I AM THE INTERNET
(AND SO CAN YOU!)

But we're both slower than we could be...

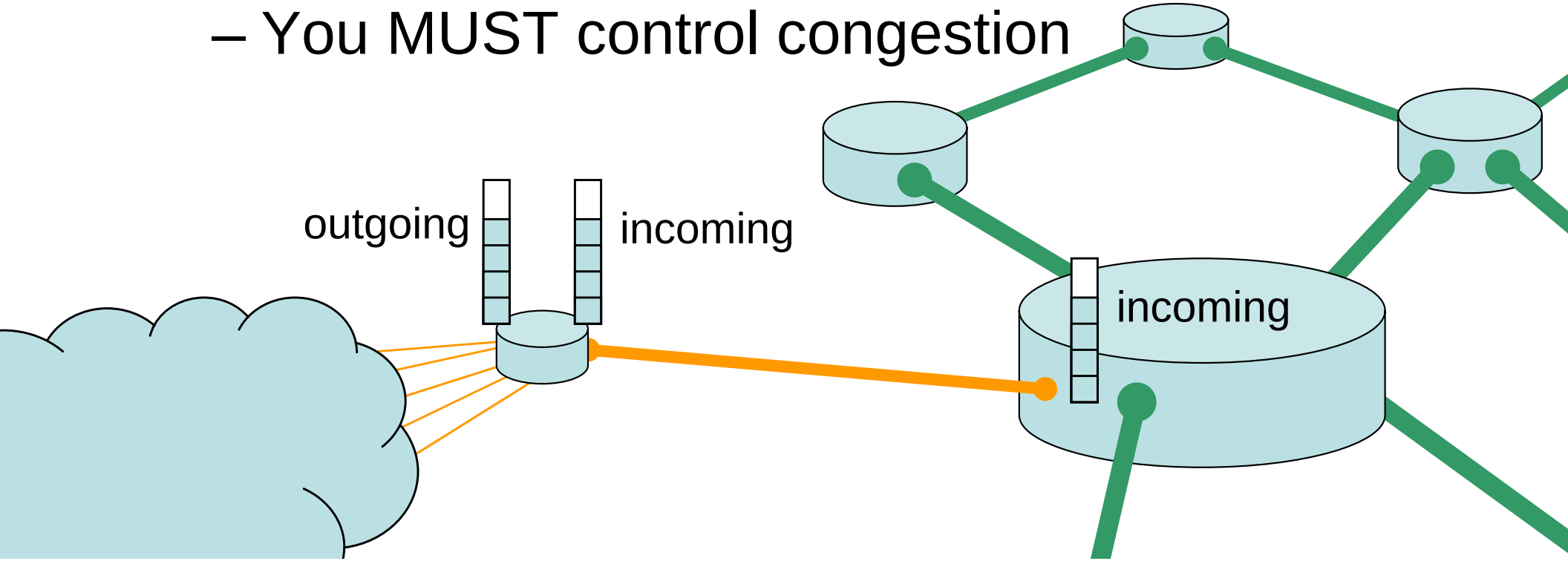
Ryan Witt

California Institute of Technology

and now fastsoft

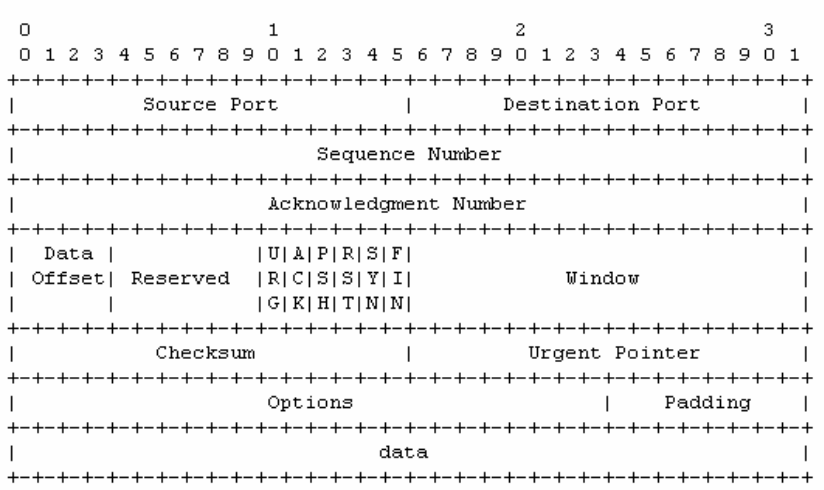
Series of Tubes

- Carriers overprovision core
- Traffic bottlenecks mostly at ingress/egress
- Bottlenecks = queues = drops = ☹
- You MUST control congestion



How do we control congestion?

- **TCP** (Transmission Control Protocol)
  - Sits on top of IP

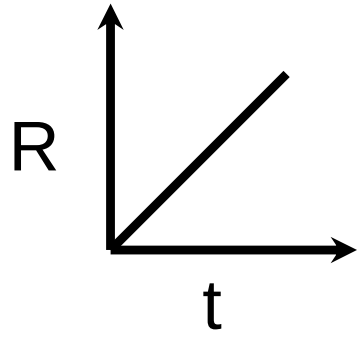


  - Ensures Delivery
    - Handshake
    - Sequence numbers
    - Retransmissions
  - Carries out Congestion Control
    - Backoff on loss

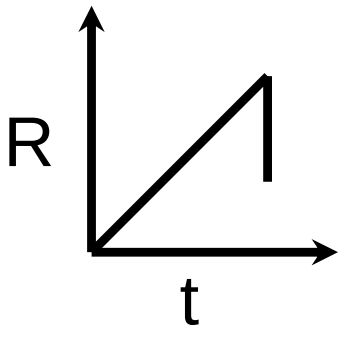
Looking closer @ TCP

- (congestion) Controlling Idea:
  - Conservatively increase send rate until loss
  - Liberally back-off when loss occurs

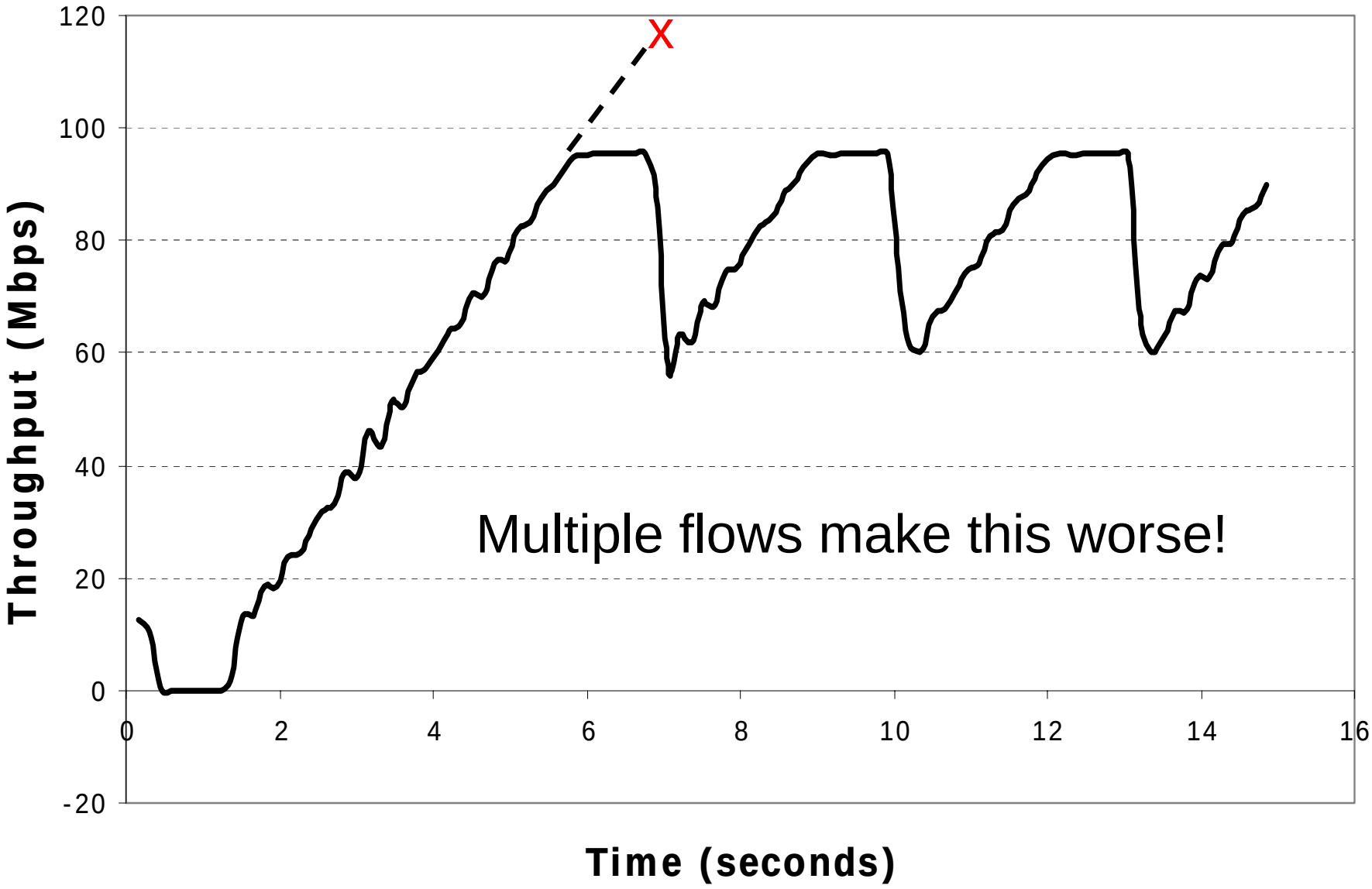
- AI (Additive Increase)
  - R(t) = K + R(t-1)



- MD (Multiplicative Decrease)
  - R(t) = C * R(t-1)
  - Typically C = 0.5



Single Flow Example



Problems with TCP

- AIMD doesn’t scale
  - AI too conservative for high bandwidth x delay
  - AI too liberal when pipe is full
  - MD too liberal, long recovery
- Router Queues fill up
  - Makes for unnecessary delay
  - Introduces congestion based loss
  - AQM (RED, others) primary weapon so far

Practical

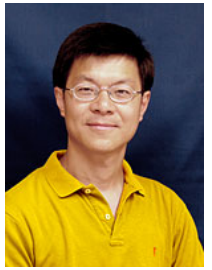
Problems with TCP

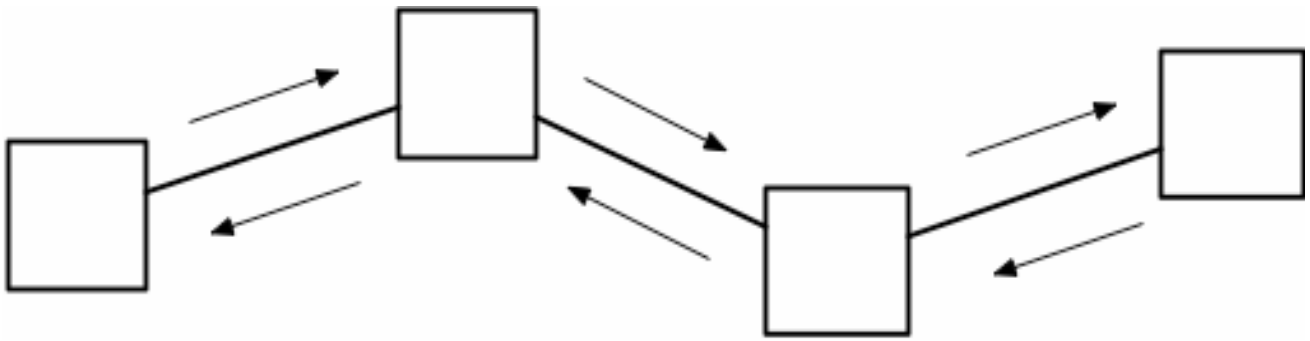
- Centralization difficult
  - Affects content delivery, business communication, replication
  - Rise of the CDN
  - Rise of Riverbed
- Data collaboration difficult
  - Affects sience, film production, manufacturing
  - Try downloading direct from APAC or Europe!
  - Rise of download optimizers
- Real time interaction difficult
  - Affects voice, video, ssh, rdp, control lines
  - Ever sent a large attachment or upload video during ssh/rdp?
  - Rise of QOS

Solution?

- Limitation is boolean congestion signal
  - What if we use something better?
  - Many attempts (HSTCP, STCP, BIC, etc.)

- FAST TCP -- Netlab
  - Steven Low -- Control Theory perspective
  - Use Round Trip Time (RTT) to estimate delay



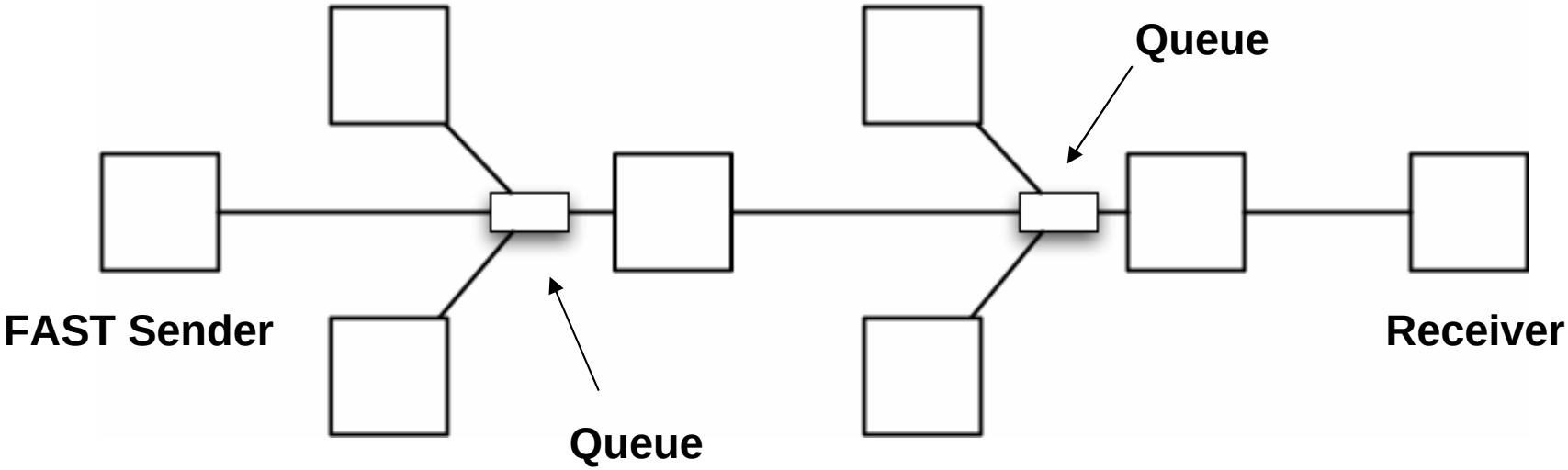


Looking closer @ FAST

– R(t) = baseRTT / RTT * R(t-1) + α

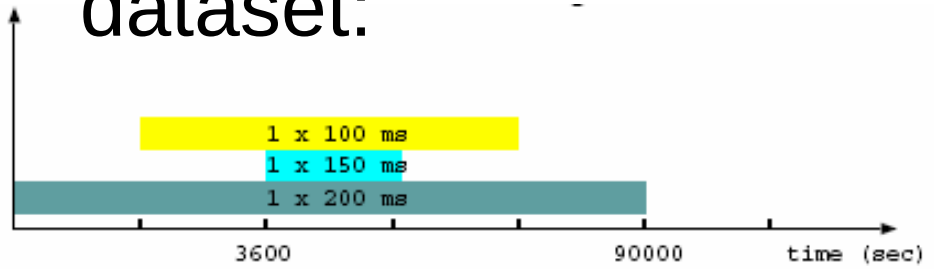
Minimum Observed RTT

Current RTT

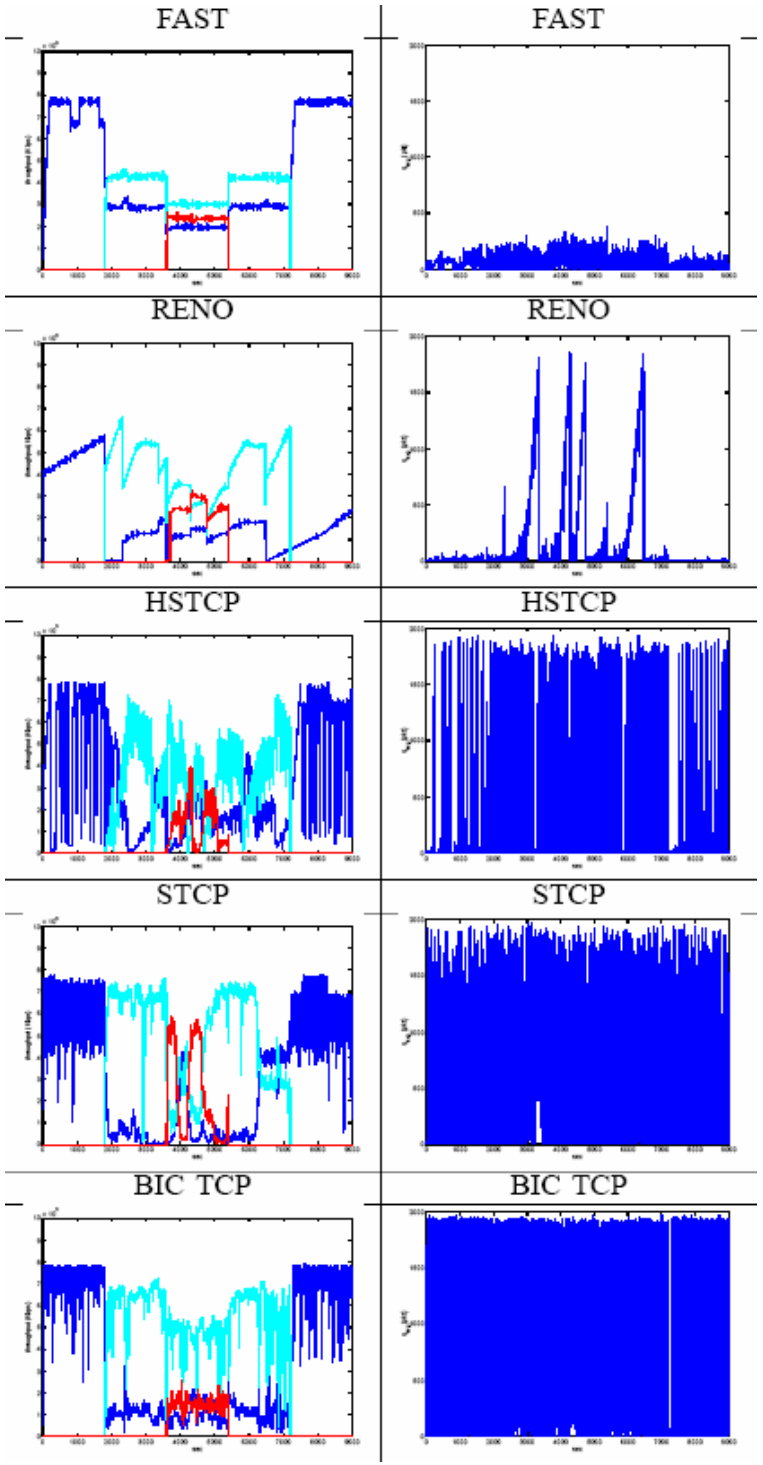


TCP bake off

- Dynamic, three flow dataset:



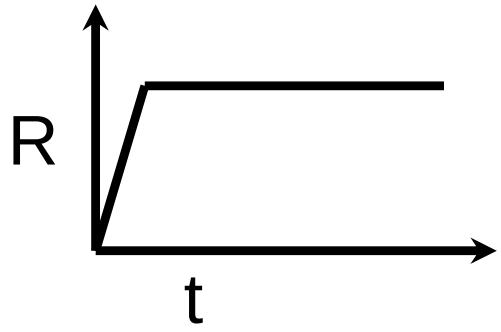
- Thorough models beat out ad-hoc protocol design!



Wei, Jin, Low, Hegde, 07

FAST Features

- Scalable to arbitrary network parameters
  - Size, Bandwidth, Delay
- Doesn’t require modifying existing architecture
- Full utilization of Bandwidth



- Drastically reduced delay if it controls a bottleneck

FAST Achievements

- With SLAC, CERN, Fermilab, and others…
  - Broke Internet2 Land Speed record (IPv4)
  - Quadrupled world record at SC04
  - Broke record again at SC05
- Standardization? ☹
- fastsoft ☺
  - Refine FAST
  - Deploy as appliance
  - Change the world?

SLAC

CERN

Fermilab