An experimental study of the learnability of congestion control

Anirudh Sivaraman, Keith Winstein, Pratiksha Thaker,
Hari Balakrishnan

MIT CSAIL

http://web.mit.edu/remy/learnability

August 31, 2014

This talk

- How easy is it to learn a network protocol to achieve a desired goal, despite a mismatched set of assumptions?

This talk

- How easy is it to learn a network protocol to achieve a desired goal, despite a mismatched set of assumptions?
- cf. Learning: “Knowledge acquisition without explicit programming” (Valiant 1984)

Preview of key results

Preview of key results

- Can tolerate mismatched link-rate assumptions

Preview of key results

- Can tolerate mismatched link-rate assumptions
- Need precision about the number of senders

Preview of key results

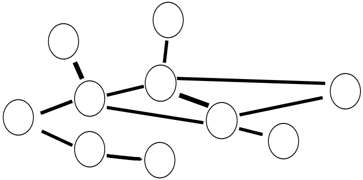
- Can tolerate mismatched link-rate assumptions
- Need precision about the number of senders
- TCP compatibility is a double-edged sword

Preview of key results

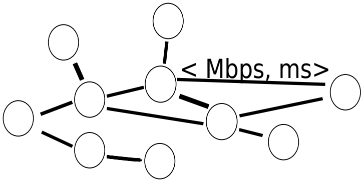
- Can tolerate mismatched link-rate assumptions
- Need precision about the number of senders
- TCP compatibility is a double-edged sword
- Can tolerate mismatch in the # of bottlenecks

Experimental method

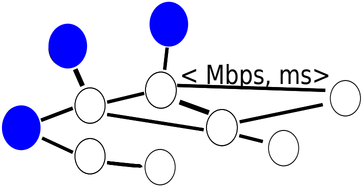
Experimental method



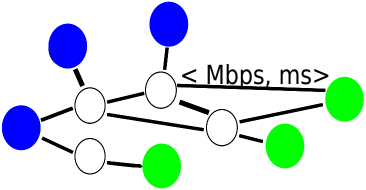
Experimental method



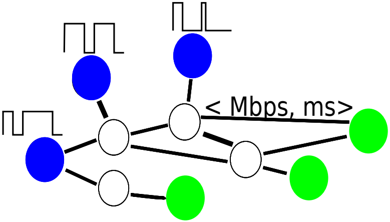
Experimental method



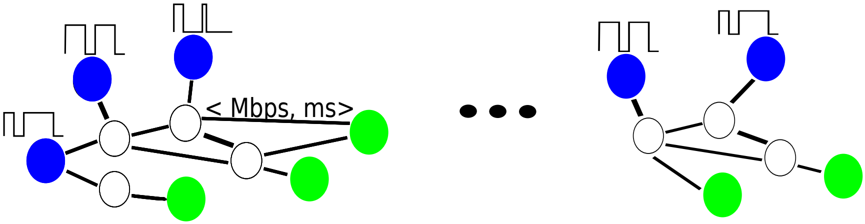
Experimental method



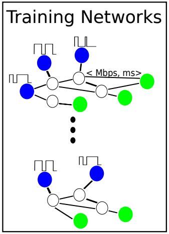
Experimental method



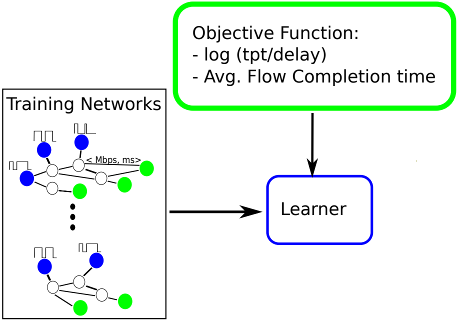
Experimental method



Experimental method



Experimental method

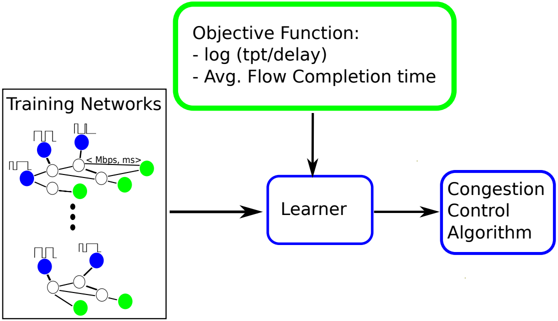


Objective Function:
- log (tpt/delay)
- Avg. Flow Completion time

Training Networks

Learner

Experimental method



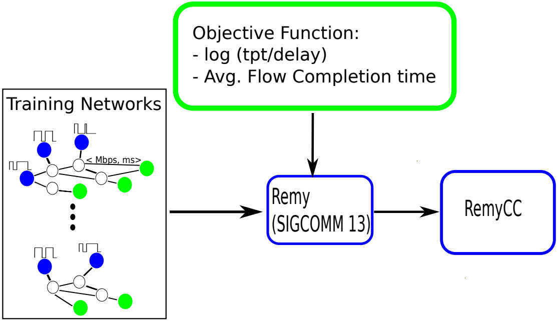
Objective Function:
- log (tpt/delay)
- Avg. Flow Completion time

Training Networks

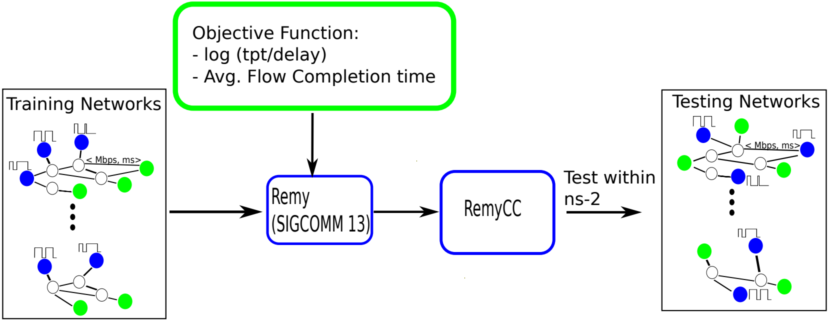
Learner

Congestion Control Algorithm

Experimental method



Experimental method



Objective Function:
- log (tpt/delay)
- Avg. Flow Completion time

Training Networks

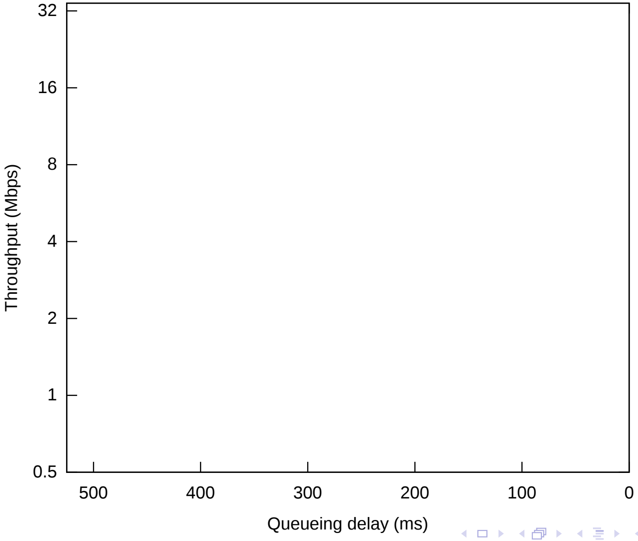
Remy (SIGCOMM 13)

RemyCC

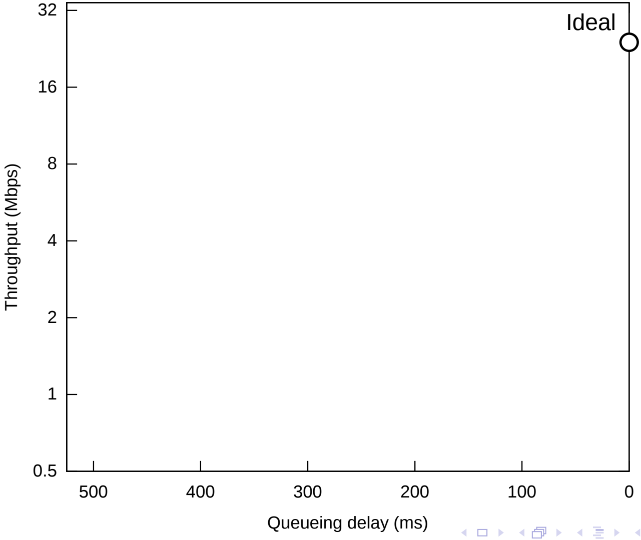
Test within ns-2

Testing Networks

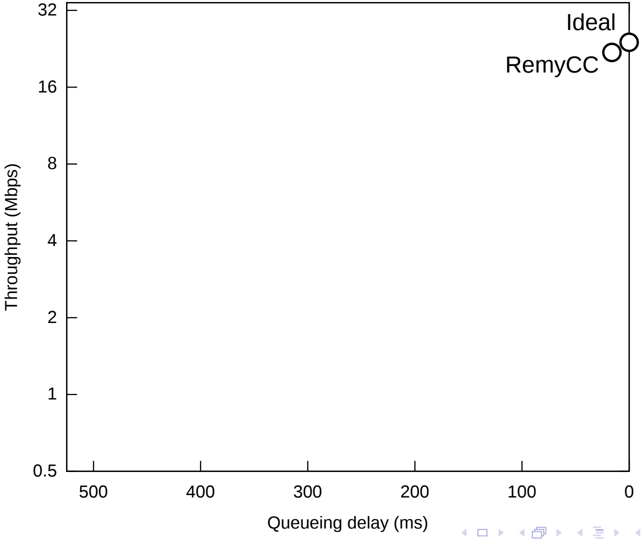
Remy compared with an ideal protocol



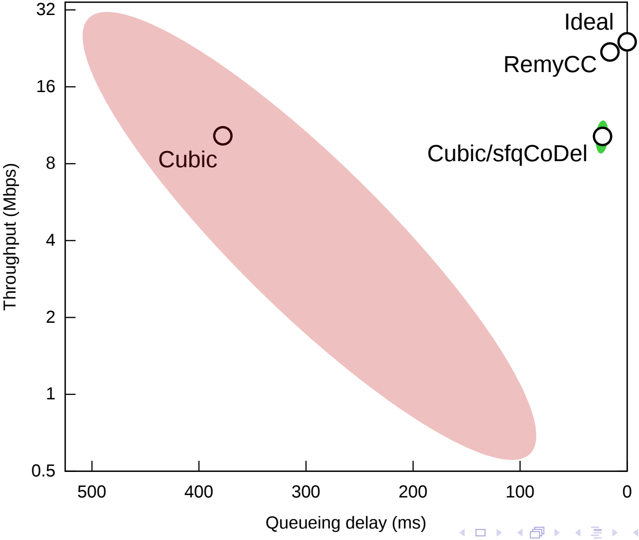
Remy compared with an ideal protocol



Remy compared with an ideal protocol



Remy compared with an ideal protocol



Learning network protocols despite mismatched assumptions

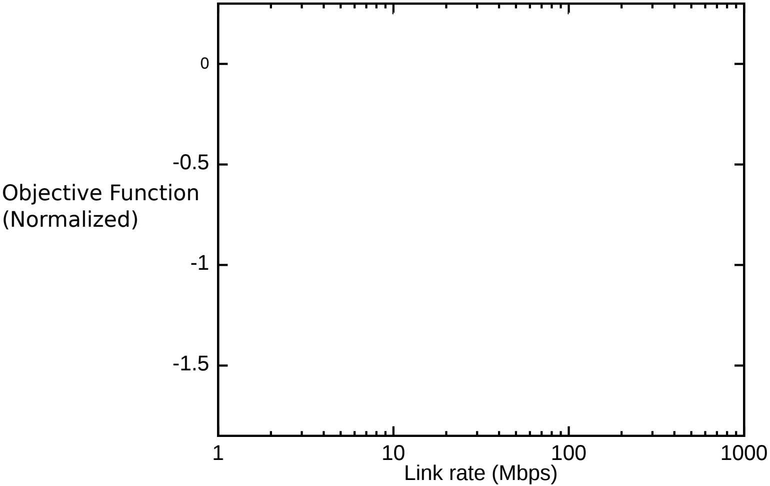
Learning network protocols despite mismatched assumptions

- Is there a tradeoff between operating range and generality in link rates?

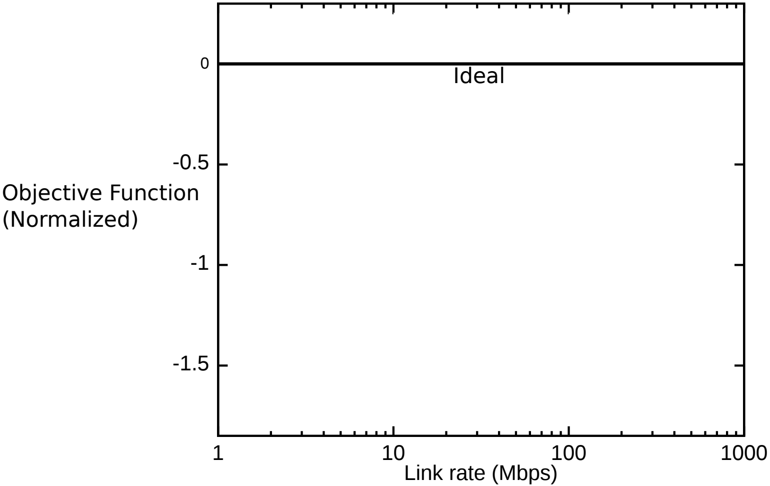
Learning network protocols despite mismatched assumptions

- Is there a tradeoff between operating range and generality in link rates?
- Is there a tradeoff between performance and operating range in link rates?

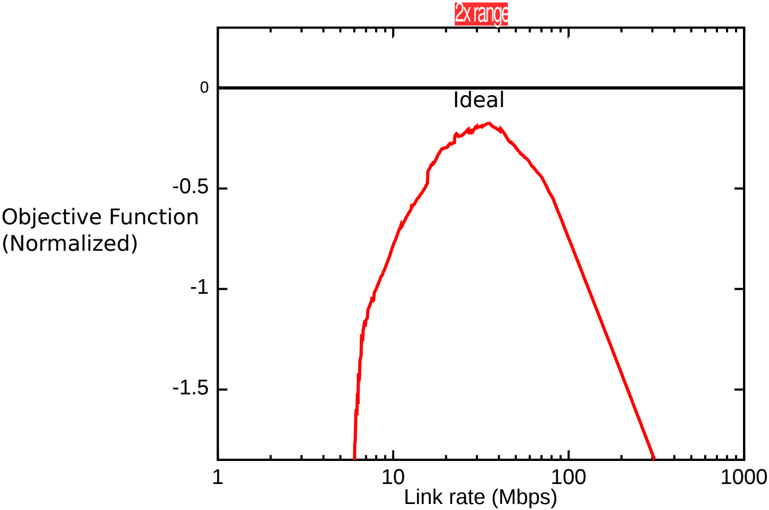
Performance and link-rate operating range



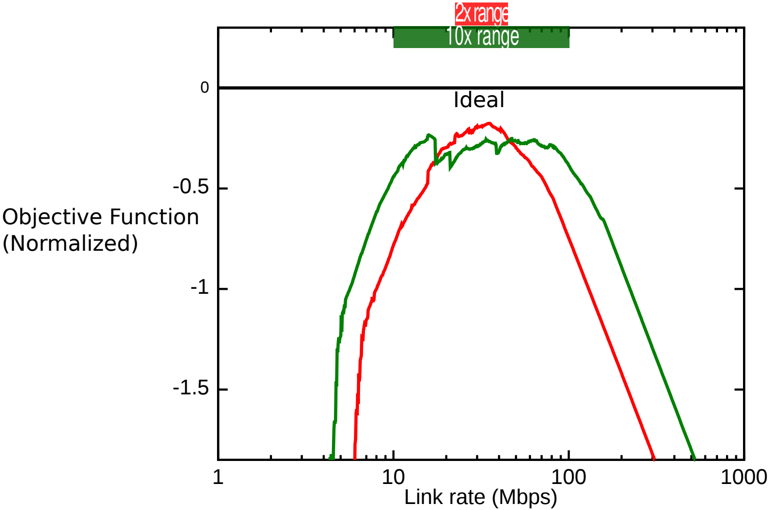
Performance and link-rate operating range



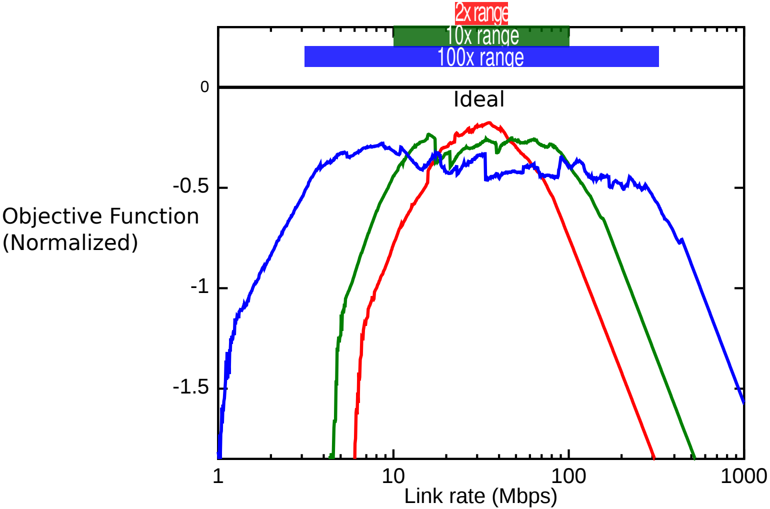
Performance and link-rate operating range



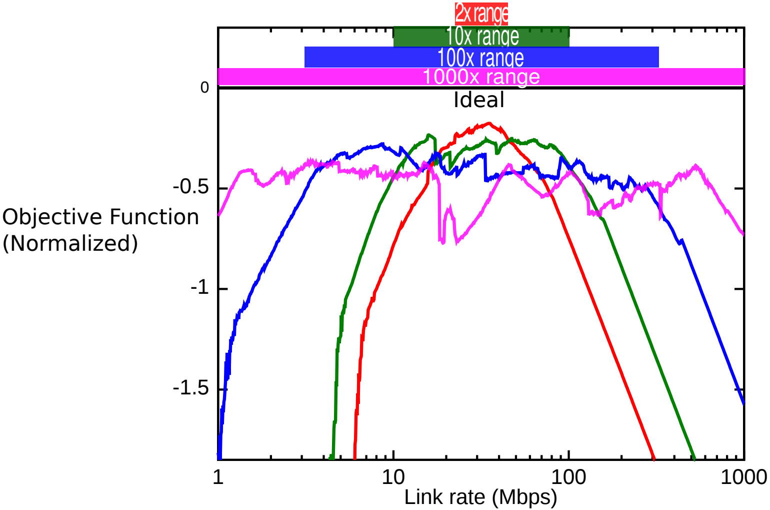
Performance and link-rate operating range



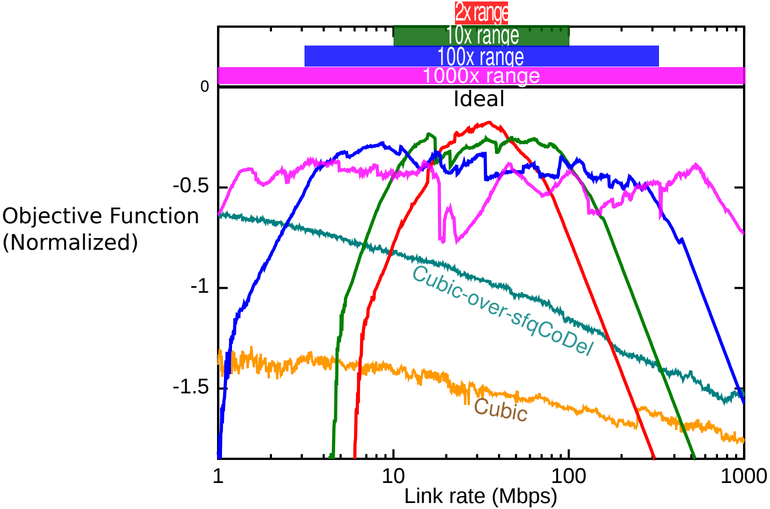
Performance and link-rate operating range



Performance and link-rate operating range



Performance and link-rate operating range



Performance and link-rate operating range

Performance and link-rate operating range

- Very clear generality vs. operating range tradeoff

Performance and link-rate operating range

- Very clear generality vs. operating range tradeoff
- Only weak evidence of a performance vs. operating range tradeoff

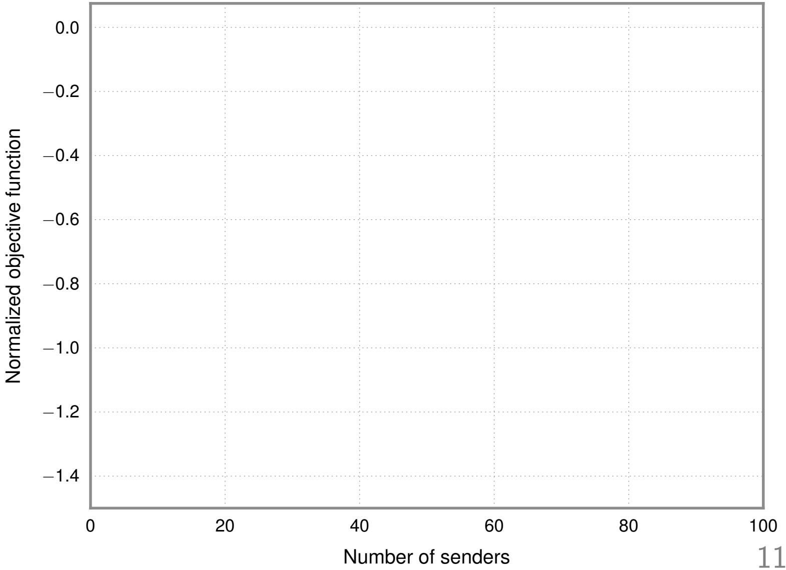
Performance and link-rate operating range

- Very clear generality vs. operating range tradeoff
- Only weak evidence of a performance vs. operating range tradeoff
- Possible to design a forwards-comptabible protocol handling a wide range in link rates

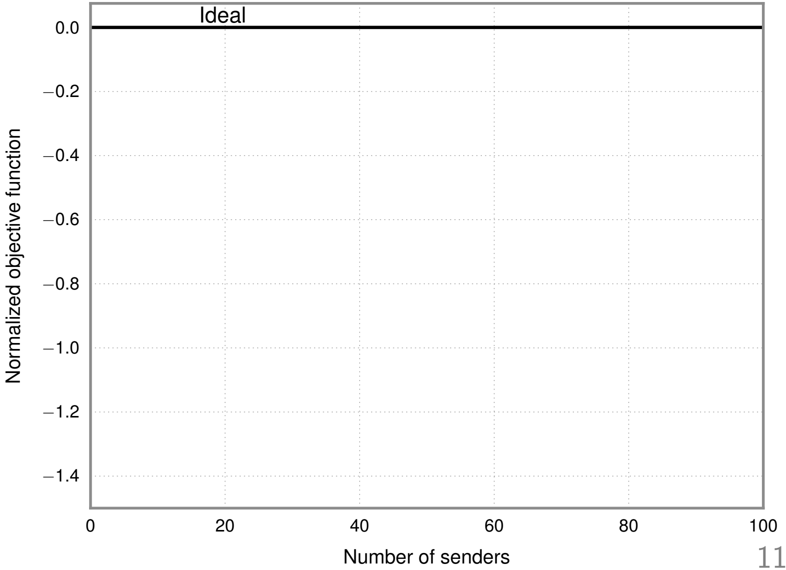
Learning network protocols despite mismatched assumptions

Can we learn a protocol that performs well both when there are few senders and when there are many senders?

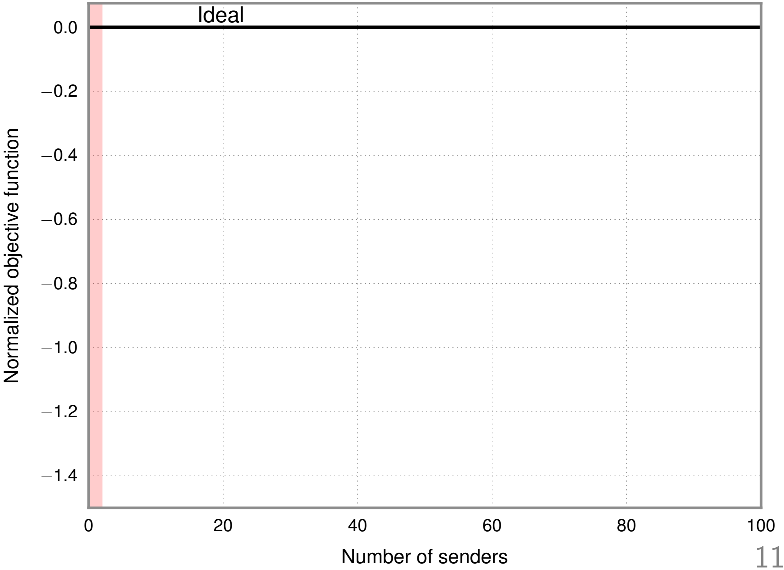
Imperfections in the number of senders



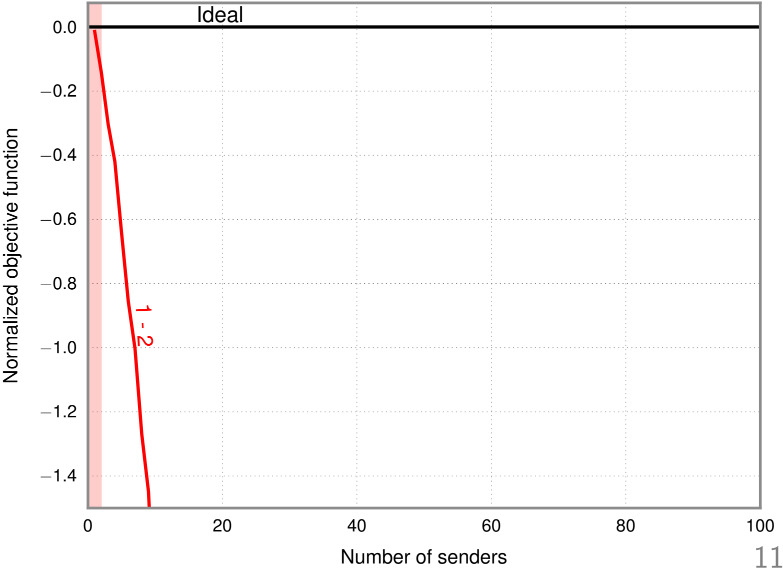
Imperfections in the number of senders



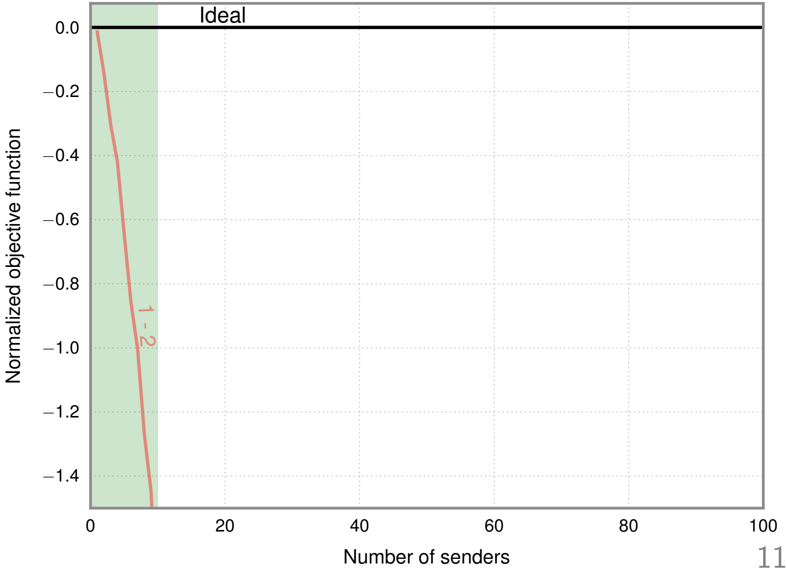
Imperfections in the number of senders



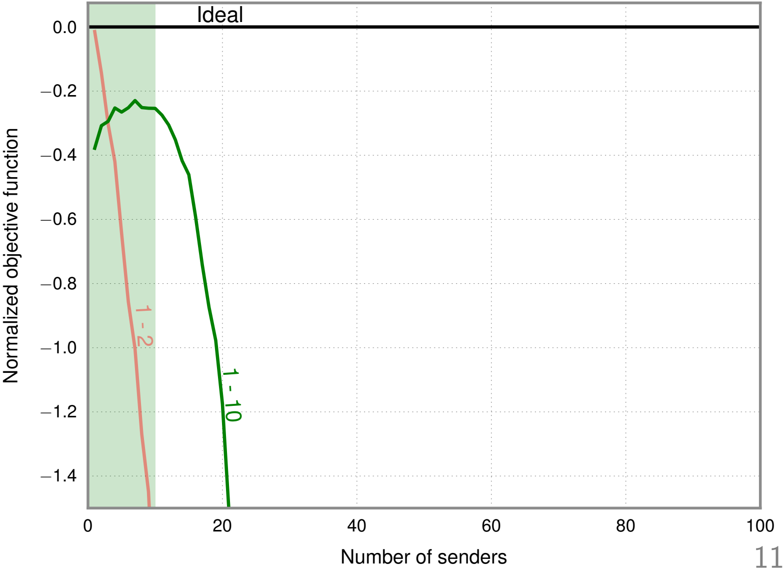
Imperfections in the number of senders



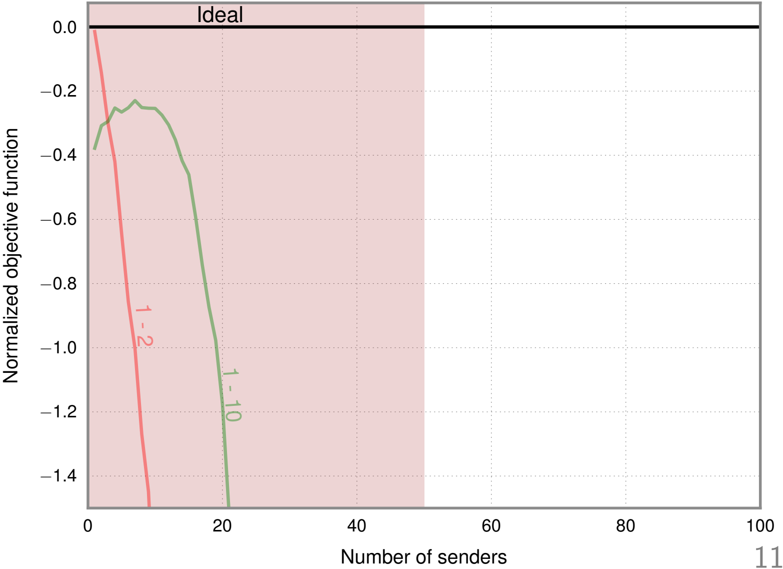
Imperfections in the number of senders



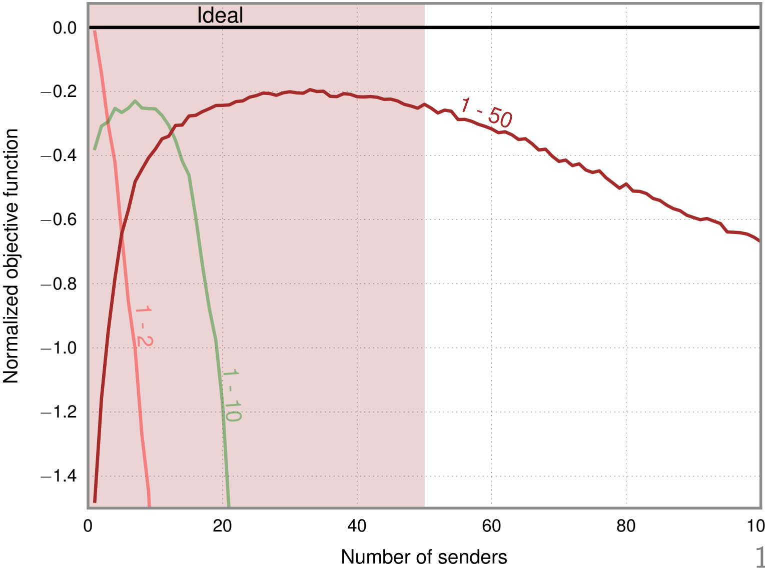
Imperfections in the number of senders



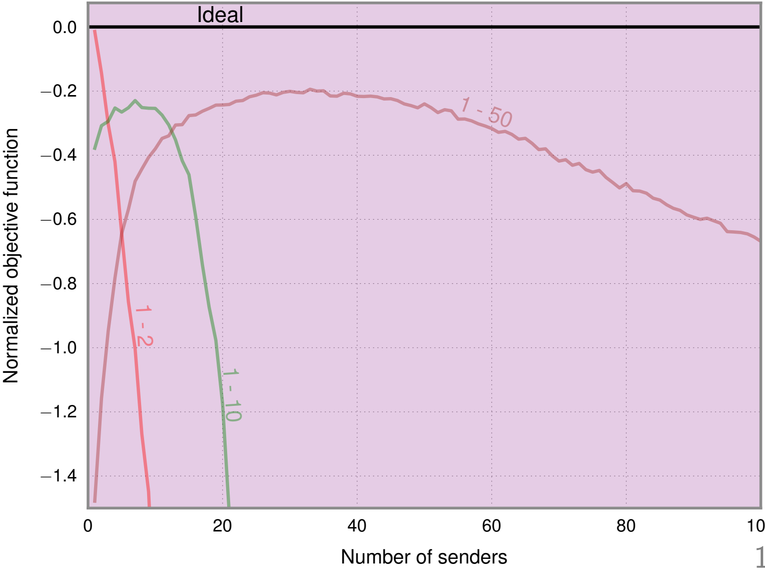
Imperfections in the number of senders



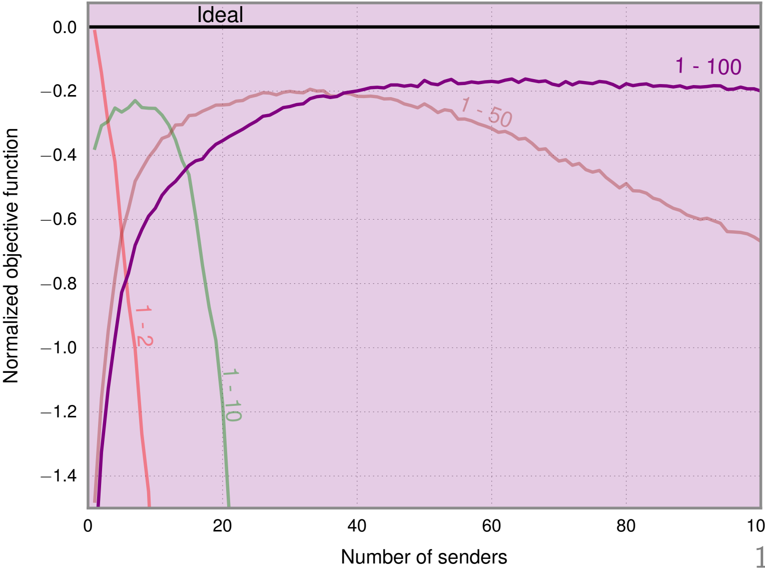
Imperfections in the number of senders



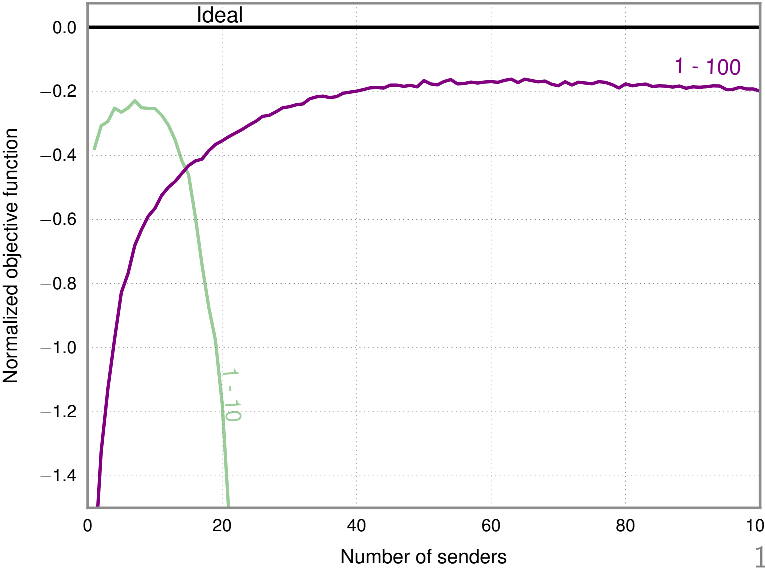
Imperfections in the number of senders



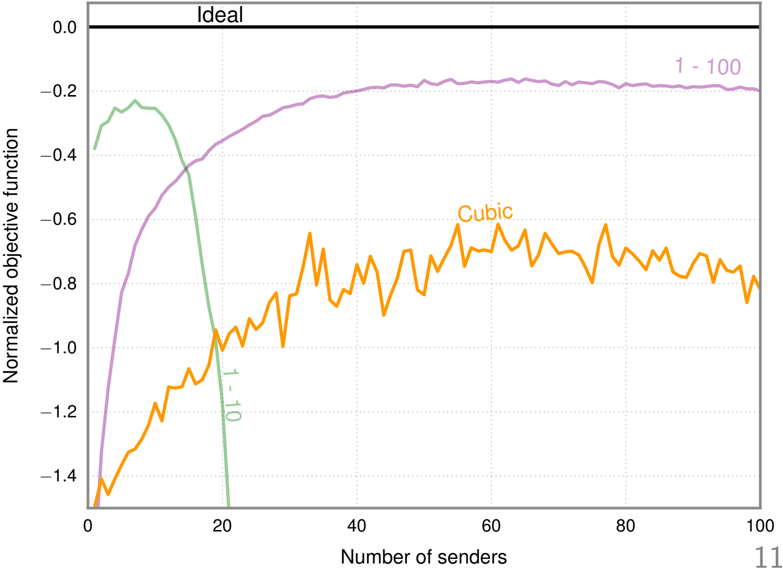
Imperfections in the number of senders



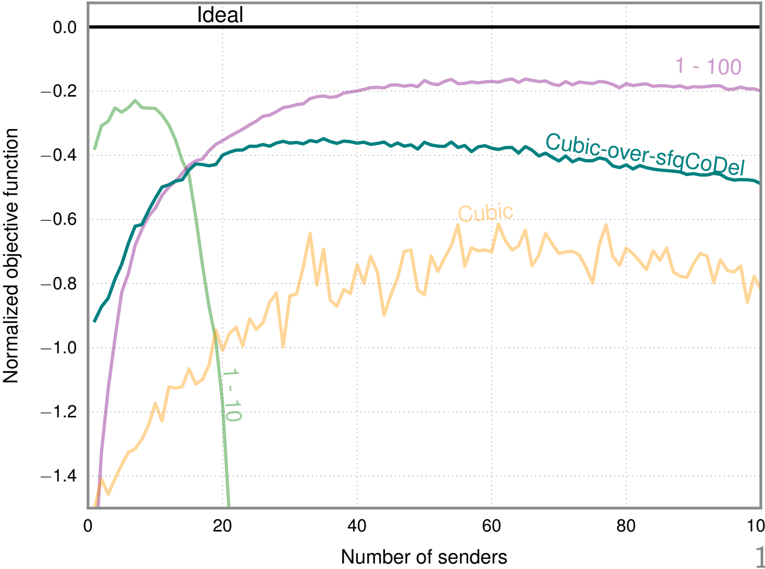
Imperfections in the number of senders



Imperfections in the number of senders



Imperfections in the number of senders



Imperfections in the number of senders

Tradeoff between performance with few senders and performance with many senders

Learning network protocols despite mismatched assumptions

What are the costs and benefits of learning a new protocol that shares fairly with a legacy sender?

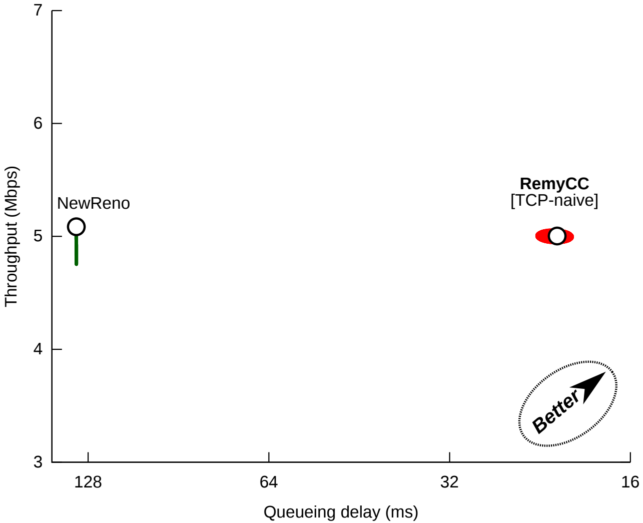
Imperfect assumptions about the nature of other senders

- **TCP-Aware** RemyCC: Contends with:
    - **TCP-Aware** RemyCC half the time
    - TCP NewReno half the time.

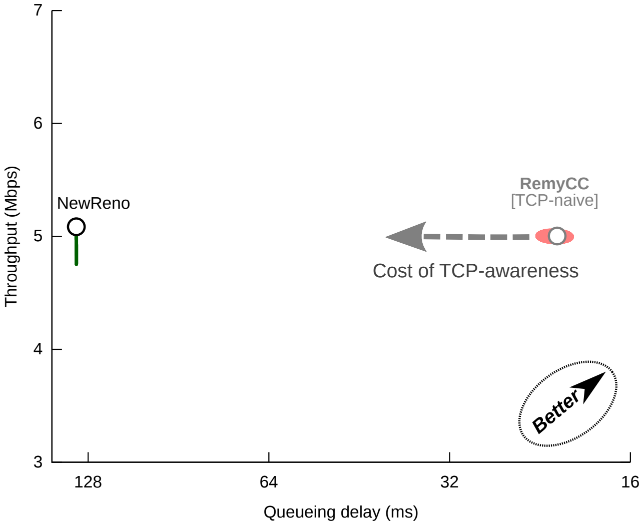
Imperfect assumptions about the nature of other senders

- **TCP-Aware** RemyCC: Contends with:
  - **TCP-Aware** RemyCC half the time
  - TCP NewReno half the time.
- **TCP-Naive** RemyCC: Contends with:
  - **TCP-Naive** RemyCC all the time

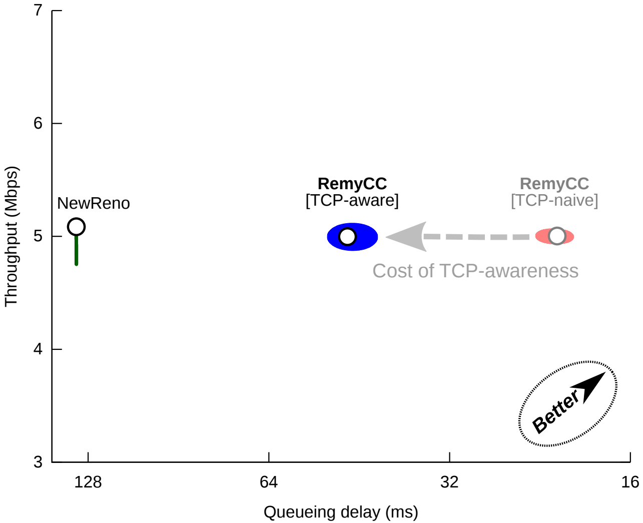
RemyCC competing against itself



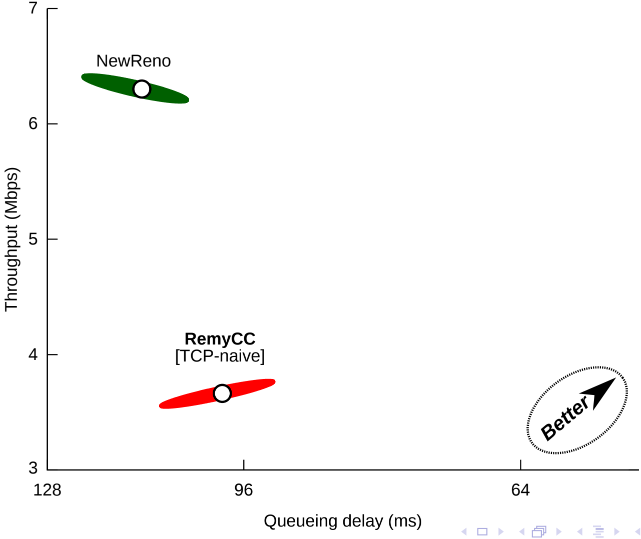
RemyCC competing against itself



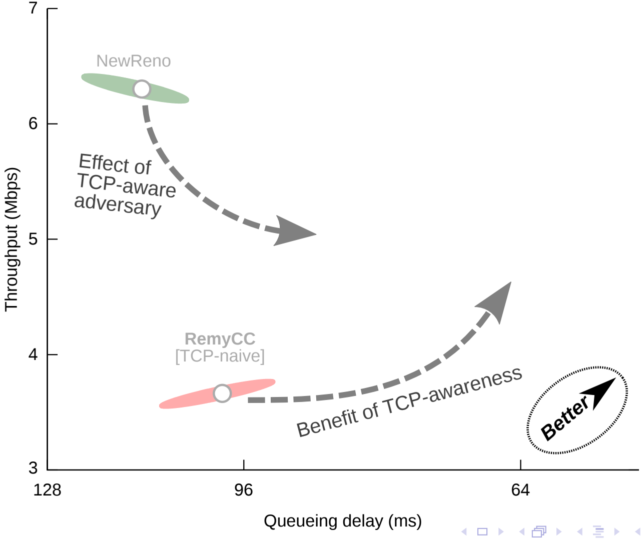
RemyCC competing against itself



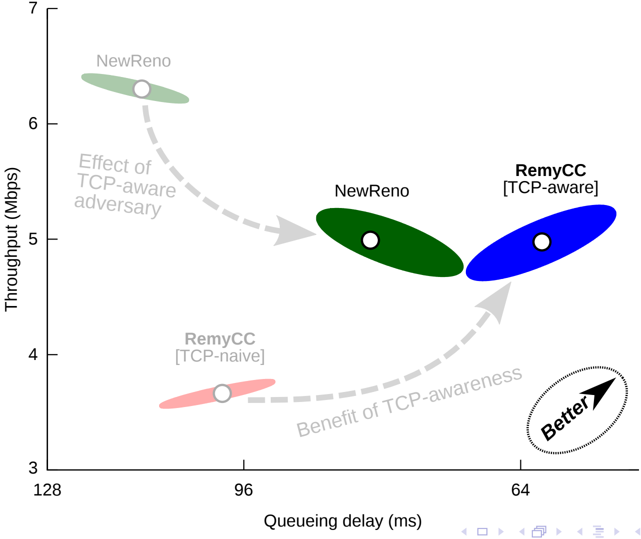
RemyCC competing against TCP NewReno



RemyCC competing against TCP NewReno



RemyCC competing against TCP NewReno



RemyCC competing against TCP NewReno

TCP awareness benefits you when needed, costs if you don't

Caveats

Caveats

- Remy as a proxy for an optimal learner

Caveats

- Remy as a proxy for an optimal learner
- Results may change with better learners

Caveats

- Remy as a proxy for an optimal learner
- Results may change with better learners
- Negative results may no longer hold

The learnability of congestion control

The learnability of congestion control

- Can tolerate mismatched link-rate assumptions

The learnability of congestion control

- Can tolerate mismatched link-rate assumptions
- Need precision about the number of senders

The learnability of congestion control

- Can tolerate mismatched link-rate assumptions
- Need precision about the number of senders
- TCP compatibility is a double-edged sword

The learnability of congestion control

- Can tolerate mismatched link-rate assumptions
- Need precision about the number of senders
- TCP compatibility is a double-edged sword
- Can tolerate mismatch in the # of bottlenecks

The learnability of congestion control

- Can tolerate mismatched link-rate assumptions
- Need precision about the number of senders
- TCP compatibility is a double-edged sword
- Can tolerate mismatch in the # of bottlenecks
- Ongoing work in using findings:

The learnability of congestion control

- Can tolerate mismatched link-rate assumptions
- Need precision about the number of senders
- TCP compatibility is a double-edged sword
- Can tolerate mismatch in the # of bottlenecks
- Ongoing work in using findings:
    - improve Google's datacenter transport

The learnability of congestion control

- Can tolerate mismatched link-rate assumptions
- Need precision about the number of senders
- TCP compatibility is a double-edged sword
- Can tolerate mismatch in the # of bottlenecks
- Ongoing work in using findings:
  - improve Google's datacenter transport
  - user-space implementation of RemyCC

The learnability of congestion control

- Can tolerate mismatched link-rate assumptions
- Need precision about the number of senders
- TCP compatibility is a double-edged sword
- Can tolerate mismatch in the # of bottlenecks
- Ongoing work in using findings:
  - improve Google's datacenter transport
  - user-space implementation of RemyCC
- http://web.mit.edu/remy/learnability

Backup slides

The Remy protocol synthesis procedure

- Protocol: range-based rule table from *state* to *action*

The Remy protocol synthesis procedure

- Protocol: range-based rule table from *state* to *action*
- State: Congestion signals tracked by the sender
    - s_ewma : EWMA over packet inter-transmit times
    - r_ewma : EWMA over ACK inter-arrival times
    - rtt_ratio: Ratio of RTT to minimum RTT
    - slow_r_ewma: Slower version of s_ewma

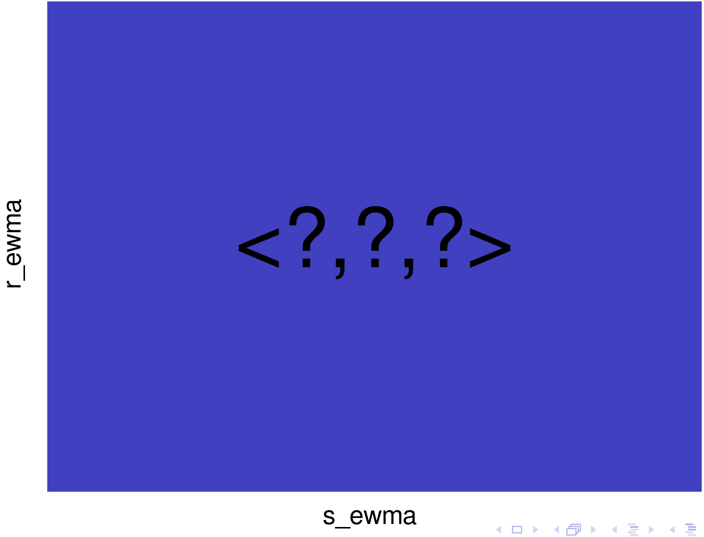
The Remy protocol synthesis procedure

- Protocol: range-based rule table from *state* to *action*
- State: Congestion signals tracked by the sender
    - s_ewma : EWMA over packet inter-transmit times
    - r_ewma : EWMA over ACK inter-arrival times
    - rtt_ratio: Ratio of RTT to minimum RTT
    - slow_r_ewma: Slower version of s_ewma
- Action: modify window, transmission rate
    - Multiplier m to current window
    - Increment c to current window
    - Minimum inter-transmit time.

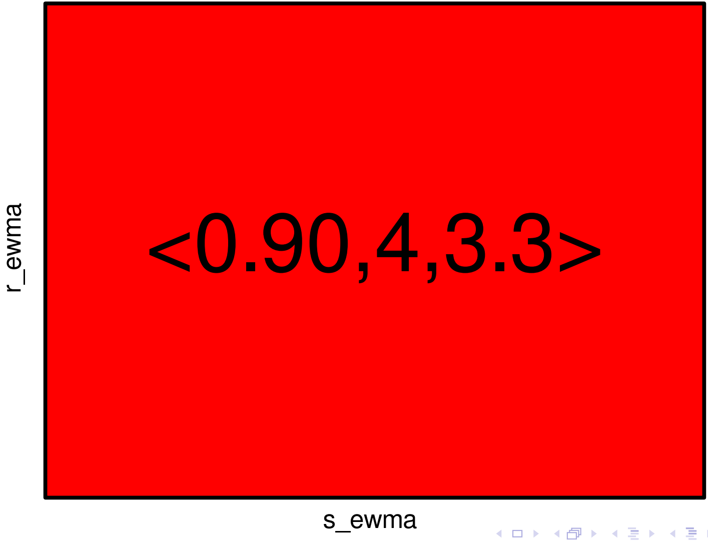
The Remy protocol synthesis procedure

1. Start with one rule: one action for all states
2. Optimize each action to maximize objective
3. Find most used rule
4. Median split that rule based on state usage
5. Repeat 2, 3, and 4 till you converge

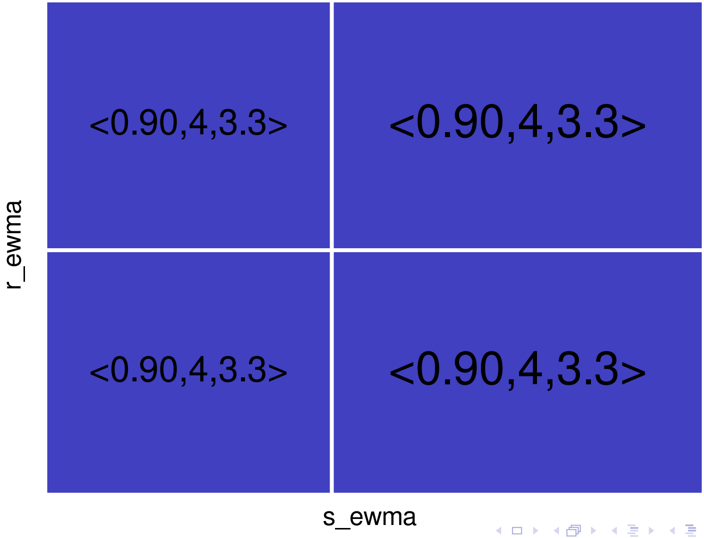
One action for all states. Find the best value.



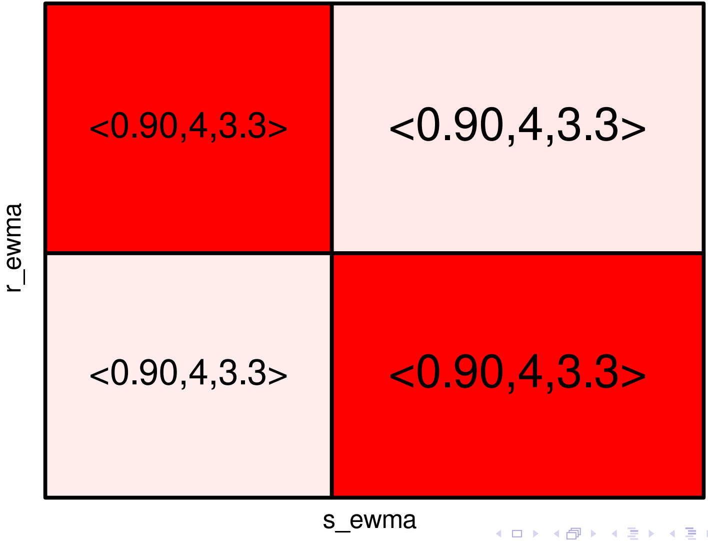
The best (single) action. Now split it on median.



Simulate

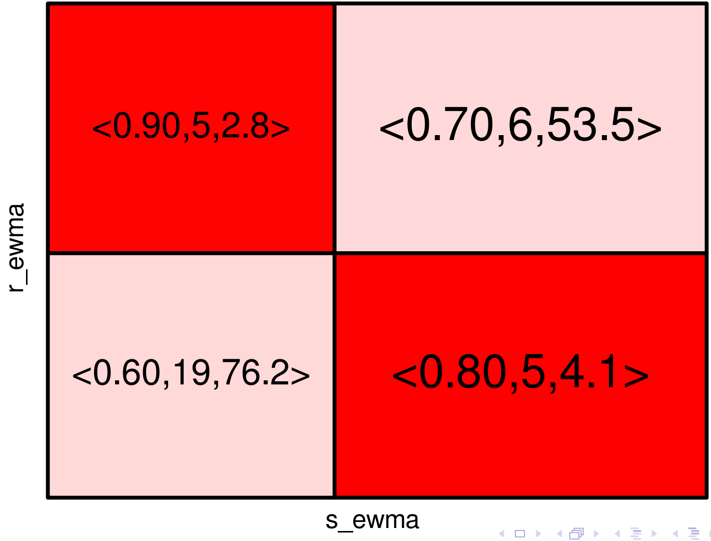


Optimize each of the new actions



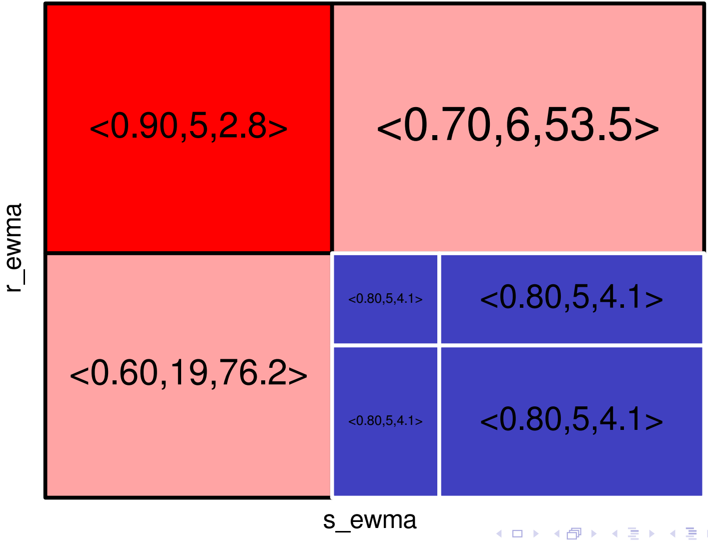
r_ewma	<0.90,4,3.3>	<0.90,4,3.3>
	<0.90,4,3.3>	<0.90,4,3.3>
	s_ewma	

Now split the most-used rule

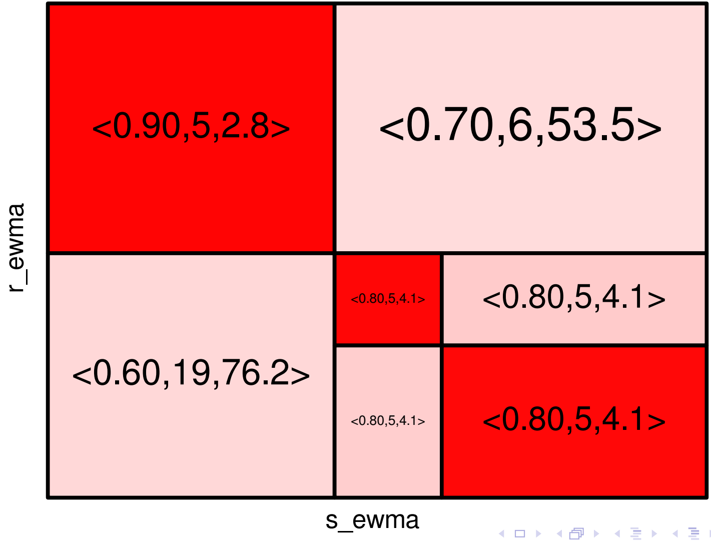


<0.90,5,2.8>	<0.70,6,53.5>
<0.60,19,76.2>	<0.80,5,4.1>

Simulate


<0.90,5,2.8>
<0.70,6,53.5>
<0.80,5,4.1>
<0.80,5,4.1>
<0.60,19,76.2>
<0.80,5,4.1>
<0.80,5,4.1>
r_ewma
s_ewma

Optimize



<0.90,5,2.8>

<0.70,6,53.5>

<0.80,5,4.1>

<0.80,5,4.1>

<0.60,19,76.2>

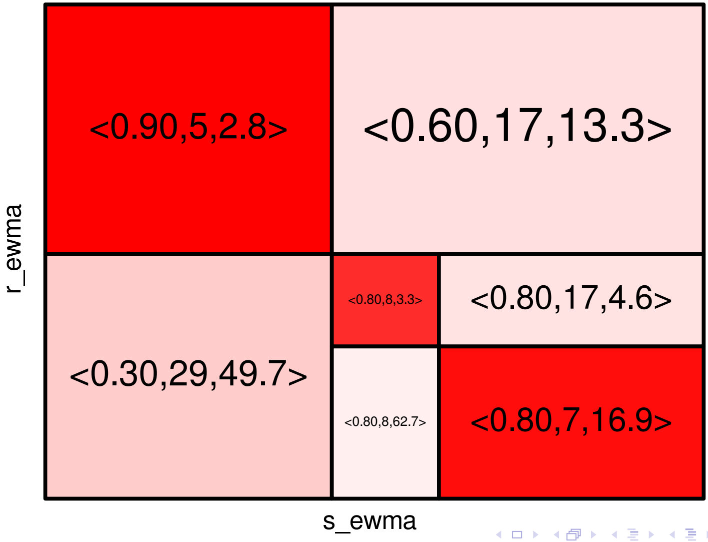
<0.80,5,4.1>

<0.80,5,4.1>

r_ewma

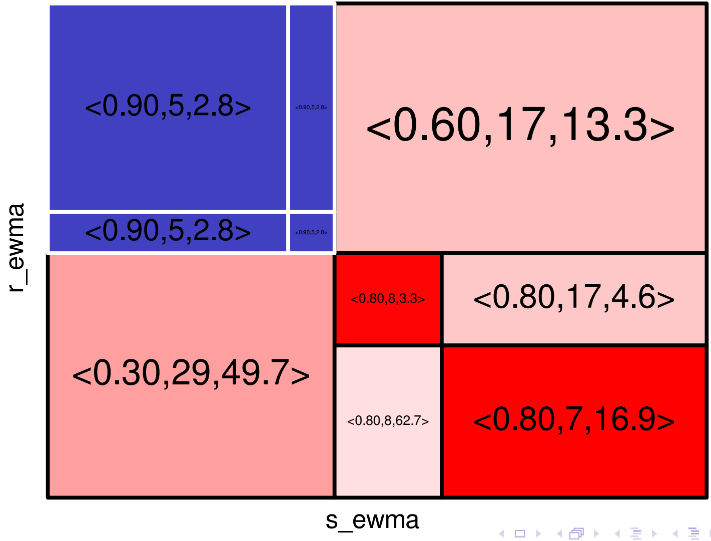
s_ewma

Split



<0.90,5,2.8>
<0.60,17,13.3>
<0.80,8,3.3>
<0.80,17,4.6>
<0.30,29,49.7>
<0.80,8,62.7>
<0.80,7,16.9>
r_ewma
s_ewma

Simulate


<0.90,5,2.8>
<0.90,5,2.8>
<0.60,17,13.3>
<0.90,5,2.8>
<0.90,5,2.8>
<0.80,8,3.3>
<0.80,17,4.6>
<0.30,29,49.7>
<0.80,8,62.7>
<0.80,7,16.9>
r_ewma
s_ewma

Can applications with different objectives coexist?

- Tpt. Sender: A throughput-intensive sender

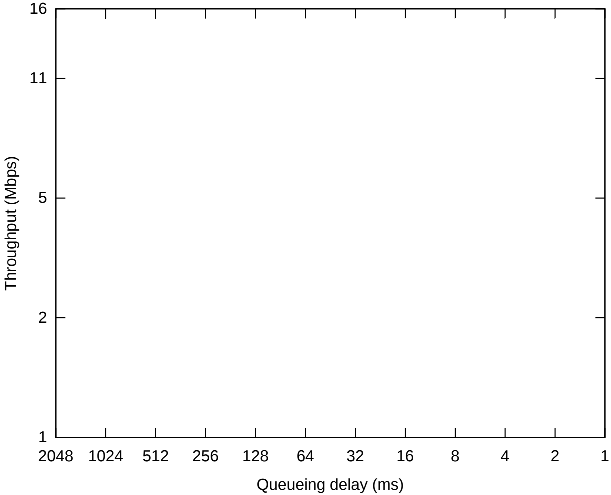
$$log(throughput) - 0.1 * log(delay) \tag{1}$$

- Lat. Sender: A latency-sensitive sender

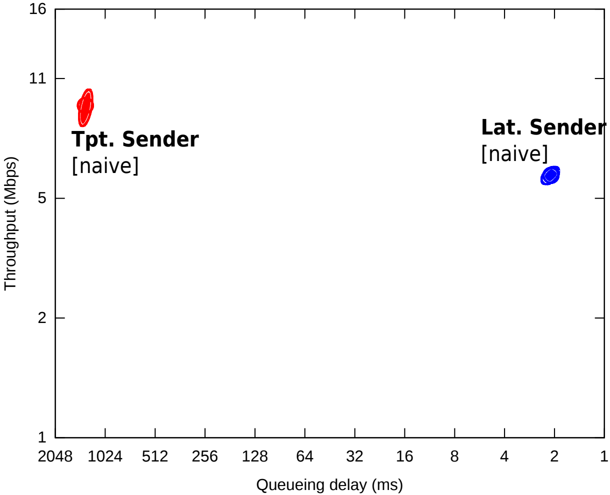
$$log(throughput) - 10.0 * log(delay) \tag{2}$$

- Running over a FIFO queue

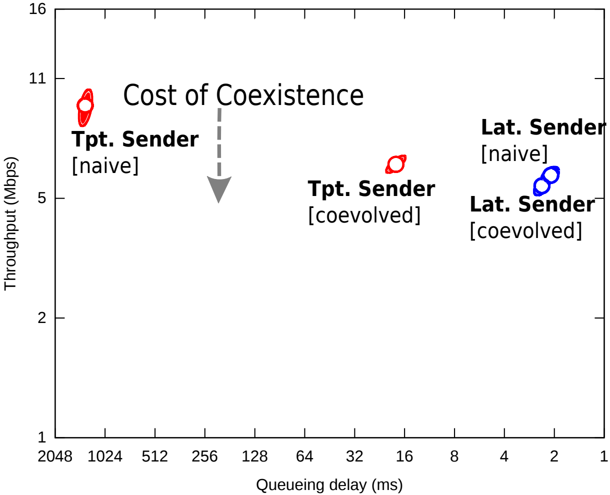
Training for diversity has a cost ...



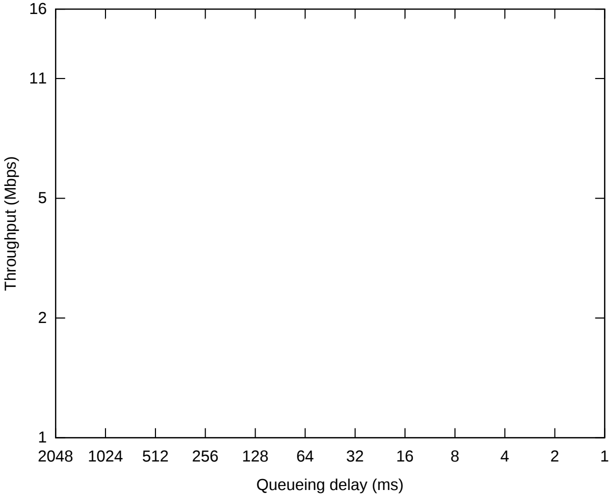
Training for diversity has a cost ...



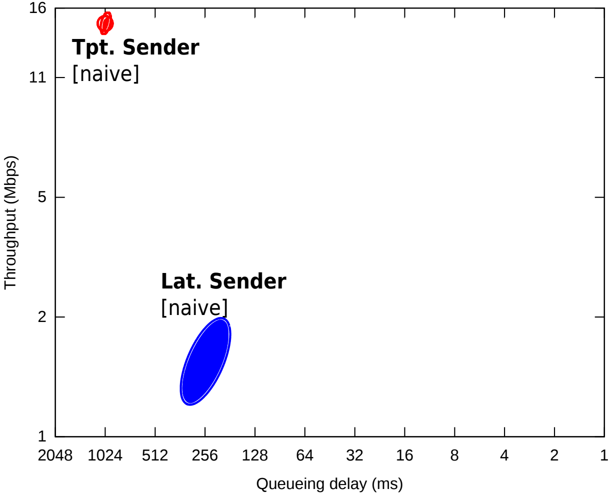
Training for diversity has a cost ...



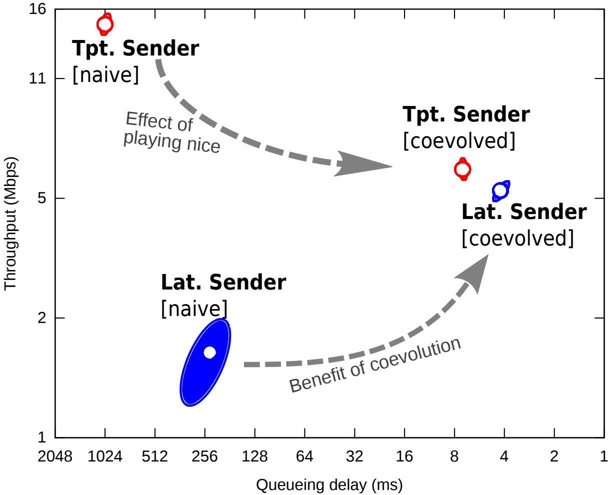
but, benefits the docile sender



but, benefits the docile sender



but, benefits the docile sender


Throughput (Mbps)
Queueing delay (ms)
Tpt. Sender
[naive]
Effect of
playing nice
Tpt. Sender
[coevolved]
Lat. Sender
[coevolved]
Lat. Sender
[naive]
Benefit of coevolution