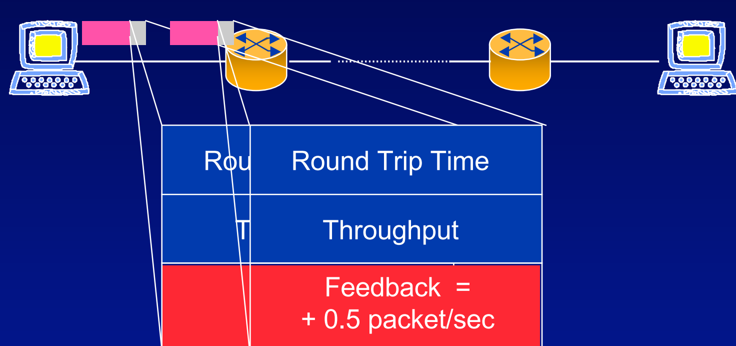
# XCP's Performance in the Presence of Malicious Flows

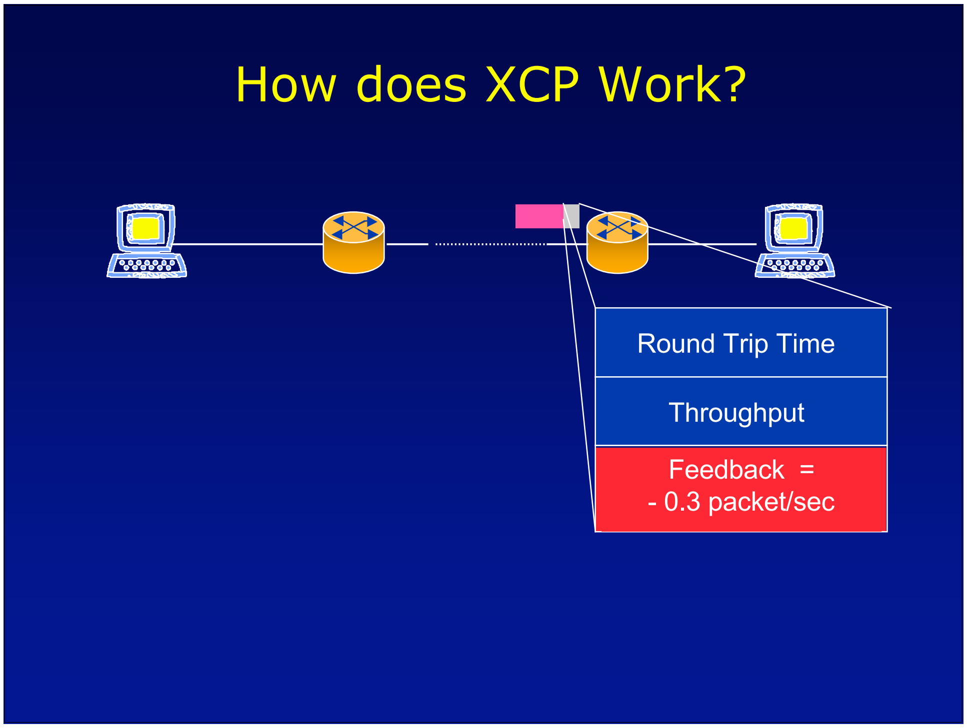
Dina Katabi

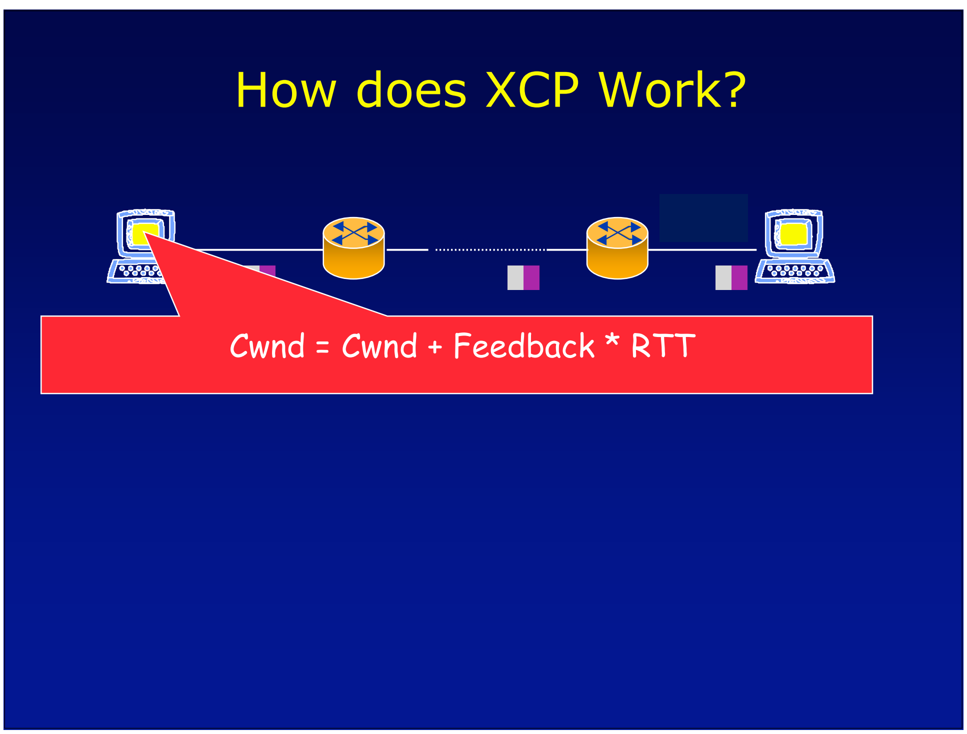
dk@mit.edu

# How does XCP Work?

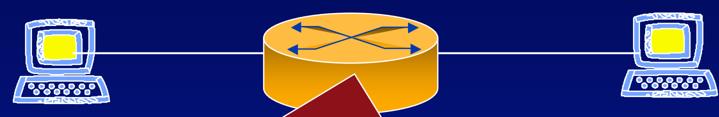



Congestion Header


How does XCP Work?
Round Trip Time
Throughput
Feedback =
- 0.3 packet/sec


How does XCP Work?
Cwnd = Cwnd + Feedback * RTT

# Decouple Congestion Control From BW Allocation Policy



1. Congestion Controller
2. Fairness Controller

# How Does an XCP Router Compute the Feedback?

## Congestion Controller

Goal: Matches input traffic to link capacity & drains the queue

Reaction is prop. to Spare and Queue.
Update every avg. RTT

Algorithm:
Every Avg. RTT,
Aggregate traffic changes by Δ
Δ ~ Spare Bandwidth
Δ ~ - Queue Size
→ quick response

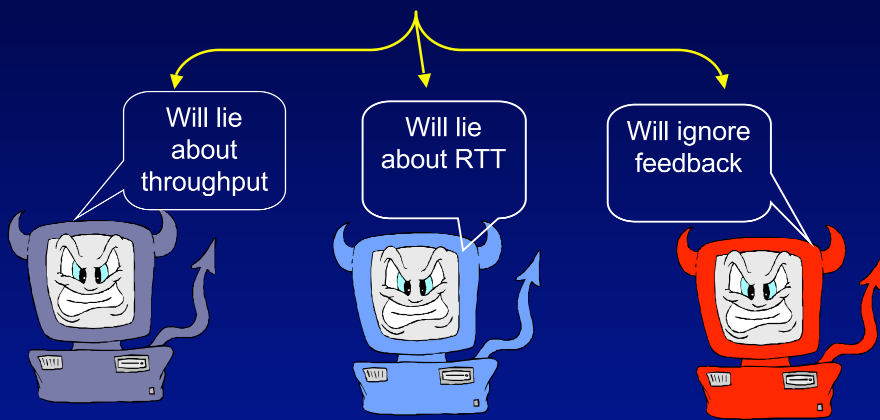
## Fairness Controller

Goal: Divides Δ between flows to converge to fairness

AIMD
De-allocation is prop. to throughput

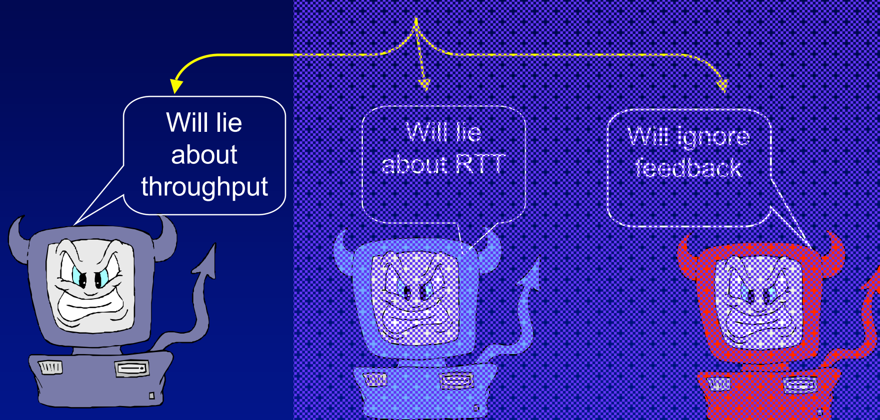
Algorithm:
If Δ > 0 ⇒ Divide Δ equally between flows
If Δ < 0 ⇒ Divide Δ between flows proportionally to their current rates
(shown to achieve Fairness [Jain])

## What if sources are malicious?




Can combine malicious attitudes!

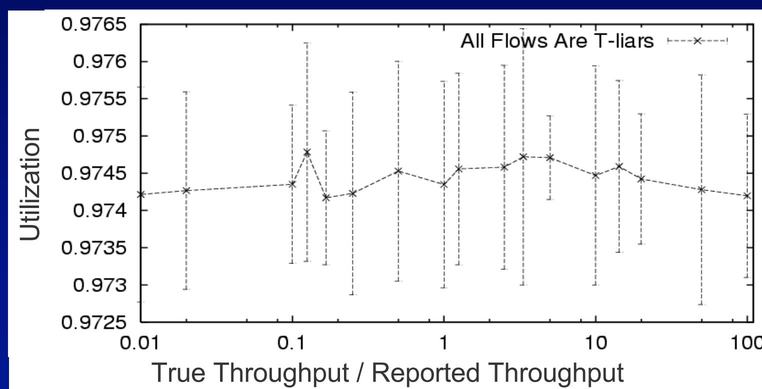
## What if sources are malicious?



## Does lying about throughput affect utilization?

No. congestion controller makes the aggregate increase/decrease proportionally to the spare and the queue

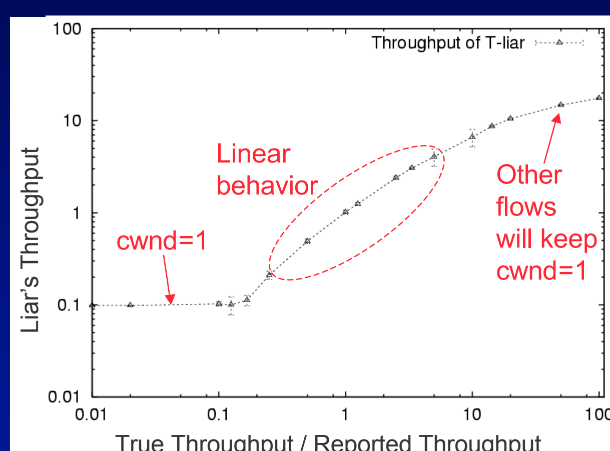
Simulated 20 flows all lying about their throughput:



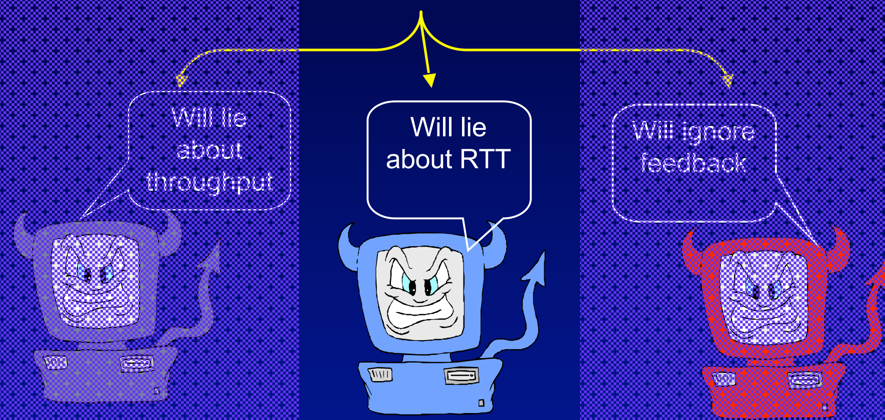

## Does lying about throughput affect fairness?

Yes. Liar simulates multiple flows → gets multiple fair shares

Simulated one liars with 20 good flows



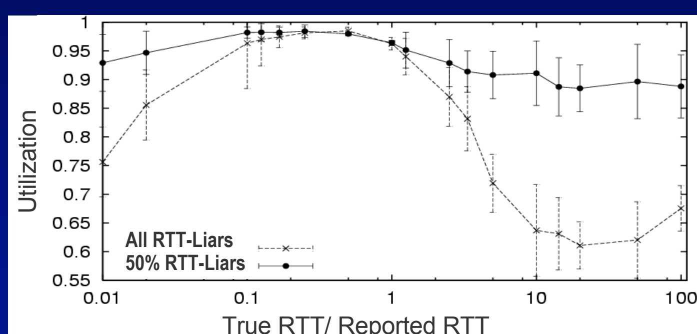
## What if sources are malicious?




## Does lying about RTT affect utilization?

Yes. congestion controller makes decision every avg. RTT
The liar can confuse the congestion controller!

Simulated 20 flows lying about RTT:

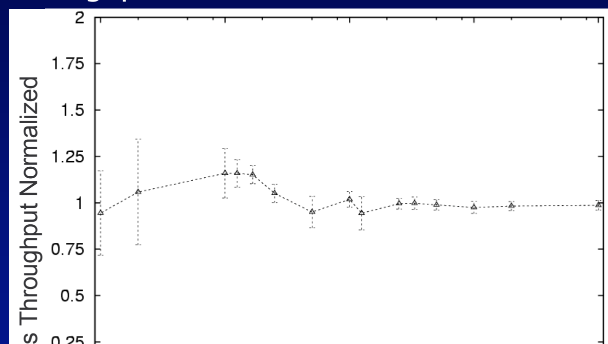



Performance stays good when a limited number of flows lie about their RTTs

## Does lying about RTT affect fairness?

No. It increases variance in the fair share but does not increase absolute throughput much

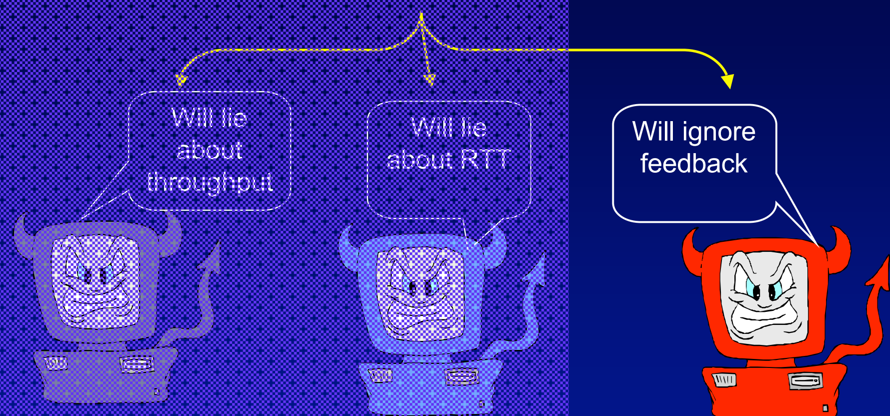
Simulated one liars with 20 good flows



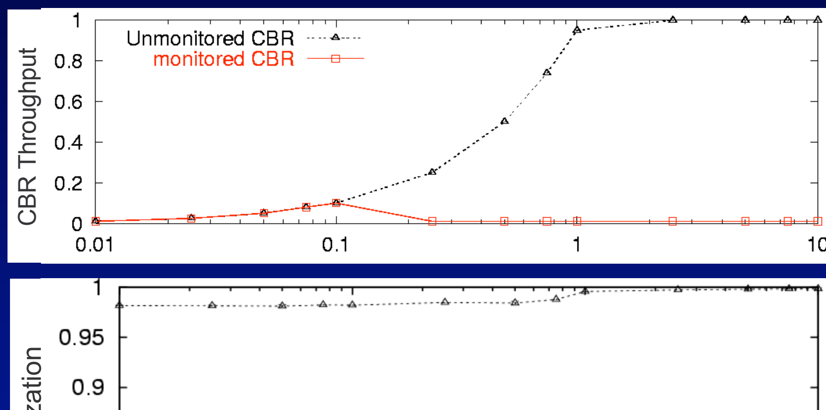

a) No big incentive for senders to lie about RTT

b) Can improve robustness to RTT-lies by making decisions every 100 ms rather than every Avg. RTT, but that would reduce responsiveness

## What if sources are malicious?



## When a flow ignores the feedback, the router tries to balance the utilization given the leftover capacity




**Solution: Sample & Test**

With probability *p=0.05* sample the flows

Send the flow negative feedback & monitor it for 5 avg. RTTs

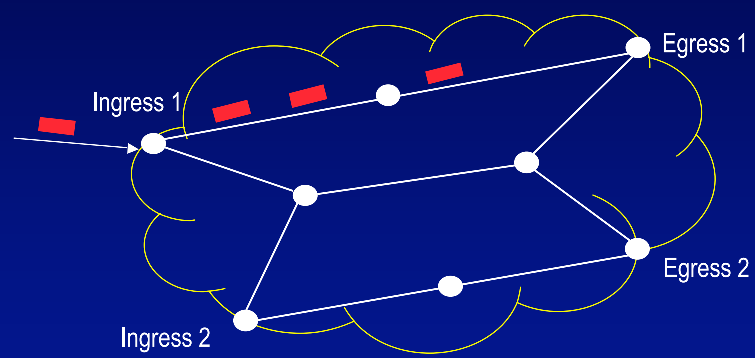
If the flow doesn't react, punish it

---

Next with XCP

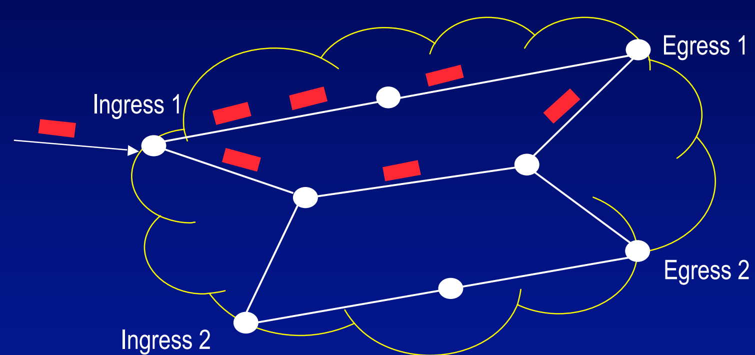
## TeXCP: Using the XCP Framework for Traffic Engineering

# Intra-Domain TE

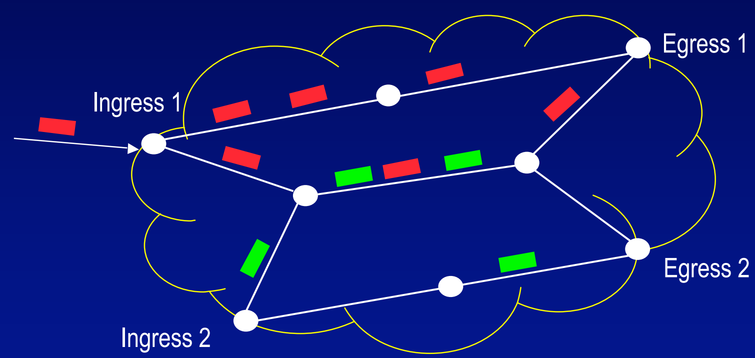



Each ingress-egress pair has traffic demands

# Intra-Domain TE



# Intra-Domain TE




Multi-path routing to minimize max utilization

# Why Minimize the Max. Utilization ?

- Removes hot spots
- Deals with link failures
- Deals with unpredictable traffic spikes, flash crowds, and worm spreading

Prior work uses *offline* approaches (e.g., OSPF optimal weight setting)

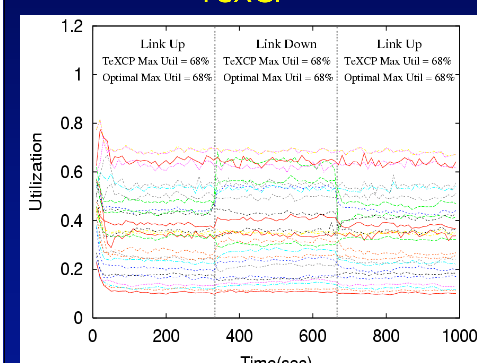
# TeXCP: Online In-Network Approach for Minimizing Max Utilization

- Multi-paths between ingress-egress pair
  - Paths are tunnels pinned using MPLS
- Think of ingress-egress tunnels as flows
- Generalize congestion control
  - One path → Multi-paths
  - 100% utilization → Balanced utilization
- Replace congestion header with occasional control packets on the slow path
  - Easy to deploy in router software
  - Doesn't assume XCP

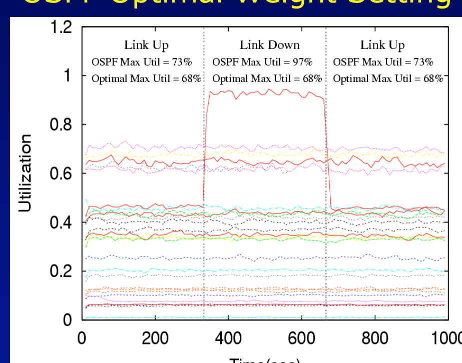
# Reaction to Link Failure

Abilene Topology & Scaled Traffic Matrix

## TeXCP



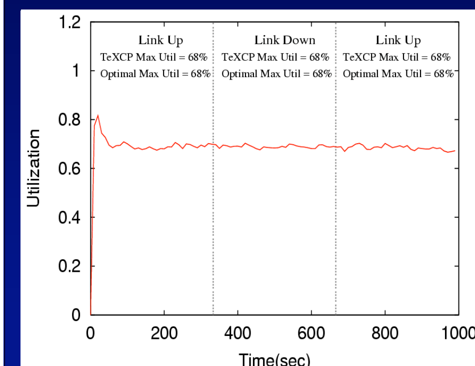

## OSPF Optimal Weight Setting



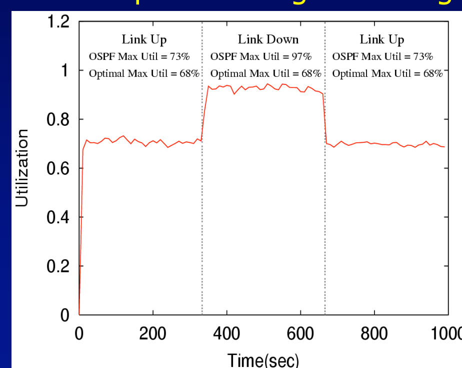
# Reaction to Link Failure

Abilene Topology & Scaled Traffic Matrix

TeXCP




OSPF Optimal Weight Setting




TeXCP reacts quickly and optimally to link failures because it reacts in real-time

# Conclusion

- Lying about a flow's throughput can increase BW share but doesn't affect utilization
- Lying about the RTT can degrade utilization
  - Need a large number of liars to degrade performance → unlikely given that it does not benefit the source
- Ignoring the feedback can result in a larger BW share
  - Deal with it using sample & test
- XCP framework can be used for online in-network traffic engineering
  - Easy to deploy with only changes to the slow path