JetMax: Scalable Max-Min Congestion Control for High-Speed Heterogeneous Networks

Yueping Zhang

Joint work with Derek Leonard and Dmitri Loguinov

Internet Research Lab
Department of Computer Science
Texas A&M University, College Station, TX 77843

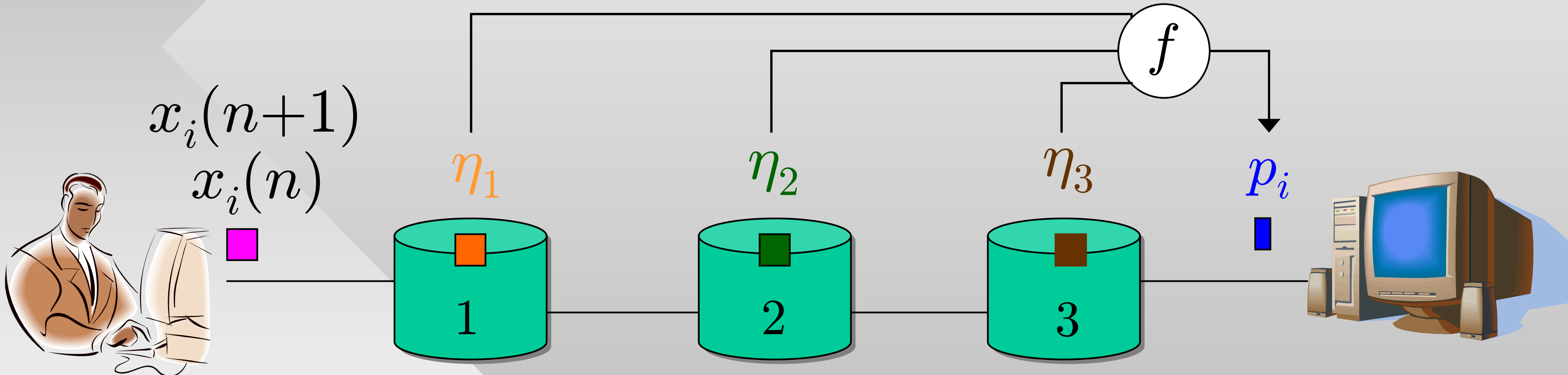
April 26, 2006

Agenda

- Introduction
  - Explicit congestion control and its designed properties
- Analysis of existing max-min methods
  - XCP, MKC, and MKC-AVQ
- JetMax
  - Stability and fairness
  - Performance and simulations
- Wrap-up

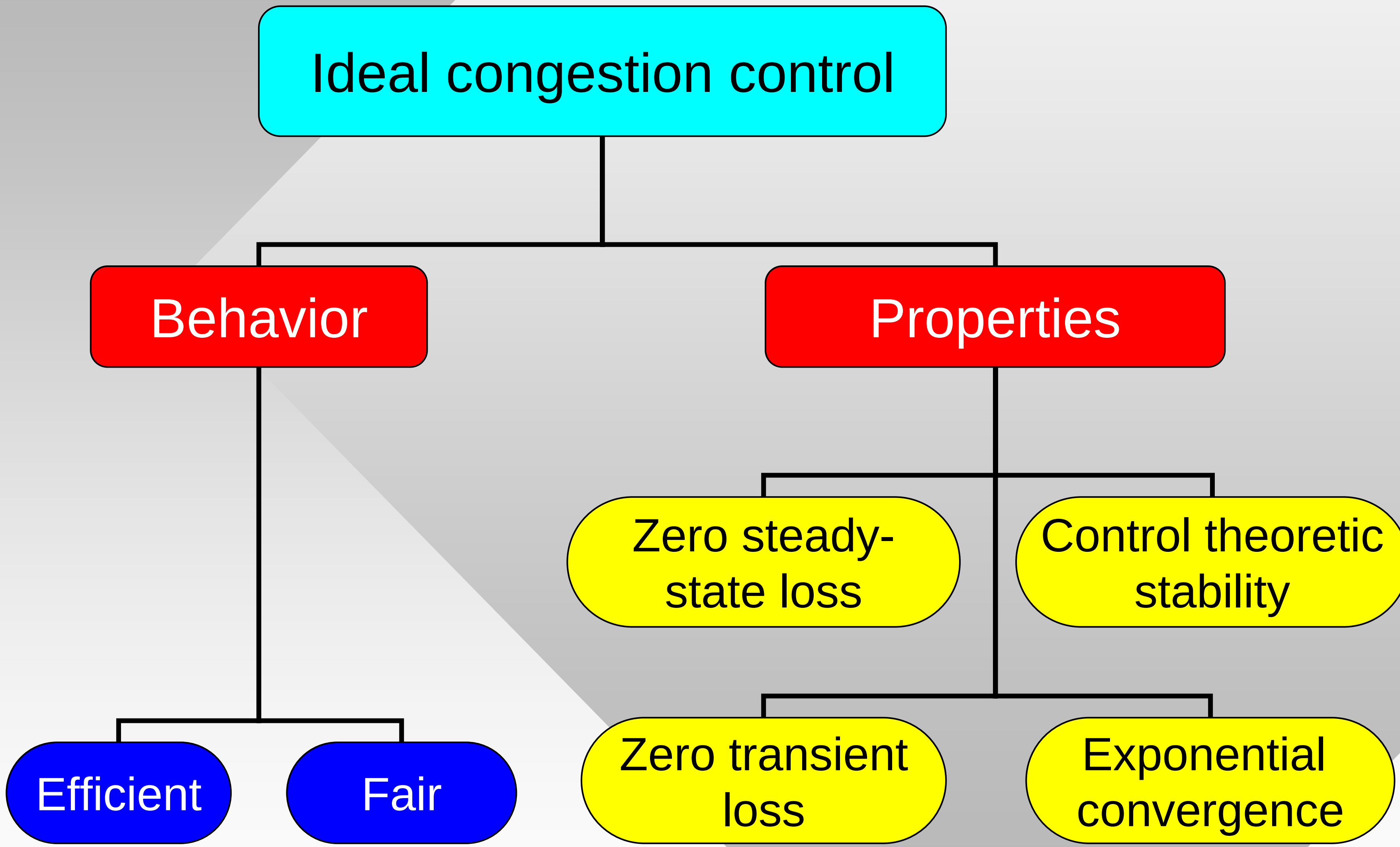
Explicit Congestion Control

- In explicit congestion control, each user i forms a control loop around its sending rate $x_i(n)$ and feedback $p_i(n)$



- Two directions emerged
  - $p_i = \Sigma\eta_l$: additive feedback (e.g., proportional fairness)
  - $p_i = \max(\eta_l)$: max-min feedback (e.g., max-min fairness)
- What properties should a congestion control protocol ideally possess?

Ideal Congestion Control



Can existing methods achieve these goals?

Agenda

- Introduction
  - Explicit congestion control and its designed properties
- Analysis of existing max-min methods
  - XCP, MKC, and MKC-AVQ
- JetMax
  - Stability and fairness
  - Performance and simulations
- Wrap-up

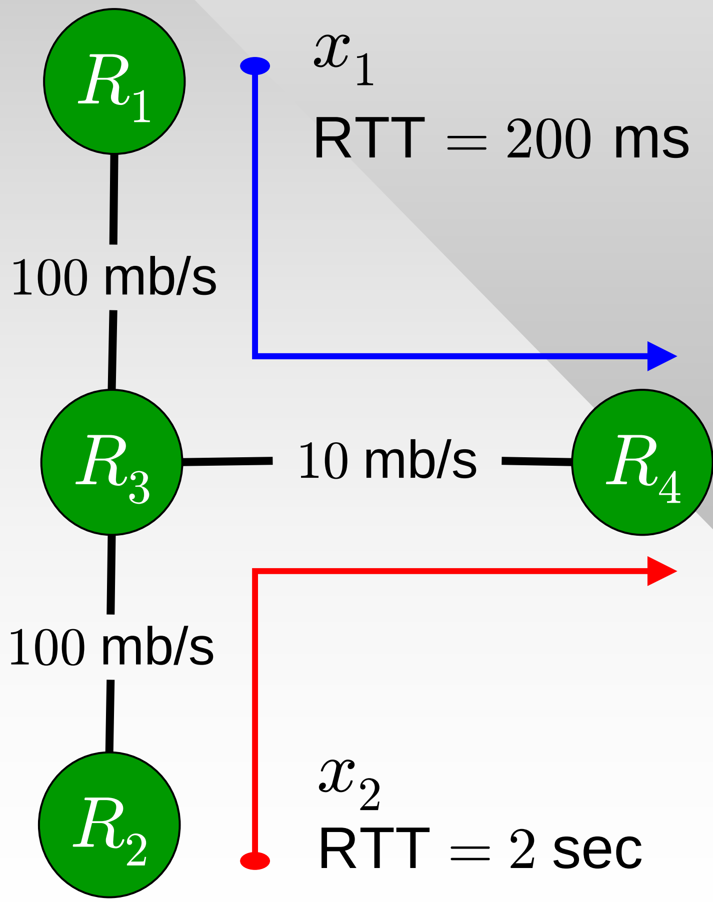
Existing Max-min Congestion Controls

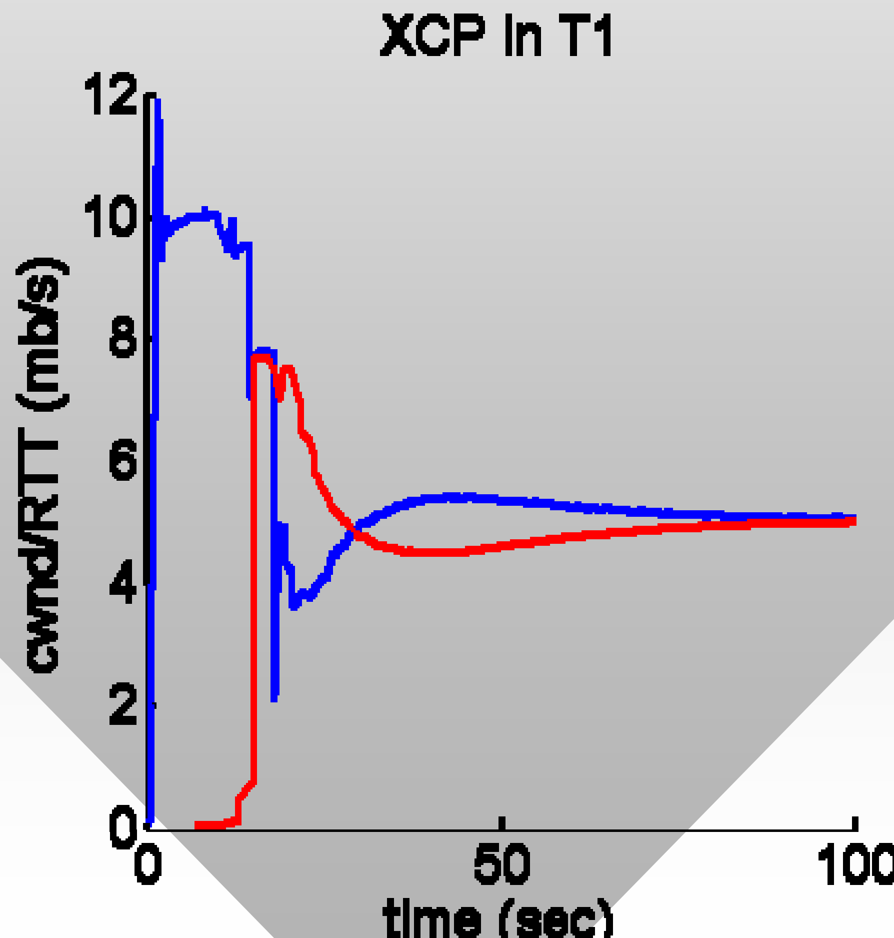
- We study three explicit congestion control methods
- XCP (eXplicit Control Protocol)
  - Each router implements a fairness and efficiency controller
- MKC (Max-min Kelly Control)
  - Modification of Kelly's equations for max-min feedback
- MKC-AVQ
  - Combination of MKC and Adaptive Virtual Queue (AVQ)
  - Eliminates steady-state loss of MKC

Stability of Existing Max-min Methods

- All three should be stable in single-link networks
  - We next compare their behavior under heterogeneous delay

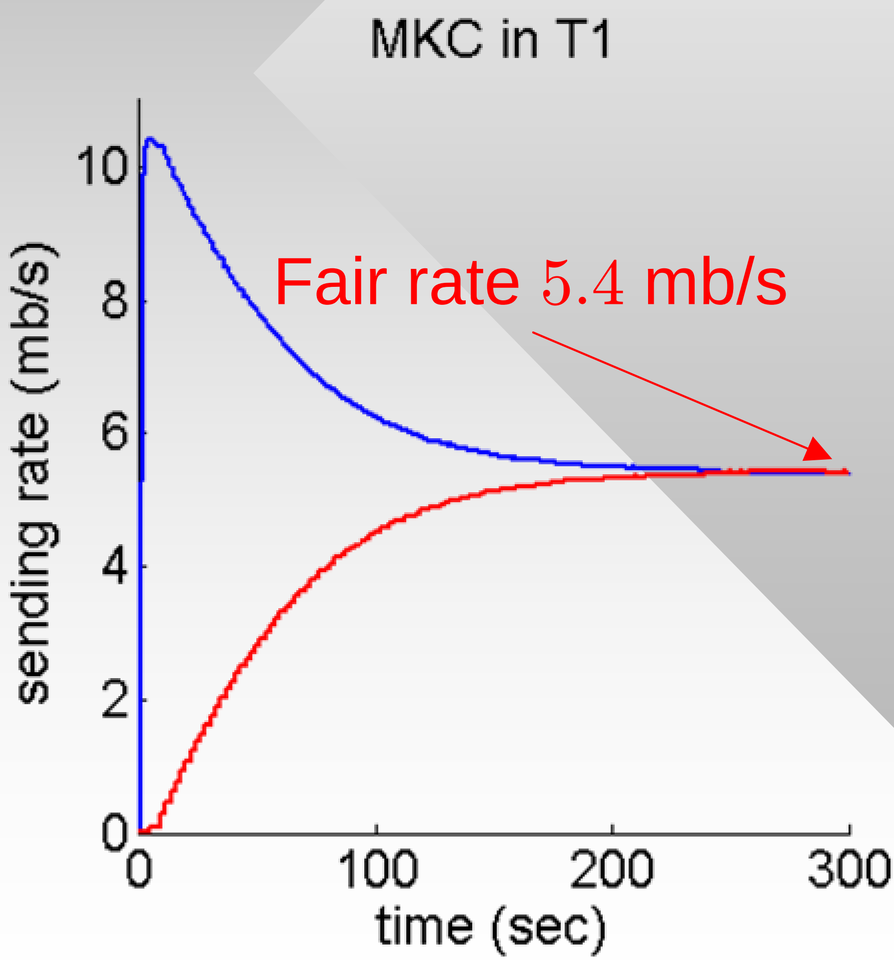
Topology T_1

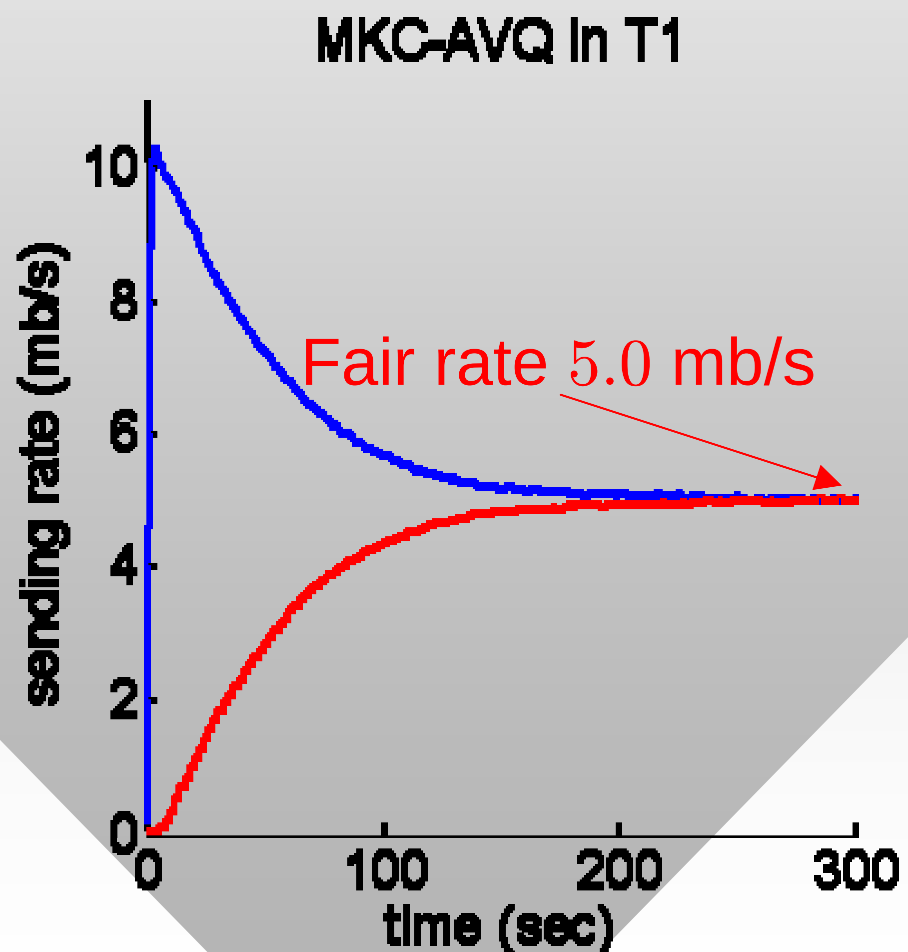




Stability of Existing Max-min Methods 2

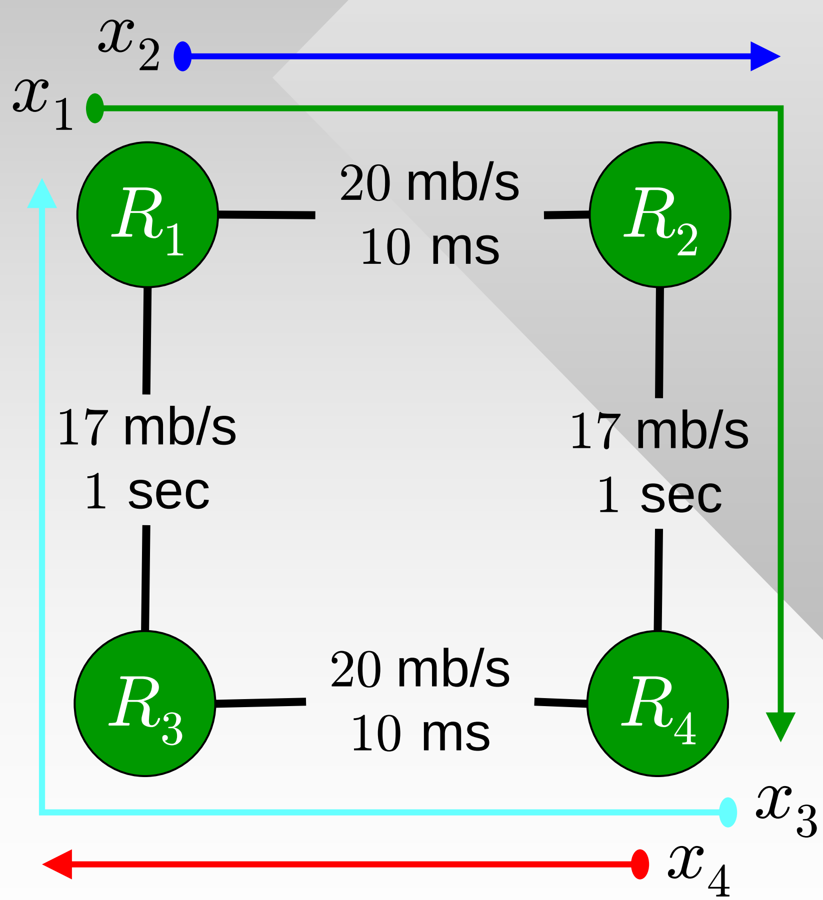
- MKC-based methods have no stability problems, but are very slow



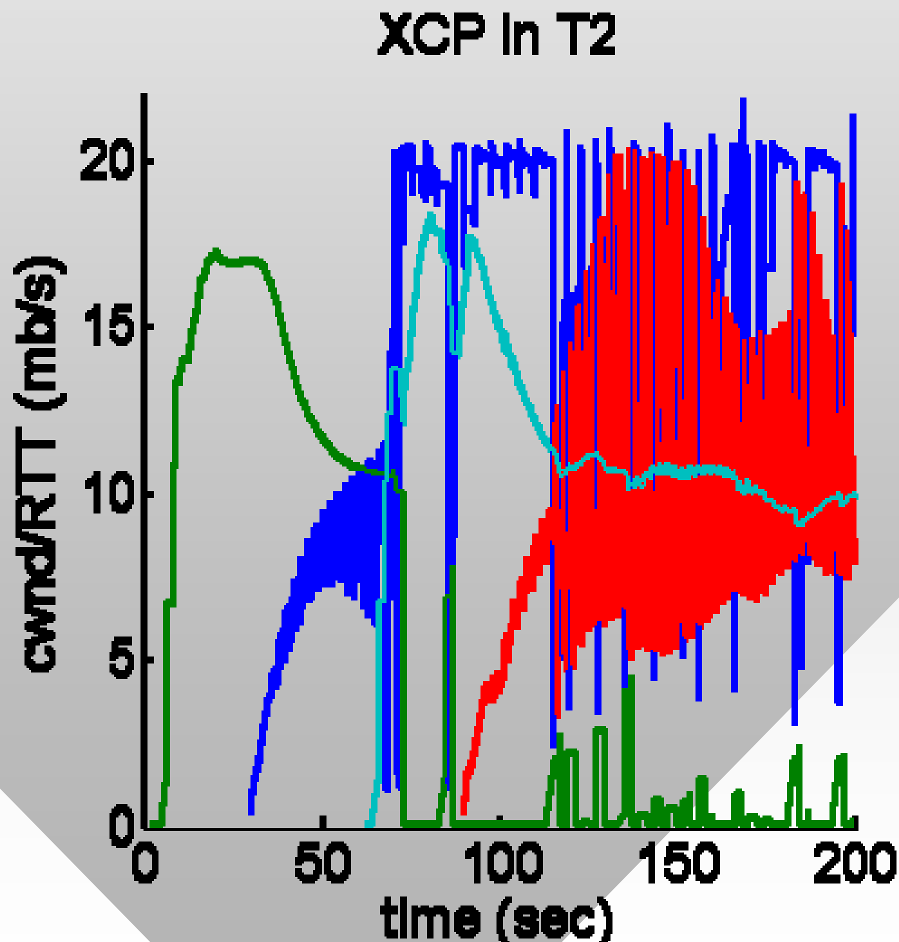


Stability of Existing Max-min Methods 3

- Multi-link behavior
  - XCP example shows that bottleneck oscillations are possible



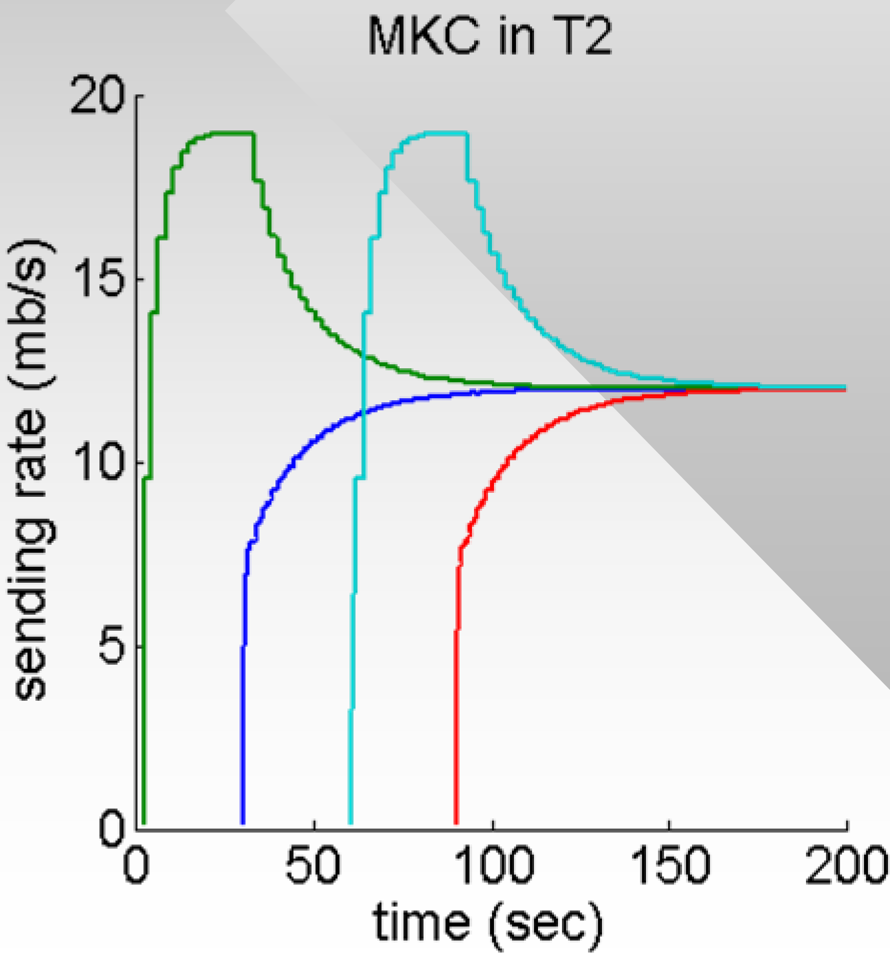
topology T_2

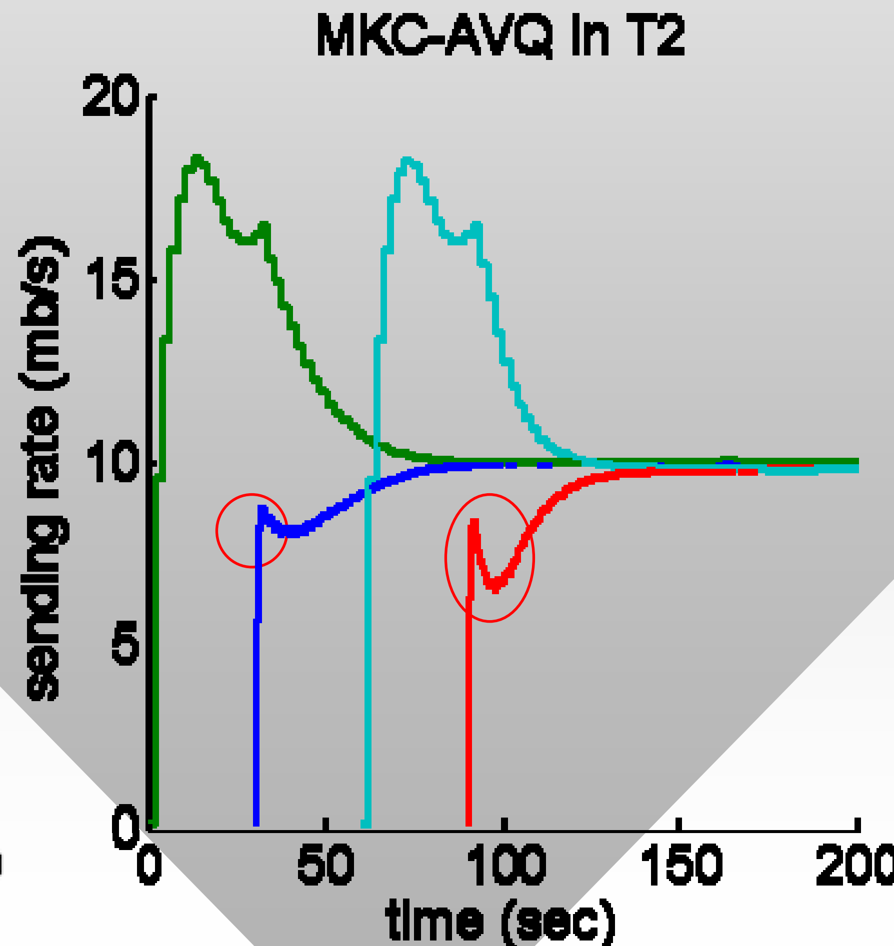


Computer Science, Texas A&M University

Stability of Existing Max-min Methods 4

- MKC and MKC-AVQ remain stable
  - AVQ overshoots available bandwidth in transient states, but is stable otherwise





Properties of Existing Max-min Methods

- XCP
  - Unstable in certain multi-link topologies
  - Convergence rate to fairness is unknown, but can be as large as 100s of RTTs if feedback delay is heterogeneous
- MKC
  - Steady-state packet loss proportional to N
  - Fairness is reached in $\Theta(C)$ steps
- MKC-AVQ
  - Transient overshoot proportional to N
  - Convergence rate the same as in MKC
- The bottom line – current explicit-feedback methods do not satisfy our design criteria and offer little benefit over TCP

Agenda

- Introduction
  - Explicit congestion control and its designed properties
- Analysis of existing max-min methods
  - XCP, MKC, and MKC-AVQ
- JetMax
  - Stability and fairness
  - Performance and simulations
- Wrap-up

JetMax Design

- Assume that $N_l(n)$ is the number of users congested by router l at time n (we call these flows responsive) and $w_l(n)$ is their combined rate
  - Label the remaining flows passing through l unresponsive and assume $u_l(n)$ is their total rate
- The main idea of JetMax is very simple
  - Divide the available bandwidth of l equally among N_l flows

capacity of link l

current fair share at router l

$$g_l(n) = \frac{\gamma_l C_l - u_l(n)}{N_l(n)}$$

combined rate of unresponsive flows

desired link utilization

number of responsive flows

JetMax Design 2

- Users receive feedback $g_l(n)$ and utilize it in their controller

RTT of user i

$$x_i(n) = x_i(n - D_i) - \tau\Big(x_i(n - D_i) - g_l(n - \overleftarrow{D_i})\Big)$$

constant

delay from the bottleneck to user i

- The final issue is bottleneck switching
  - Routers decide to make a switch based on the virtual packet loss contained in the header:

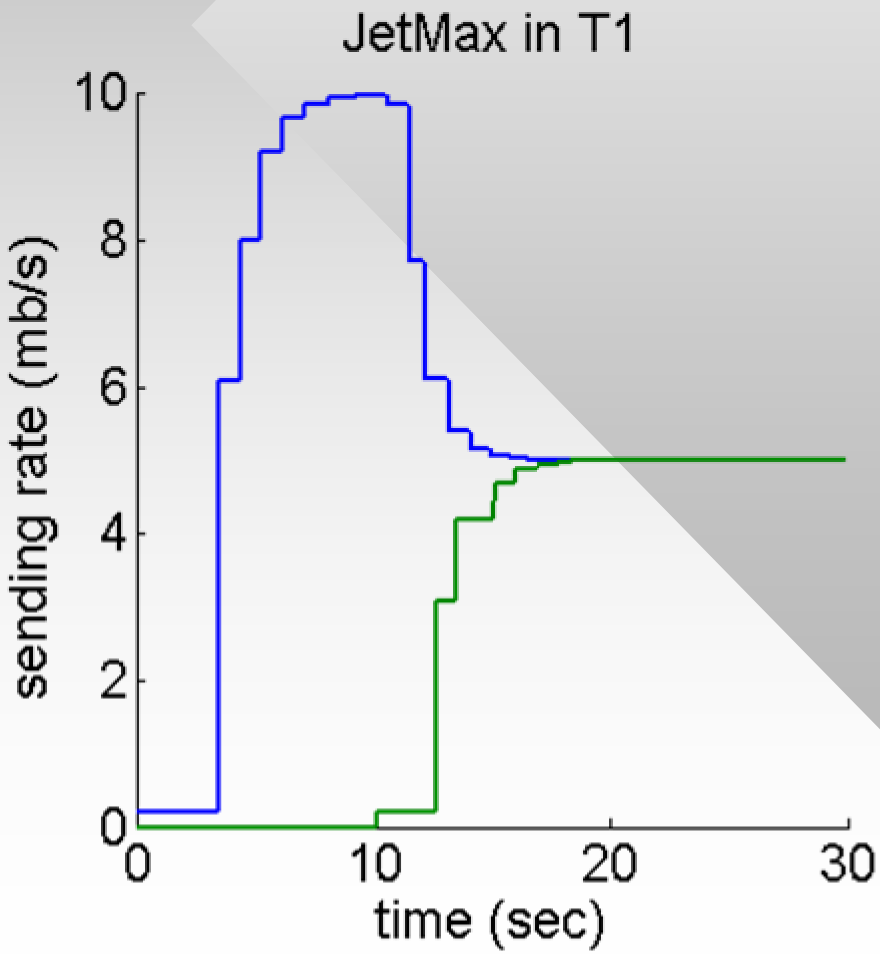
$$p_l(n) = \frac{w_l(n) + u_l(n) - \gamma_l C_l}{w_l(n) + u_l(n)}$$

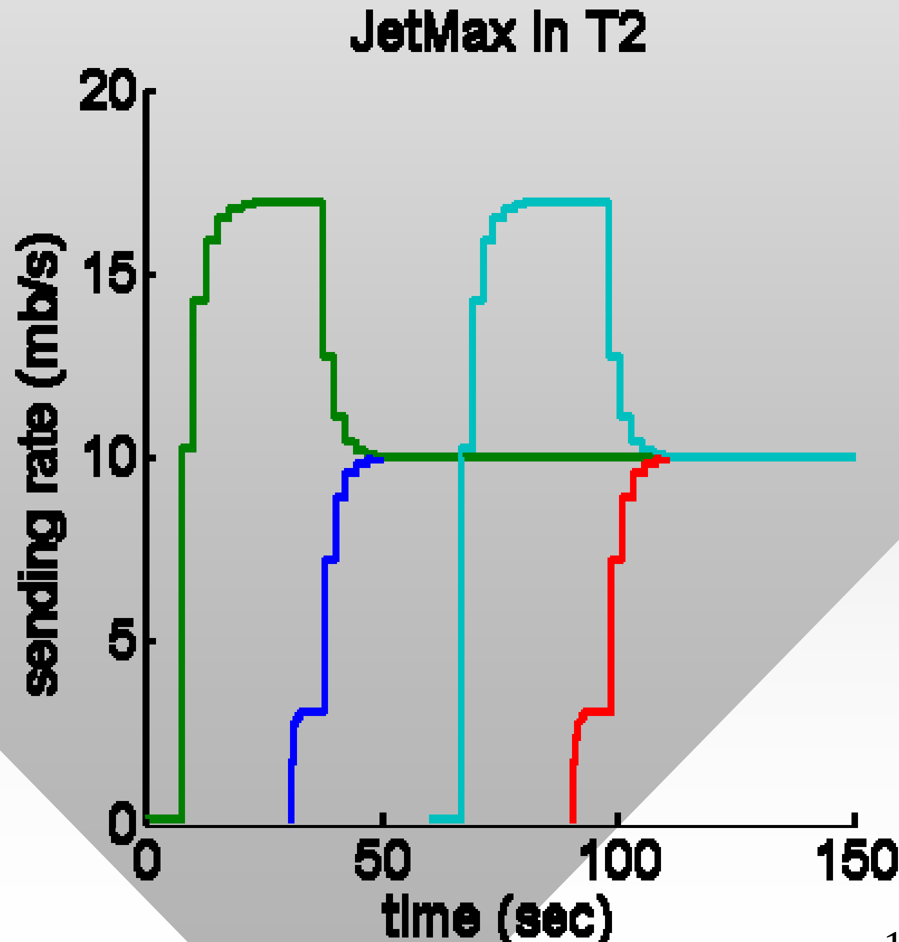
JetMax Properties

- Theorem 1: Under any fixed bottleneck assignment under max-min feedback, JetMax is globally asymptotically stable regardless of delay if and only if $0 < \tau < 2$
  - For values of $\tau \leq 1$, the controller is also monotonic
- Theorem 2: Stationary resource allocation of JetMax is max-min fair
- Theorem 3: On a single link of any capacity, JetMax converges to within ε-percent of efficiency and fairness in a fixed number of RTT steps
  - For $\tau = 0.5$ and ε=1%, this is 6 steps

JetMax Simulations

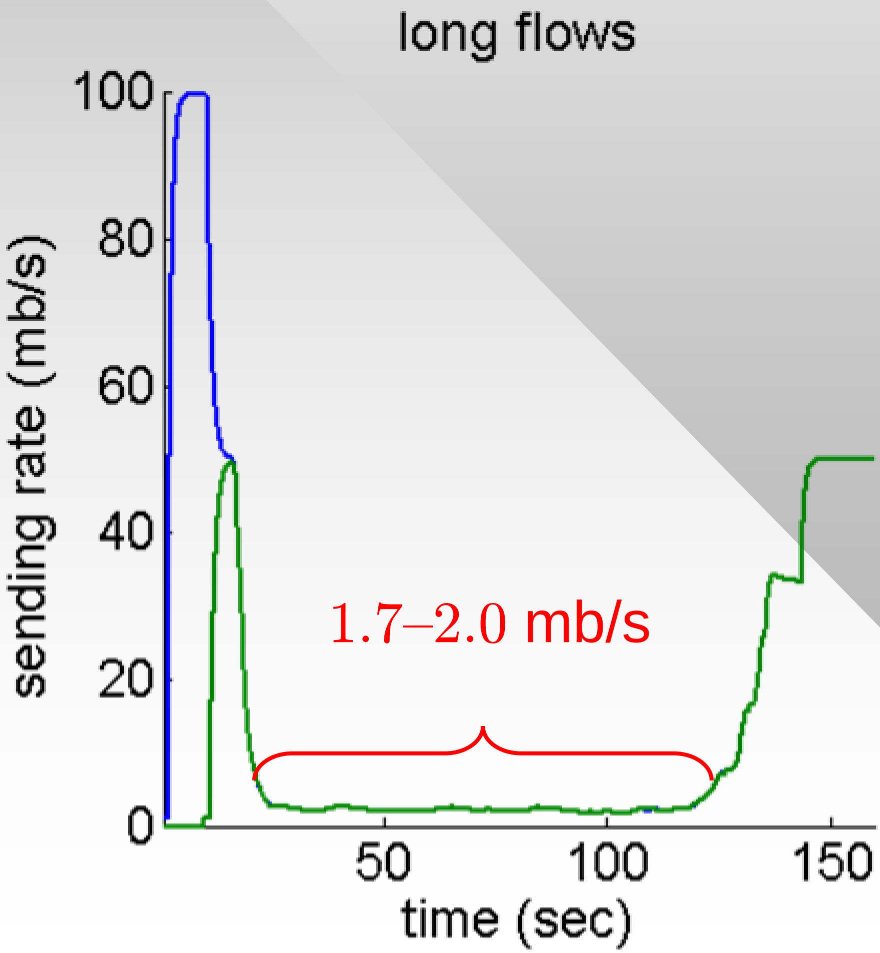
- We start with ns2 simulations in T_1 and T_2
  - Topology T_1 is changed to introduce random time-varying feedback delay in the control loop – no effect on stability

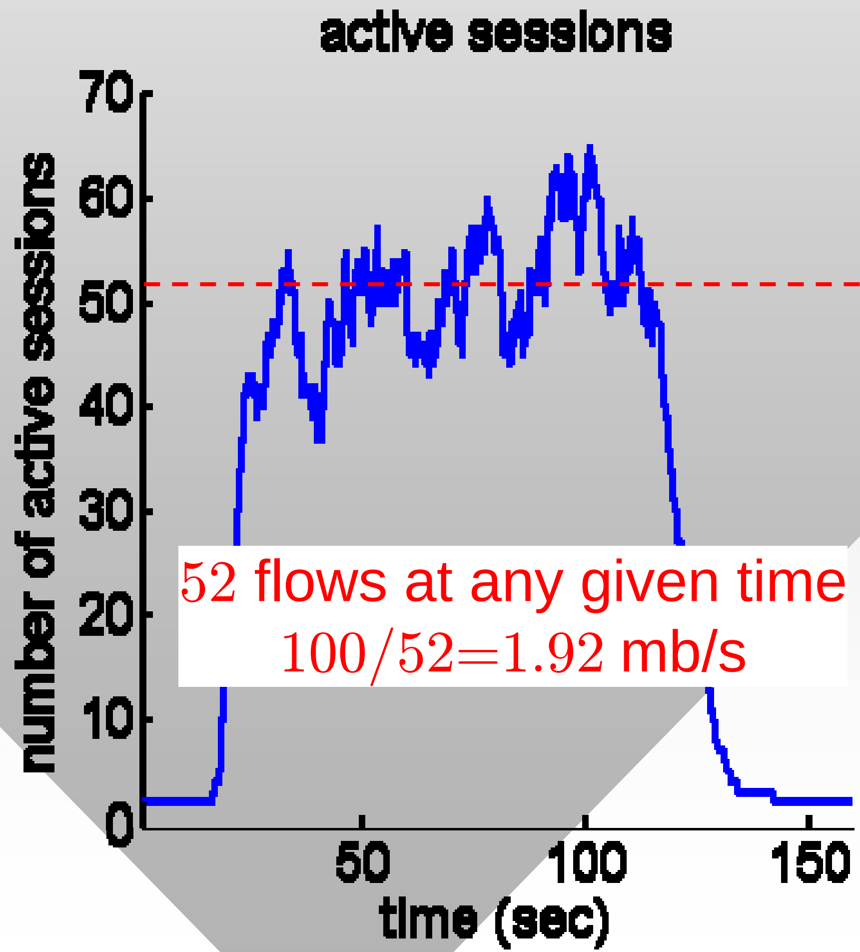




JetMax Simulations 2

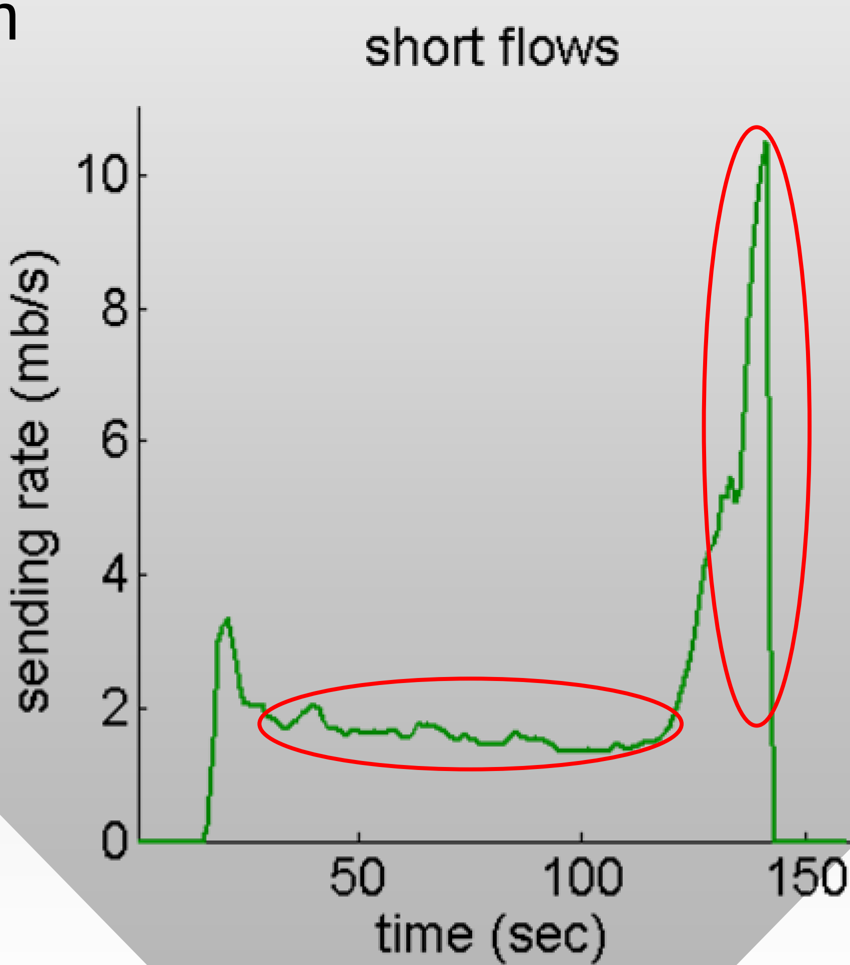
- Performance under mice cross-traffic
  - A single link of capacity 100 mb/s shared by two long flows and a total of 500 short flows with random RTT in $[40,1040]$ ms and uniform packet size in $[800,1300]$ bytes





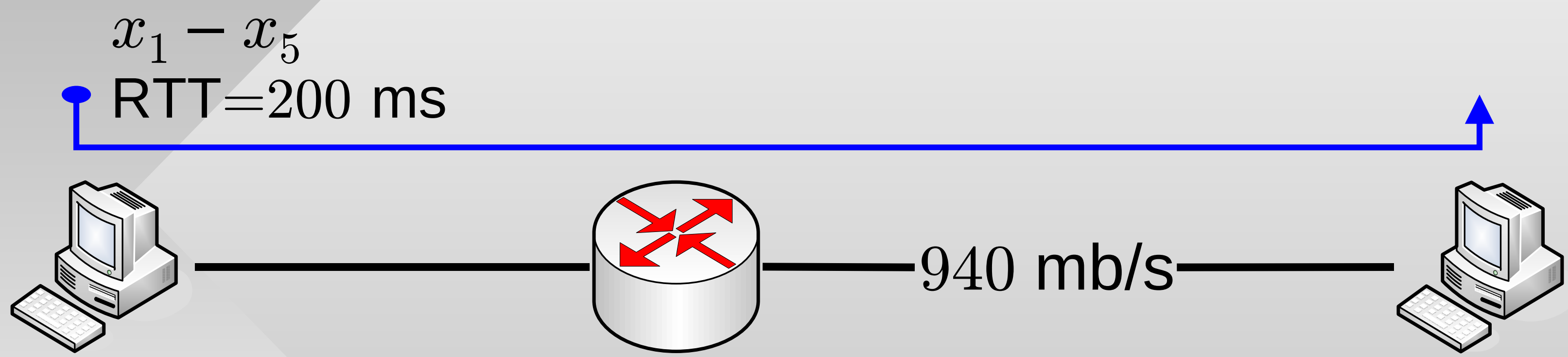
JetMax Simulations 3

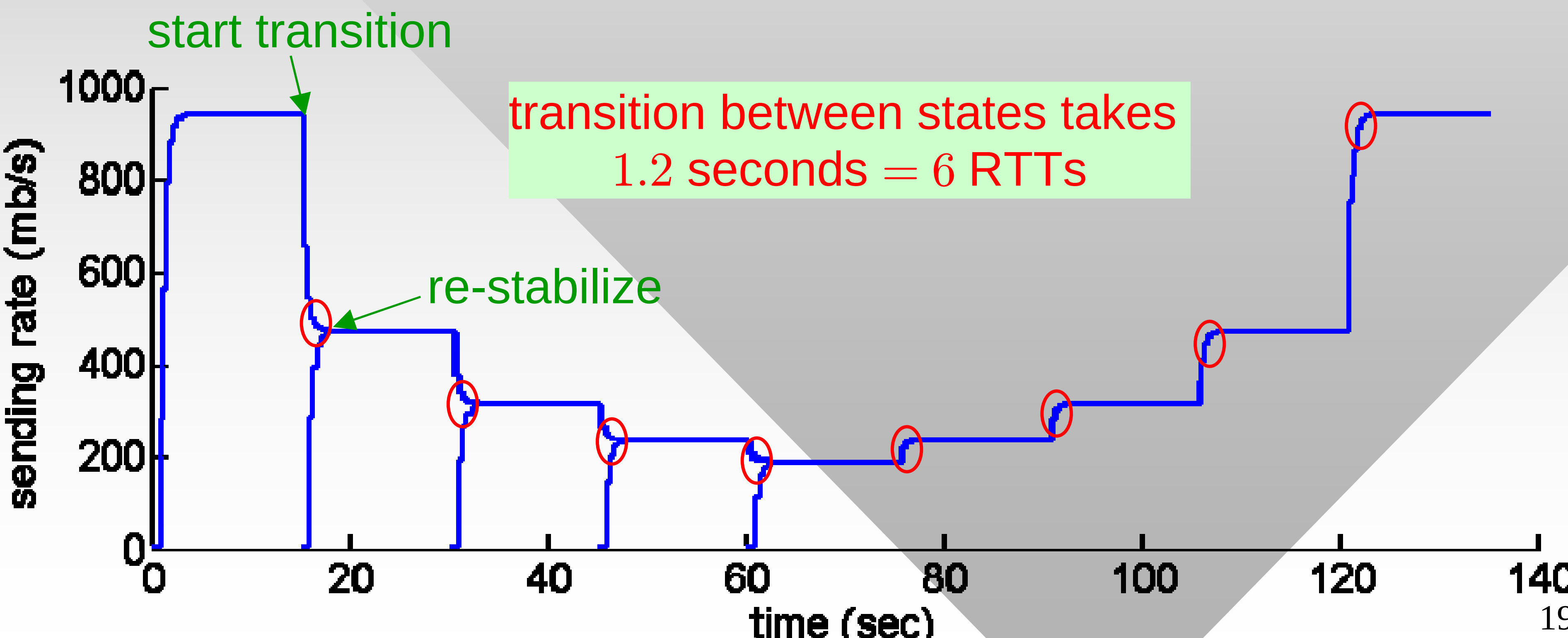
- Short flows
  - Each lasts for a random duration with mean 10 seconds
  - Achieve rates very close to fair-share 1.92 mb/s
- As flows depart after 120 seconds, their bandwidth is consumed by remaining flows



JetMax Linux Experiments 1

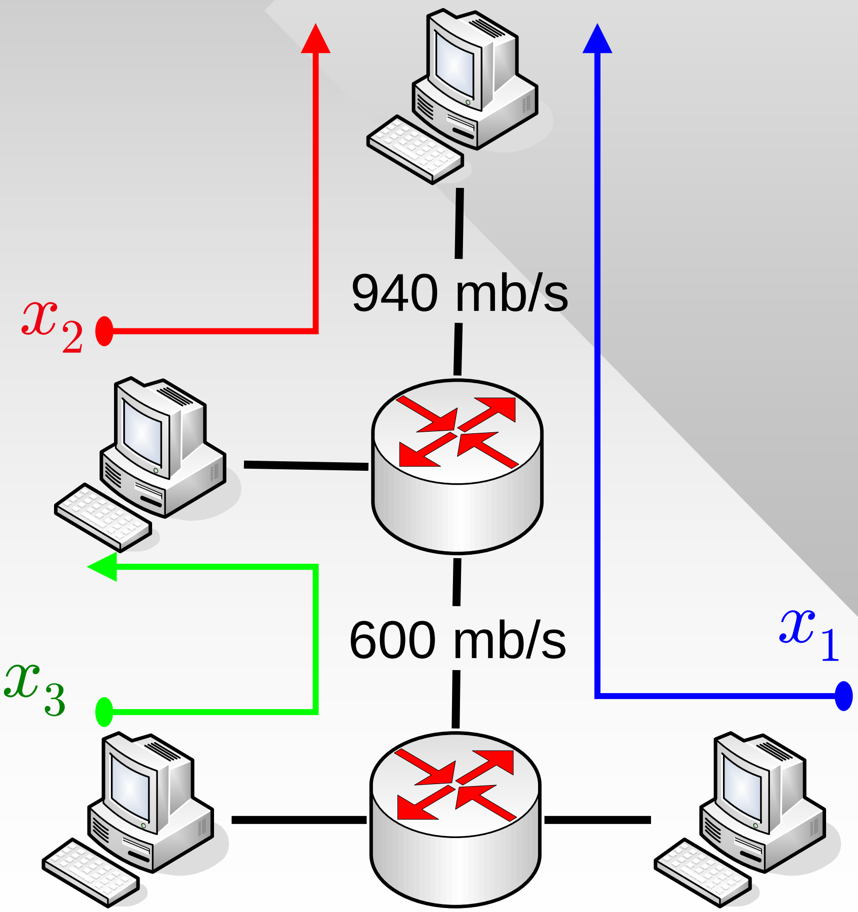
- Single-link scenario

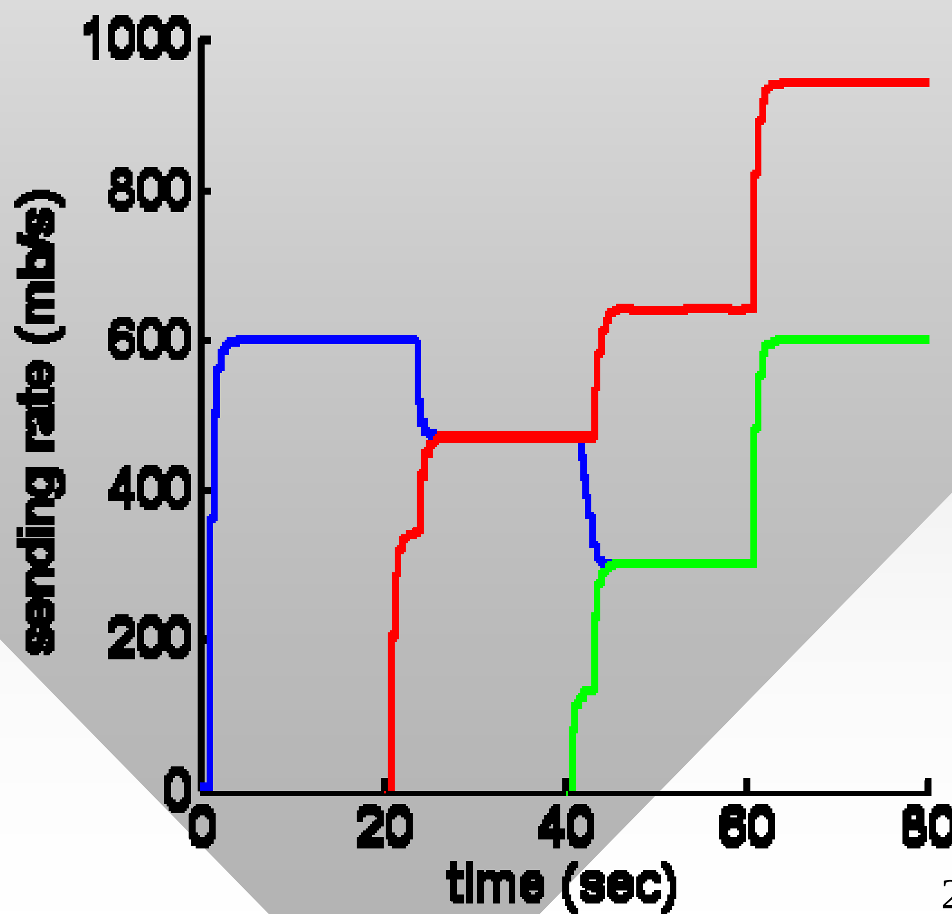




JetMax Linux Experiments 2

- Multi-link scenario
  - Flow x_1 starts first, x_2 arrives at time 20 seconds, x_3 arrives at 40 seconds, and x_1 departs at 60 seconds





Wrap-up

- JetMax summary
  - Converges to the stationary state in the same number of RTTs regardless of link capacity and the number of flows
  - Achieves tunable utilization and zero loss
  - Is monotonic and stable under any time-varying delay
- More in the paper
  - Estimation of responsive/unresponsive rates
  - Bottleneck membership maintenance
  - Avoidance of transient overshoot
  - Additional simulations and experiments
- Linux and ns2 code
  - http://irl.cs.tamu.edu/mkc