TCP MaxNet Implementation and Experiments on the WAN in Lab

Martin Suchara, Ryan Witt,
Bartek Wydrowski

California Institute of Technology

Pasadena, U.S.A.

Problems with Current Implementations of TCP

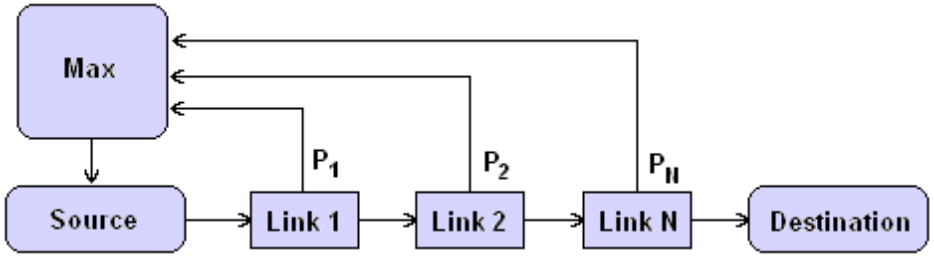
- The underlying algorithms usually do not scale with the speeds or sizes of networks properly
- They usually do not share available capacity with max-min fairness
- They usually have poor behavior on loss because loss is interpreted as a congestion signal
- The problem is that current TCPs react to correlation rather than causation of congestion…
- We provide implementation of TCP MaxNet, a protocol that solves these shortcomings by EXPLICITLY SIGNALING THE CONGESTION LEVEL

Outline

- Description of the MaxNet architecture
- Our implementation of TCP MaxNet in the Linux kernel
- Experimental setup allowing protocol performance evaluation on WAN in Lab
- Experimental results – comparison to other protocols
  - Convergence properties and RTT in the network
  - Fair sharing of available capacity
  - Behavior on packet loss
- Issues with partial deployment

Description of the MaxNet Architecture Overview

- Congestion level at each link in the network is monitored by routers.
- Sender only reacts to the maximum level of congestion on one of the links on the end-to-end path
- The window size of the sender is a function of this maximum



- Implementation involves changes at both the SOURCE and ROUTER side

Description of the MaxNet Architecture Design of the Router

- **Routers periodically calculate the current congestion level on each of their links**

$$Price(t+1) = Price(t) + \gamma(X(t) - \mu C)$$

C	**the capacity of the output link**
μ	**efficiency parameter**
$X(t)$	**number of bytes enqueued since last price calculation**
γ	**let $\gamma \approx 1 / C$**

- **Price approximates how long it takes to empty the queue if the router transmits at rate μC**

Description of the MaxNet Architecture Design of the Sender

- **Sender sets congestion window as a function of the *maximal* level of congestion on the end-to-end path**
- **Congestion window is set as follows:**

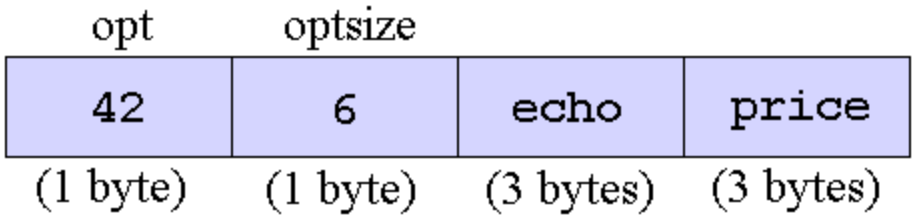
$$\eta = \eta + \frac{p1}{cwnd} - \frac{p2 \times price}{baseRTT} \times interval$$

$$estWnd = p3 \times \exp(\eta - \frac{p4 \times price}{baseRTT})$$

- **Parameters p1 and p3 determine the rate of the convergence**
- **Parameters p2 and p4 determine the severity of the reaction to the congestion signal**
- **Sender does not decrease its sending rate on loss**

Implementation of TCP MaxNet in the kernel

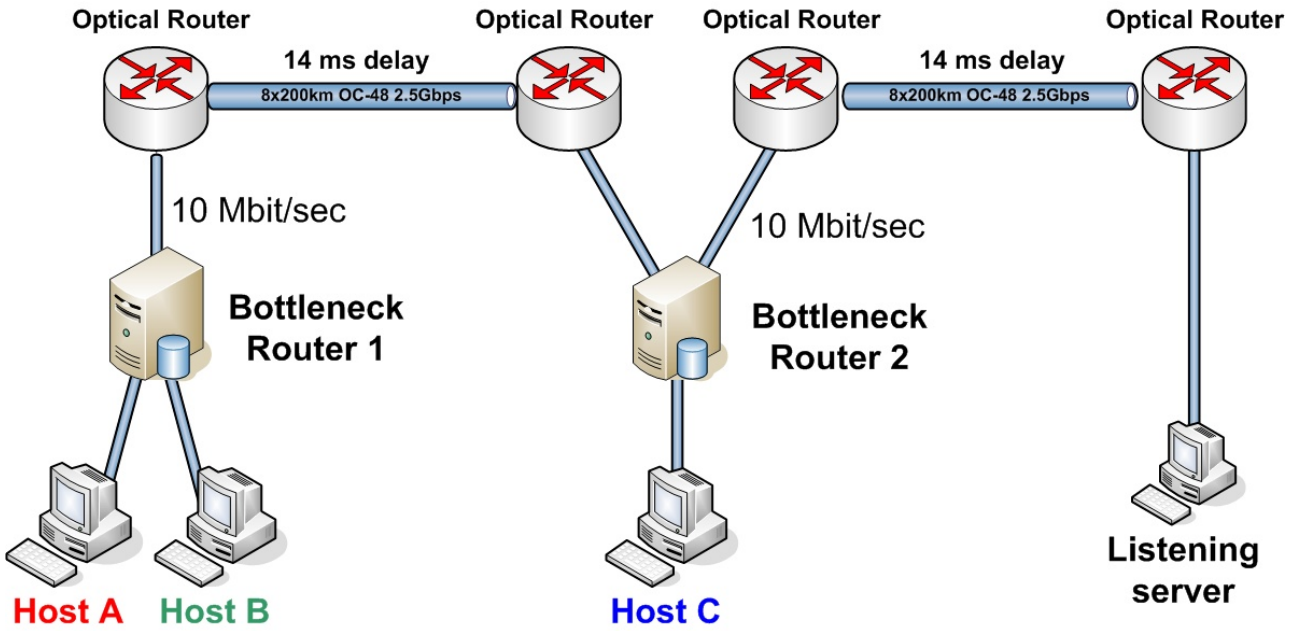
- **Communicating the maximal price to the sender**
  - **A new TCP option with two fields, `price` and `echo`, is attached to each packet**
  - **Routers overwrite the `price` option if their own price is higher than the one already there**
  - **The receiver places this value in the `echo` field and the options are returned to the sender with the ACK**



Implementation of TCP MaxNet in the kernel

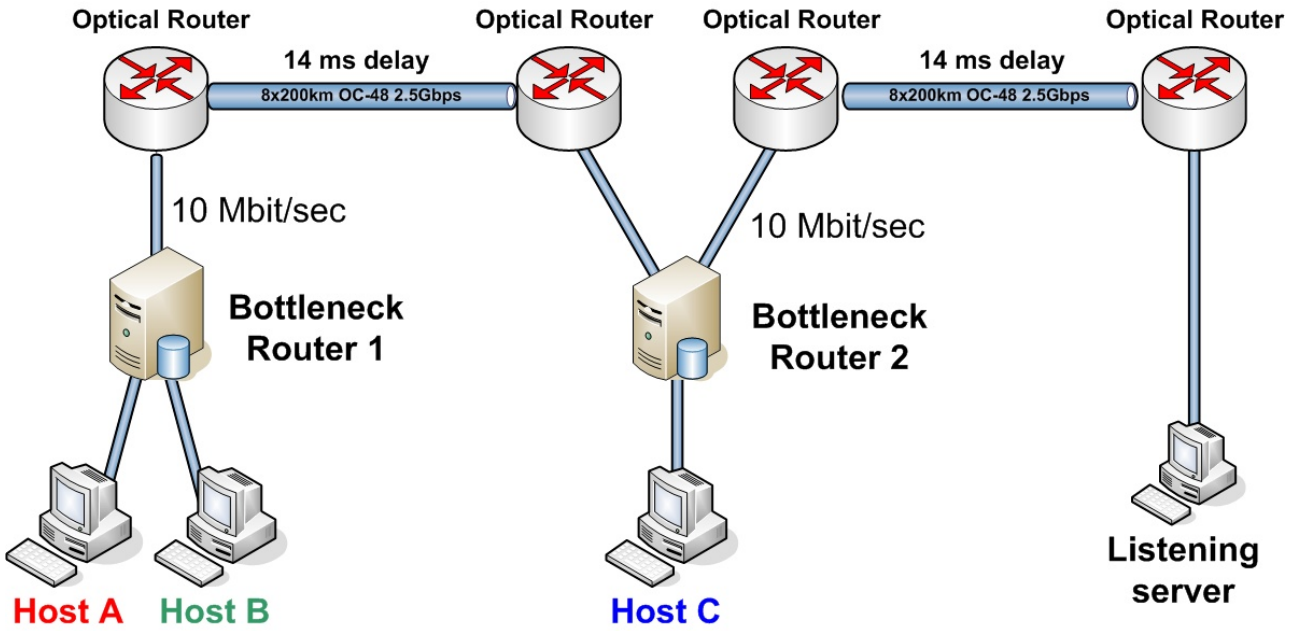
- New challenges encountered in the Linux kernel
  - The TCP header size in Linux is limited to 60 bytes
  - Up to 28 bytes are used by SACKs
  - Only 2 SACK blocks per packet header are allowed in our implementation
- Convergence requires high precision of all calculations
  - Exponential was calculated using a lookup table
  - Fractional numbers need to be used
  - We are experimenting with various encodings of price that work for extreme rates and RTTs

Experimental Setup
WAN in Lab



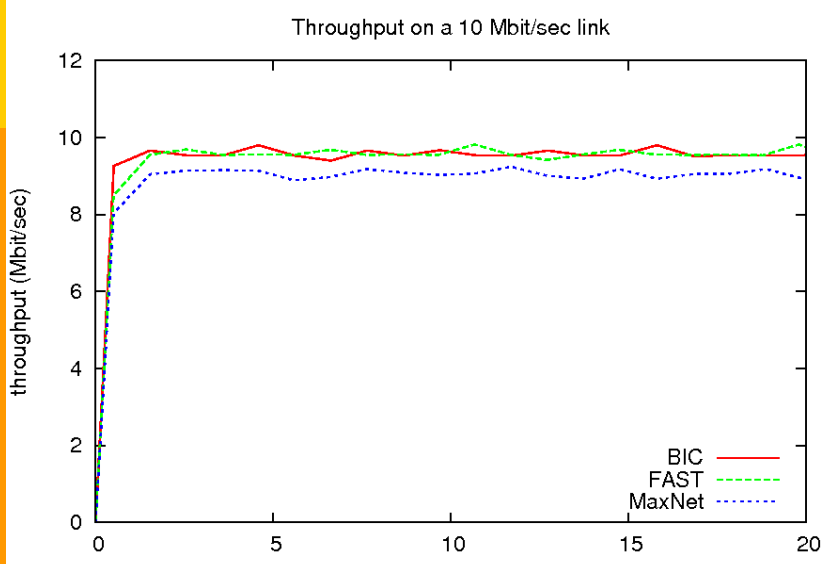
- A real WAN network in a laboratory
  - An array of reconfigurable routers, servers and clients
  - the backbone of the network connected by 2 x 1600km OC-48 introducing real propagation delay

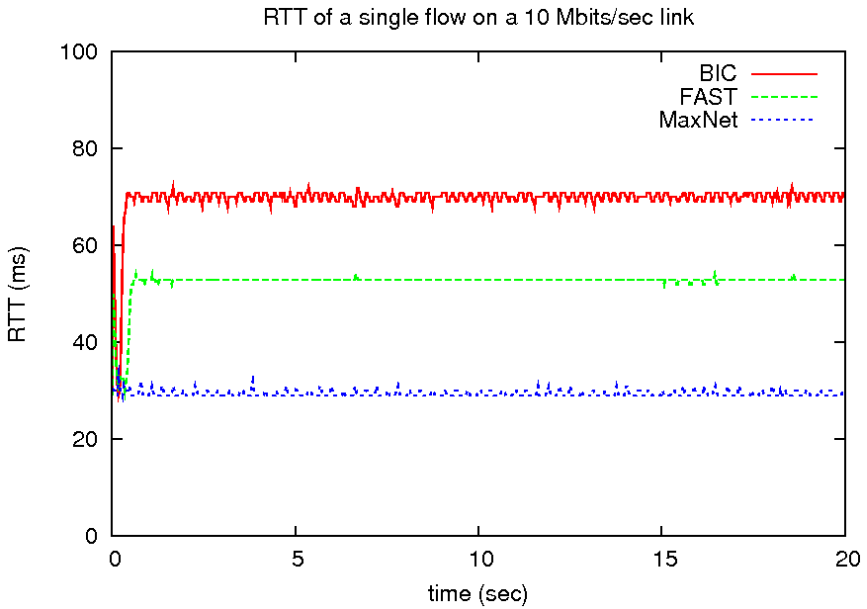
Experimental Setup
WAN in Lab



- Router code installed on Bottleneck Router 1 and 2
- Sender code installed on Host A, B, C and on the Listening server
- Performance monitored in the kernels of the hosts and routers

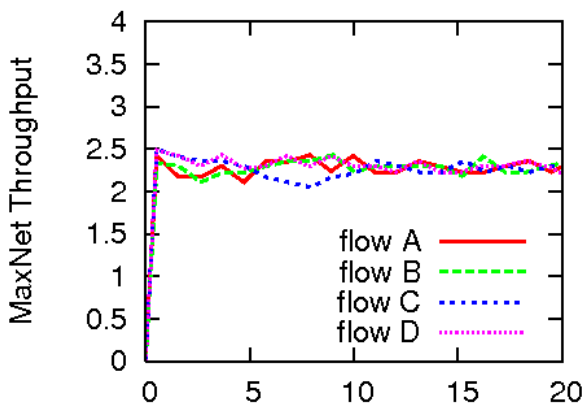
Convergence and RTT
Short Router Queues

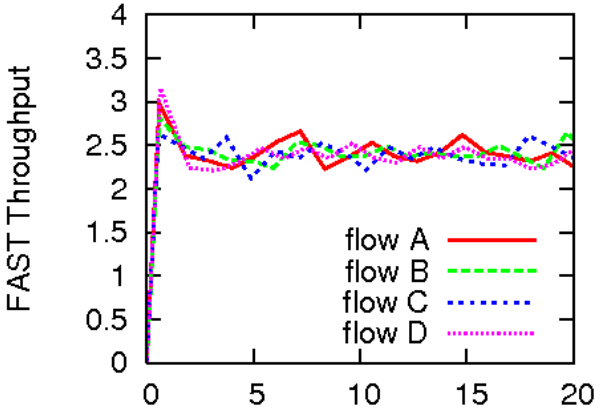


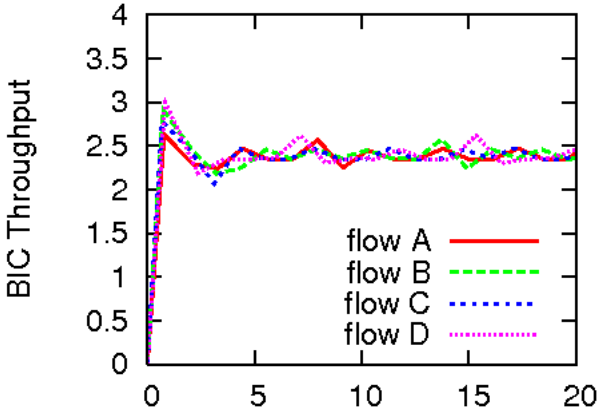


- RTT of **TCP MaxNet** stays close to the two-way propagation delay on the link
- This indicates that MaxNet keeps the router queues empty
- RTT of **TCP BIC** and **TCP FAST** is much higher

Fair Sharing
Identical Flows Through a Bottleneck



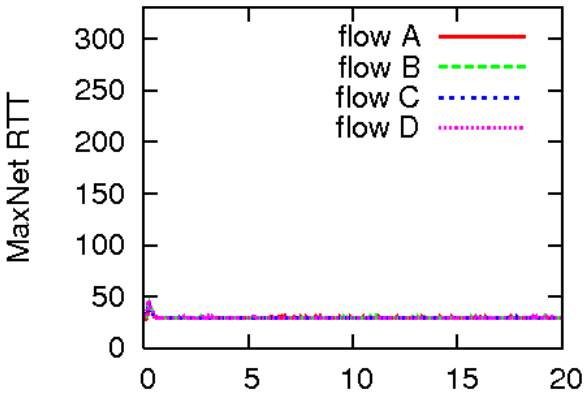


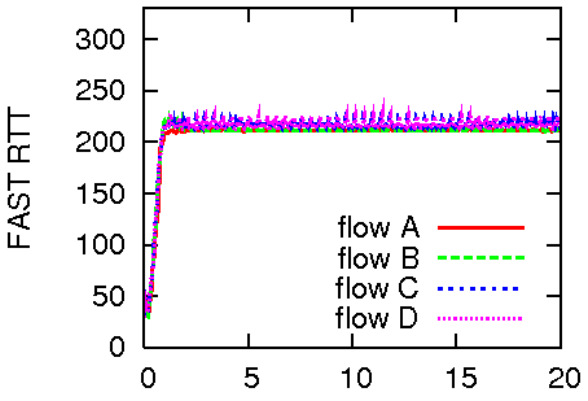


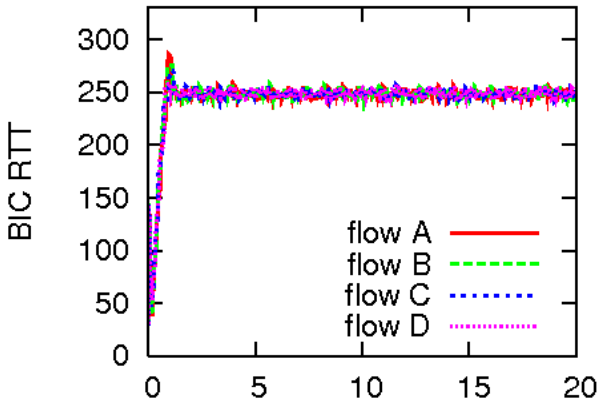
TCP MaxNet

TCP FAST

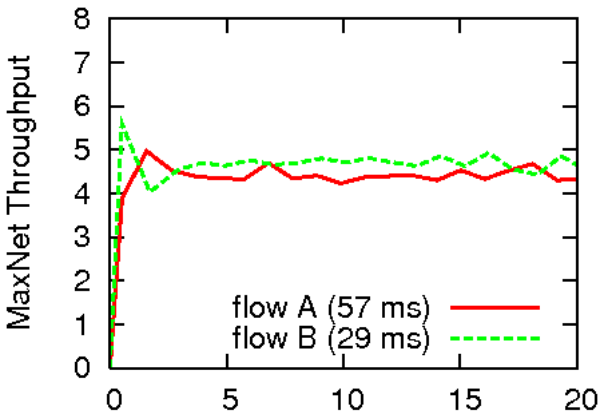
TCP BIC

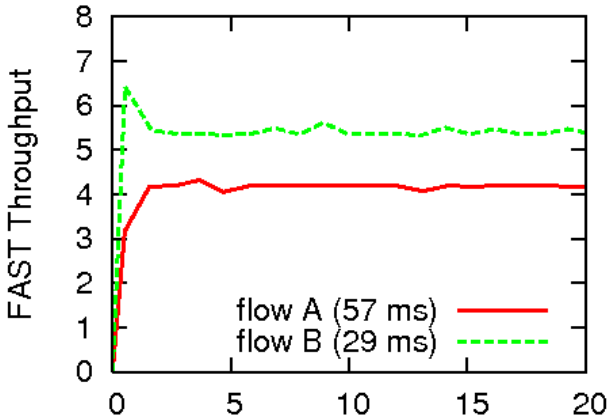


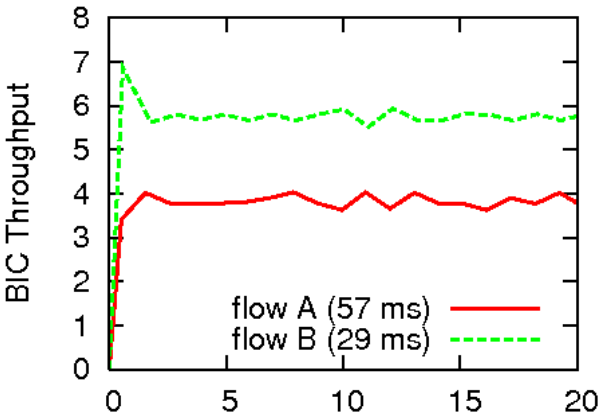




Fair Sharing
Flows with Differing RTTs



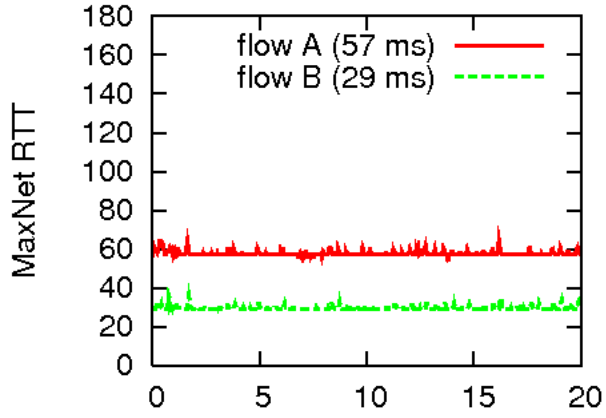


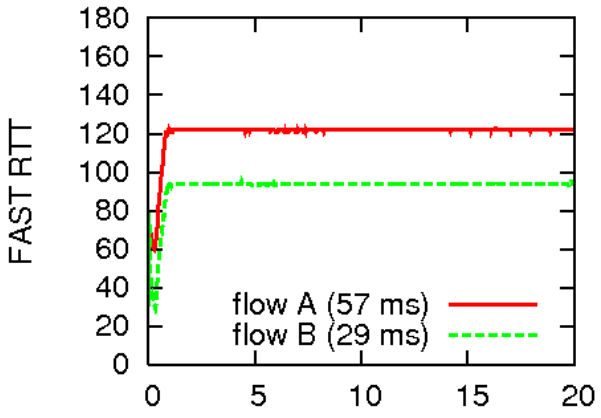


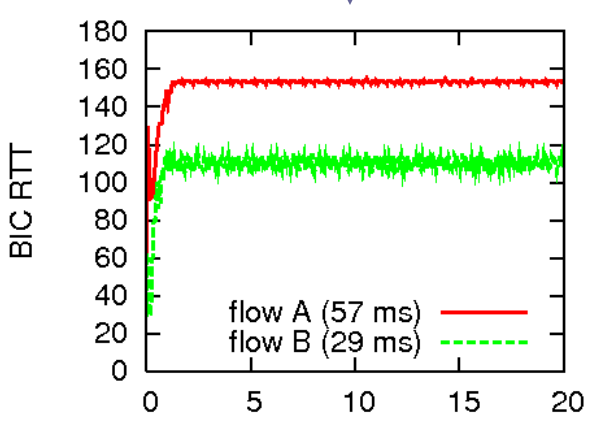
TCP MaxNet

TCP FAST

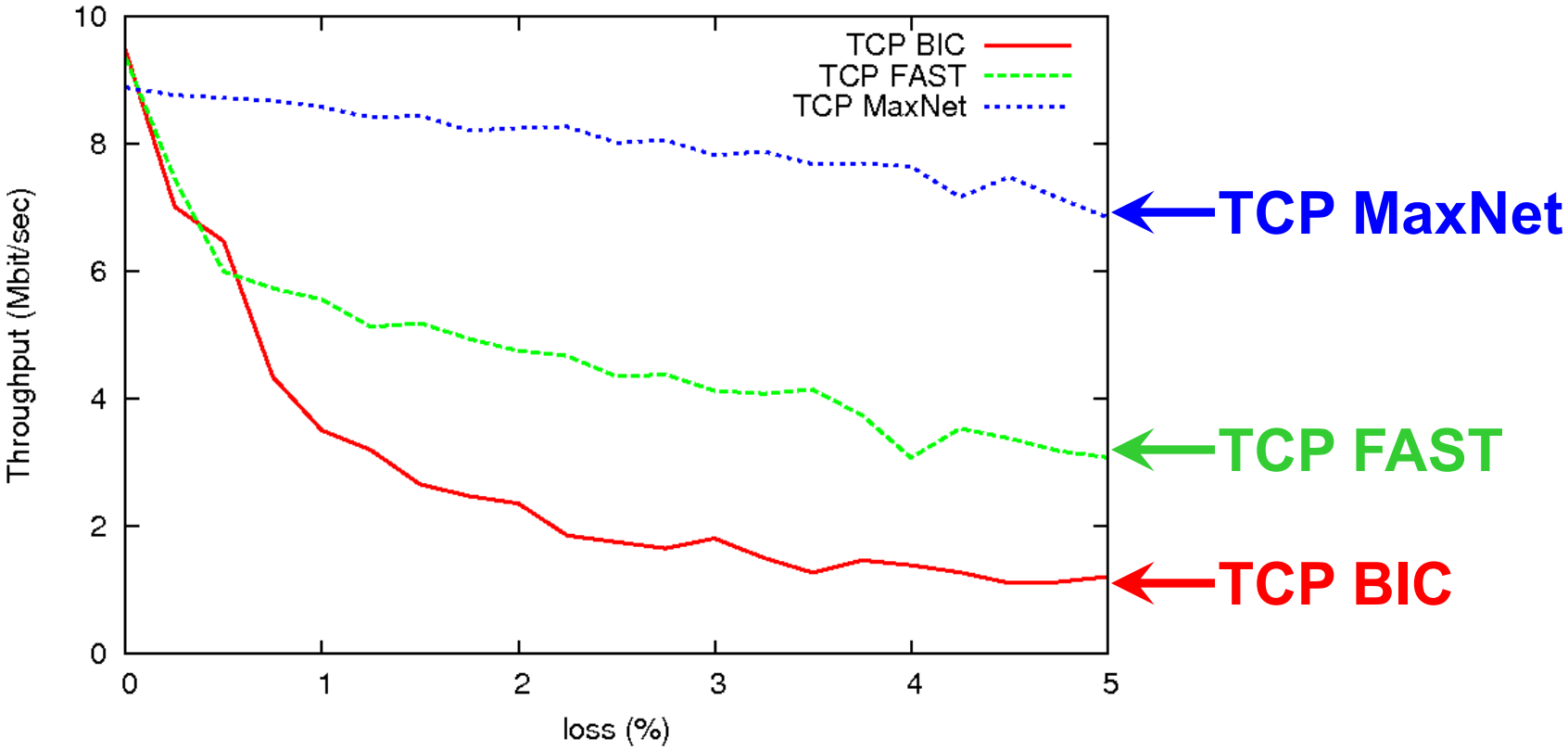
TCP BIC







Performance under Loss



- **TCP MaxNet** achieves the highest throughput because sender does not decrease its sending rate on loss

Summary

- We provide an implementation of TCP MaxNet in the Linux kernel
- We compare properties of TCP MaxNet and other protocols and observe:
  - Fair sharing of bottleneck capacity even when the propagation delays of the competing flows vary
  - Extraordinary performance in lossy environment
  - Low latency and short queue sizes (making TCP MaxNet an excellent protocol for real time applications)
- Some issues remain
  - Incremental deployment and fair sharing with other protocols

Deployment Issues

- Solving incremental deployment issues is important for usability of the new protocol
- The protocol needs to be supported by the sender, receiver and all the routers on the end-to-end path
- Initially, not all routers support TCP MaxNet
  - The best we can currently do is to detect this and fall back on loss based congestion control
  - Easy detection by duplicating time to live field that is decremented only by MaxNet routers
- Not all senders will use TCP MaxNet
  - TCP MaxNet is less aggressive and cannot compete fairly with loss or delay based TCP flows

Is Fair Dropping of Packets by MaxNet Routers the Solution?

- Using an approach similar to AFD (Approximate Fair Dropping) could solve the problem
  - AFD uses a shadow buffer that stores b recent packet headers to estimate the flow's current rate
  - Packets are dropped differentially based on the rate
  - Approximate max-min fairness is achieved while keeping queue length at q_{target}
- Some changes to AFD would be needed
  - Two separate queues: for MaxNet and for other traffic?
  - Increasing the signaled price instead of dropping a TCP MaxNet packet?

Questions & Discussion…