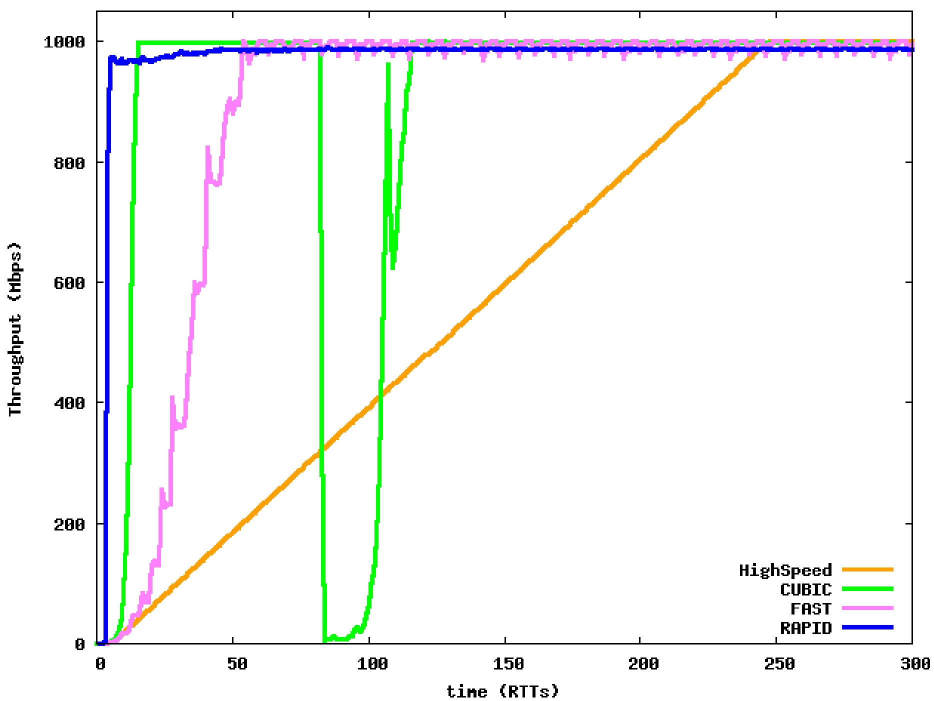


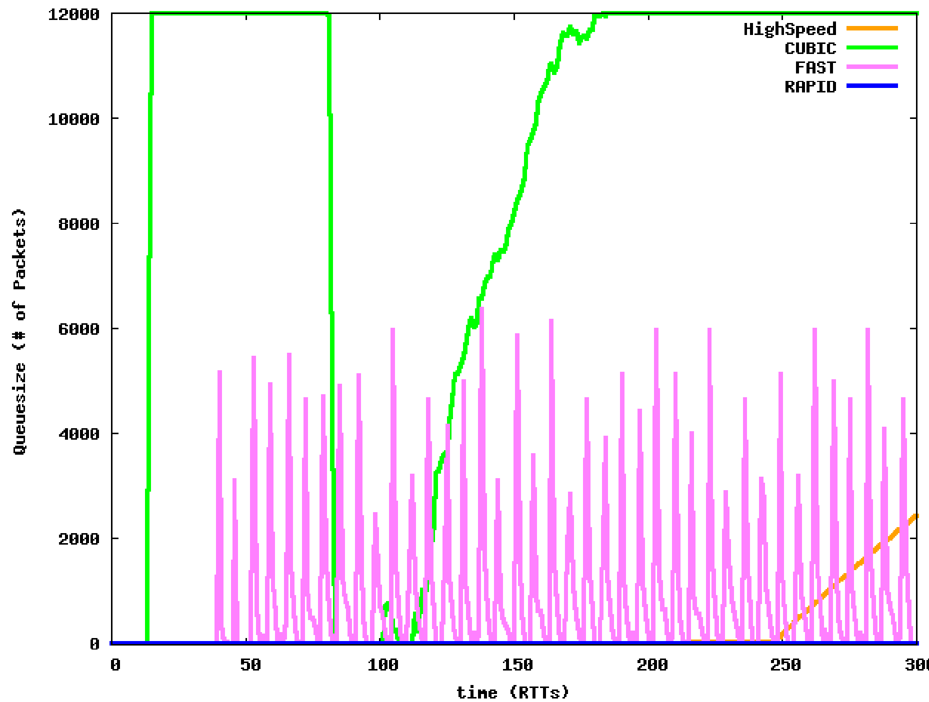


TCP NewReno	~ 70,000 RTTs
HighSpeed TCP	~ 250 RTTs
CUBIC	~ 110 RTTs

Speed of Acquiring AB: RAPID

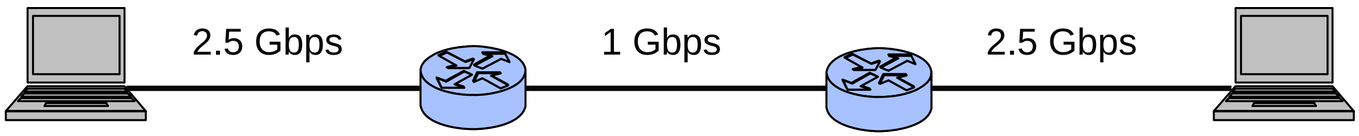


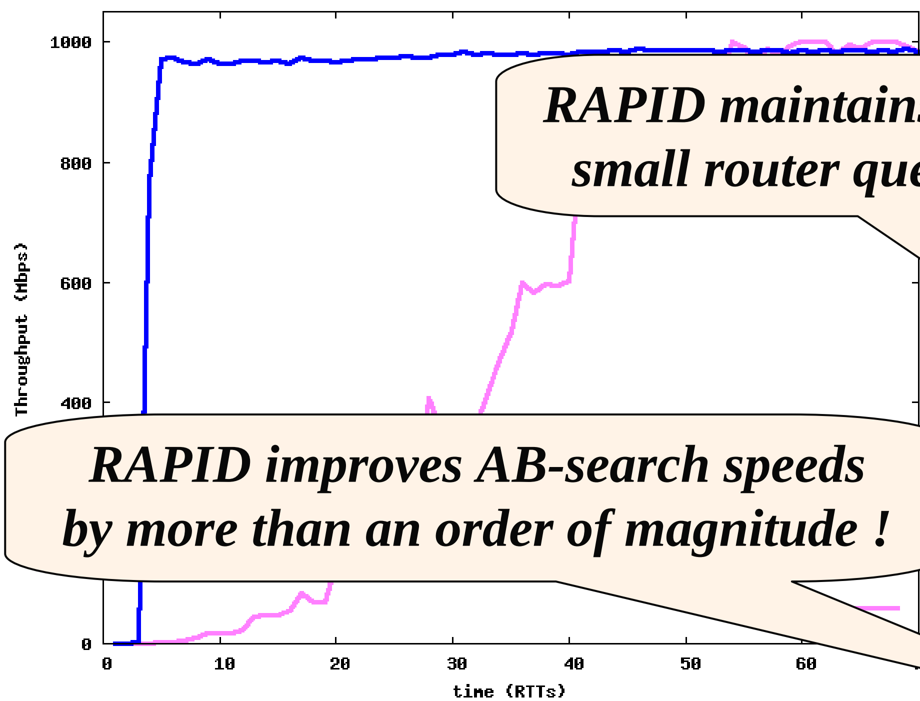


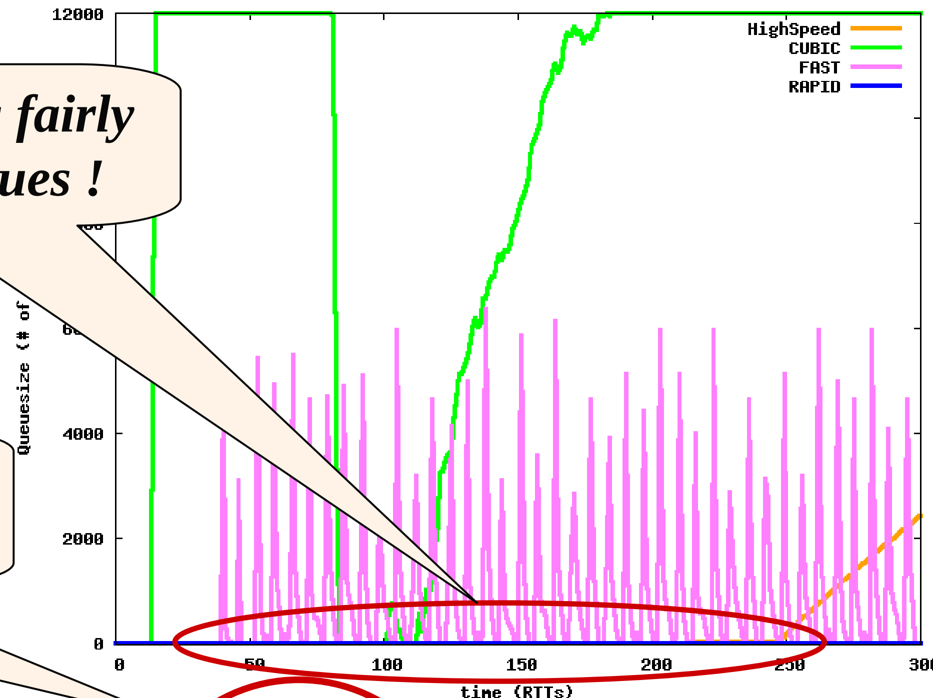


TCP NewReno	~ 70,000 RTTs
HighSpeed TCP	~ 250 RTTs
CUBIC	~ 110 RTTs
FAST TCP	~ 55 RTTs

Speed of Acquiring AB: RAPID





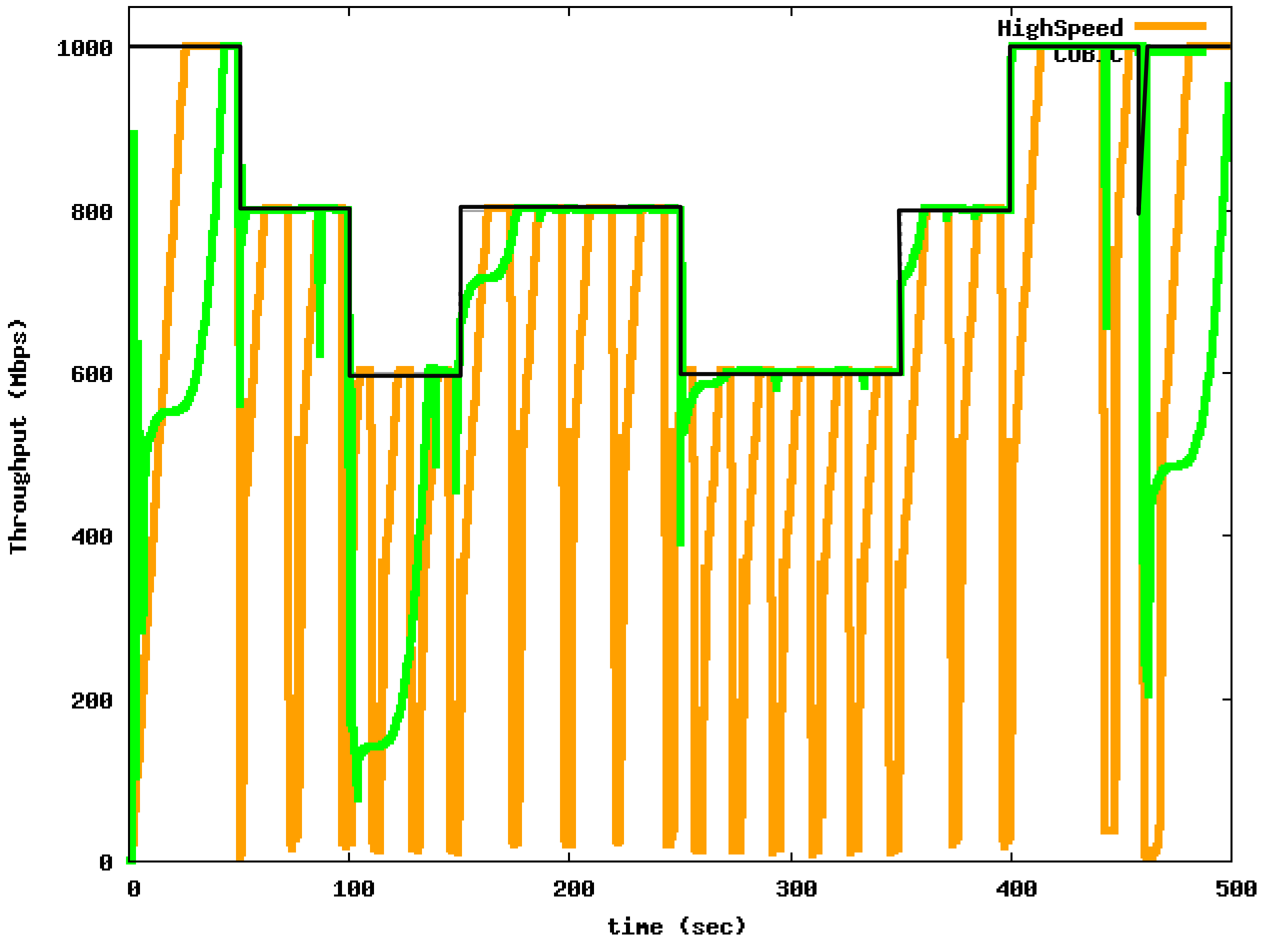


RAPID maintains fairly small router queues !

RAPID improves AB-search speeds by more than an order of magnitude !

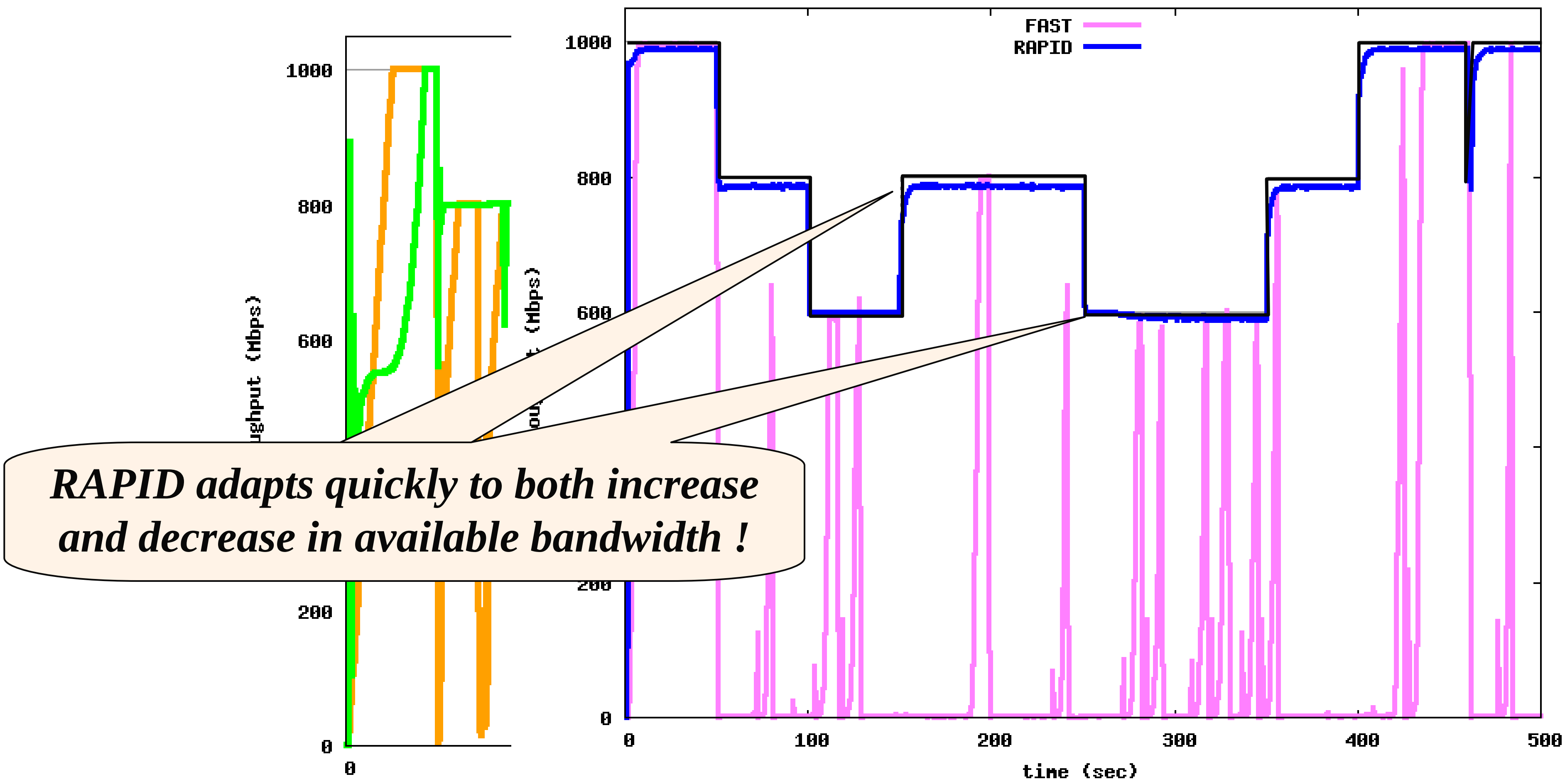
TCP NewReno	~ 70,000 RTTs
HighSpeed TCP	~ 250 RTTs
CUBIC	~ 110 RTTs
FAST TCP	~ 55 RTTs
RAPID	~ 4 RTTs

Adaptivity to Dynamic Bandwidth



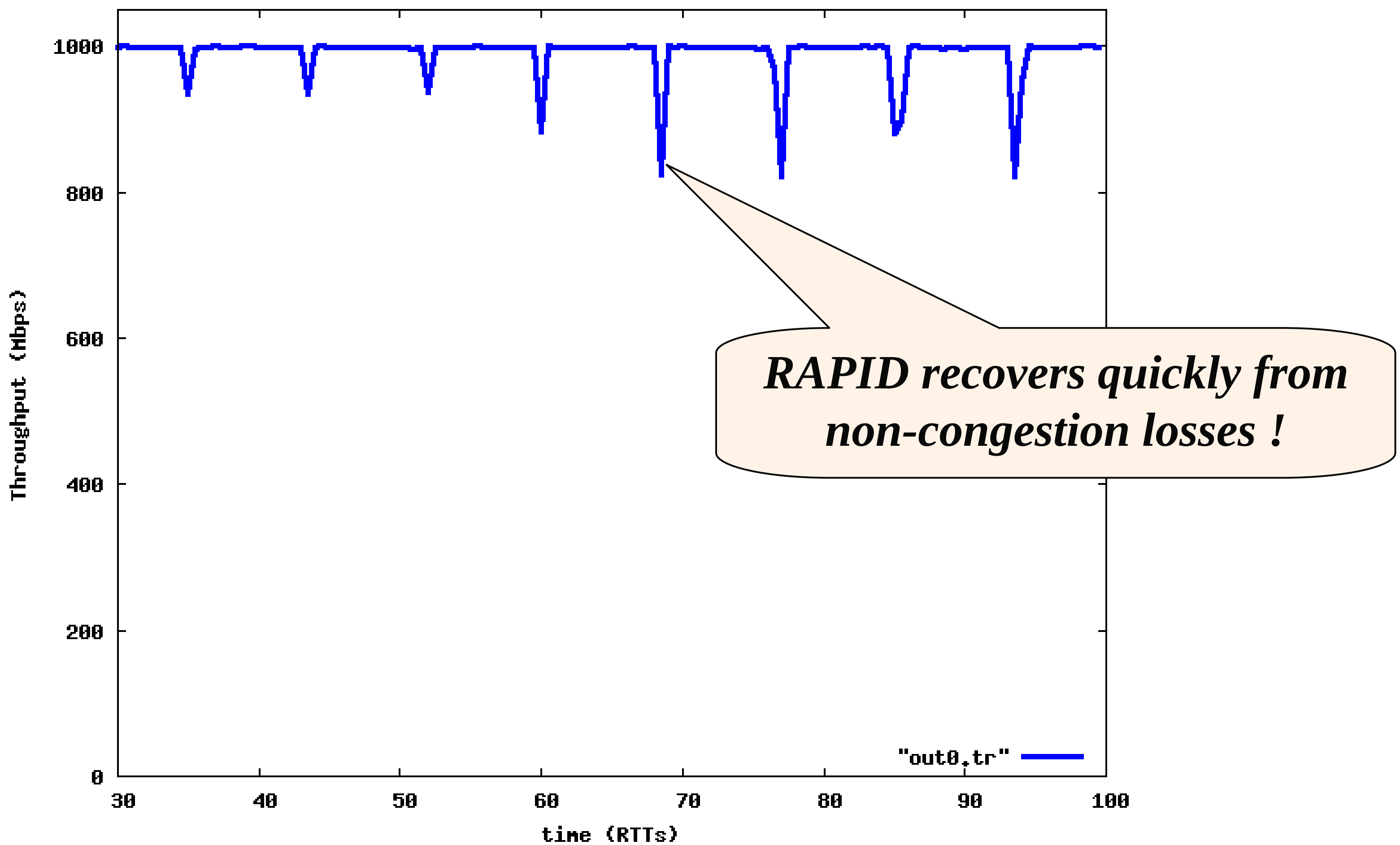
- Loss-based CUBIC and HighSpeed keep queues full
  - *Suffer heavy losses when AB decreases*

Adaptivity to Dynamic Bandwidth



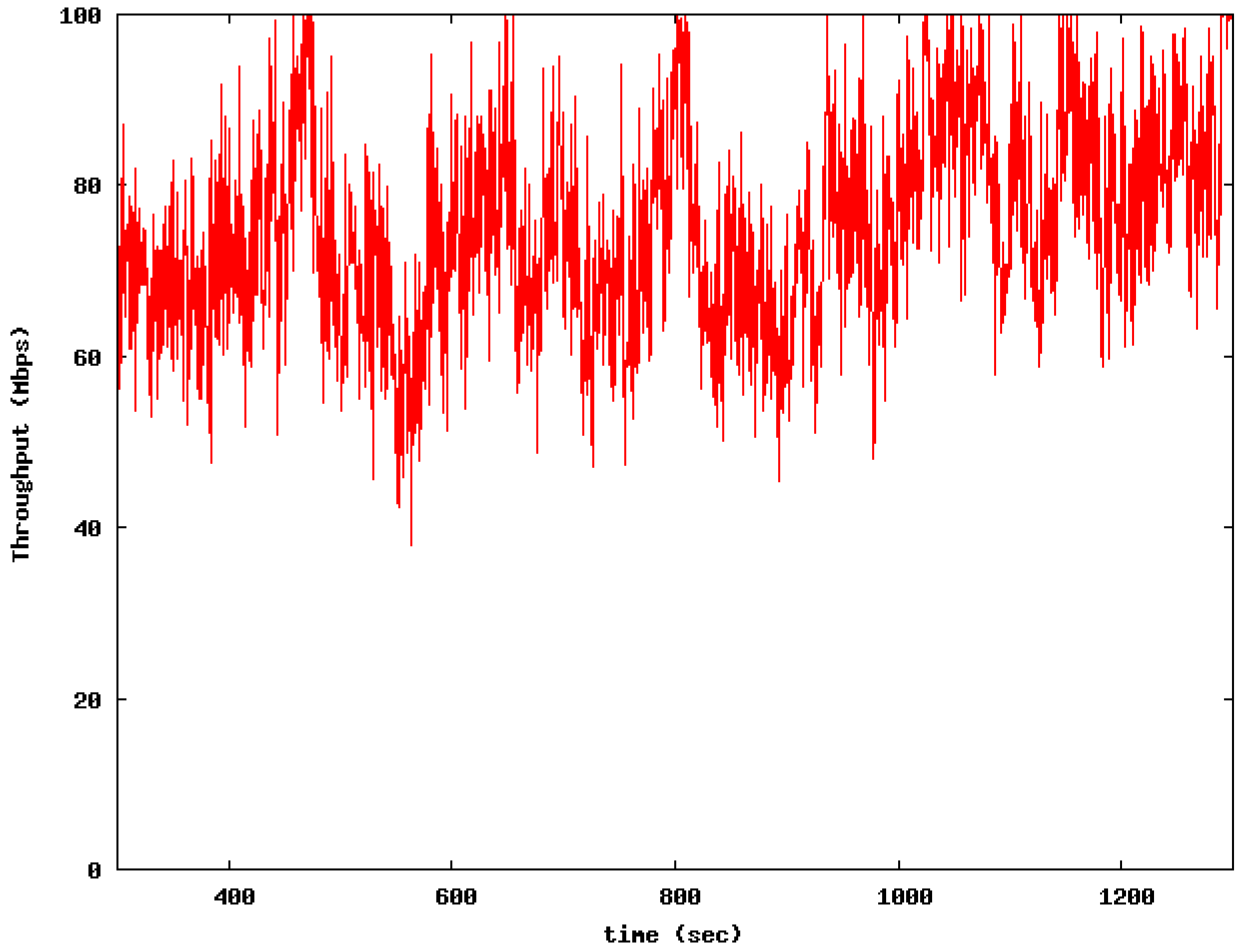
- FAST is not able to recover well from losses
- RAPID avoids losses when available bandwidth decreases
  - Quickly acquires additional spare bandwidth

Recovering From Error-based Losses



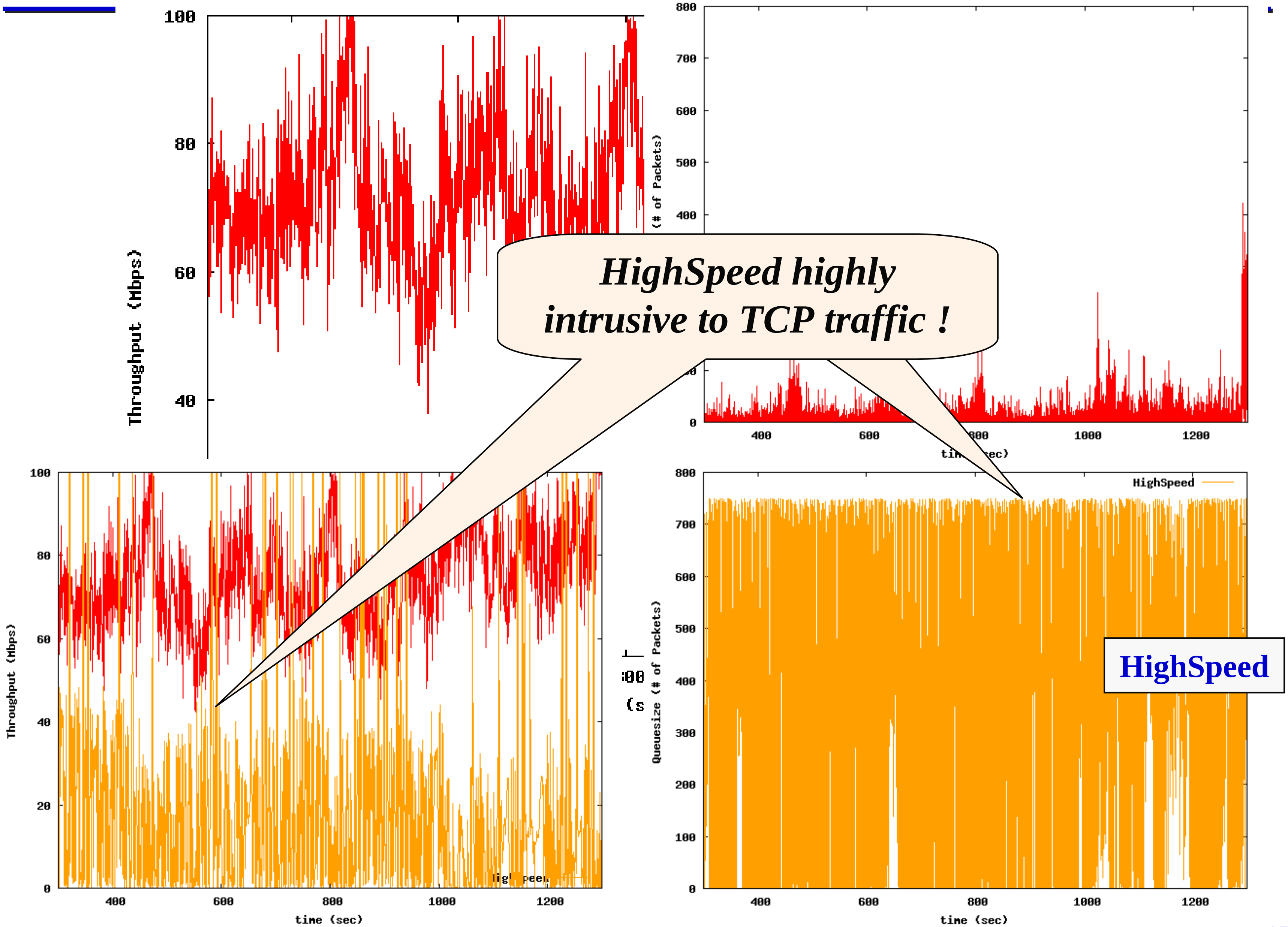
- RAPID subjected to 10^{-6} packet loss rate

Friendliness to Regular TCP Traffic

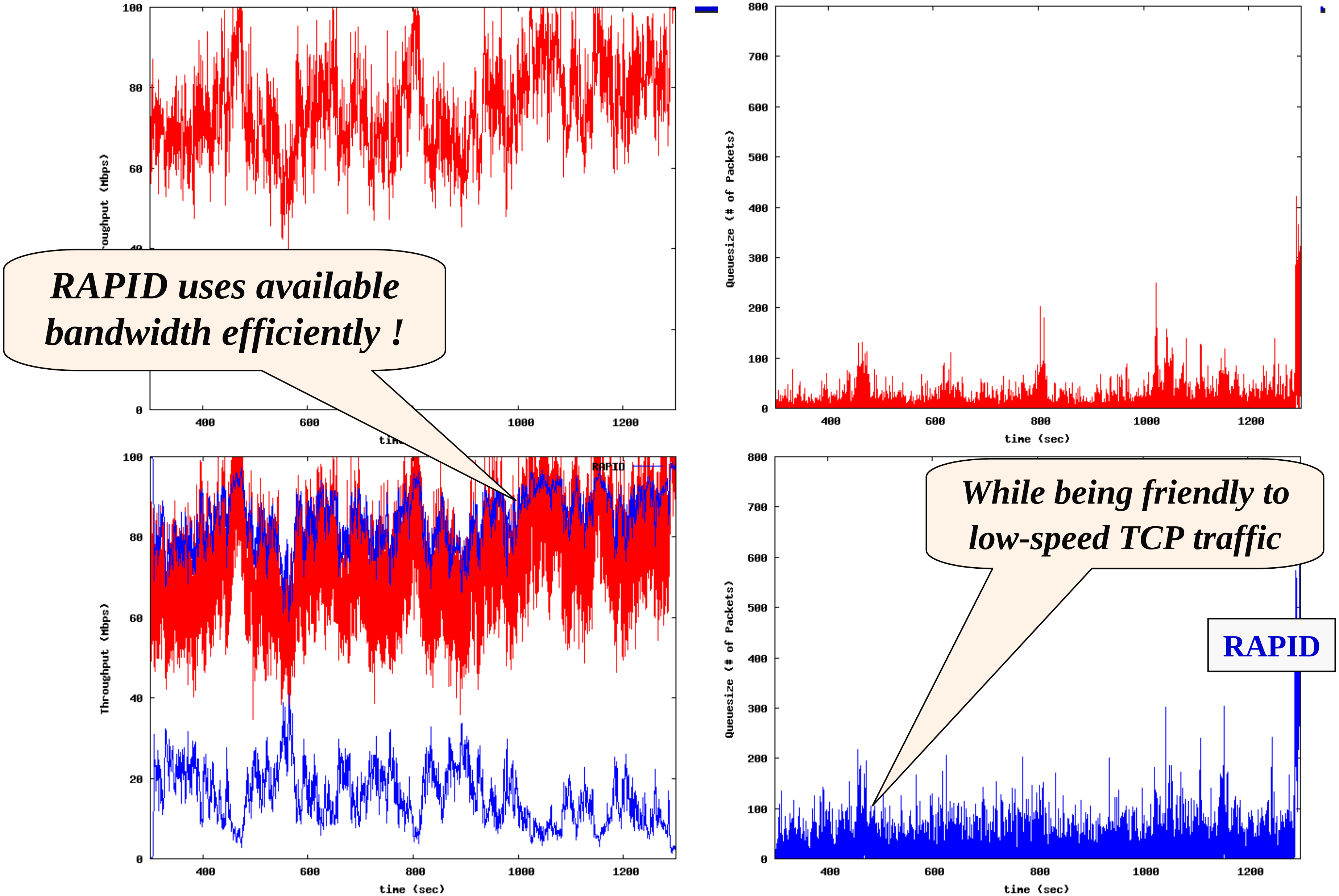


- Empirically-derived traffic mix
  - Diverse: mice and elephants, heterogeneous RTTs
  - Bursty traffic: dynamic AB

Friendliness to TCP Traffic: HighSpeed



Friendliness to Regular TCP Traffic: RAPID



Summary: RAPID Congestion Control

- ★ Reduces AB-search times by orders of magnitude
  - ➢ Probes for an exponentially-wide range of rate within an RTT
- ★ Maintains only small and transient queues
  - ➢ Maintains an average sending rate that can be supported by network
- ★ Is very efficient in dynamic bandwidth environments
  - ➢ Probes for both increase and decrease in available bandwidth
- ★ Is friendly to low-speed TCP traffic aggregates
  - ➢ Uses a delay-based strategy for estimating available bandwidth

- ➲ Main issue: implementability at high speeds
  - ➢ Accurate and fine-grained time-stamping
    - – 1.2❍s on 10 Gbps links
  - ➢ Sensitivity of p-streams to fine-timescale traffic burstiness
    - – p-streams are only a few milliseconds long