An experimental study of the learnability of congestion control

Anirudh Sivaraman, Keith Winstein, Pratiksha Thaker, Hari Balakrishnan

MIT CSAIL

May 23, 2014

Designing congestion-control protocols today

- Formulate a mental model of the target network and application workload
- Decide on the protocol's goal
- Design a protocol to achieve this goal on the target network
- Can either be implicit or explicit

But, the model is always wrong!

- Lost throughput due to stochastic loss
- Bufferbloat when queues are incorrectly sized
- Diminished fairness in small-packet regimes
- Incast in datacenters

Our work

- Can we formalize this design process?
- Quantify the consequences of model mismatch?

Approach

- Specify a *training scenario*.
  - Topology
  - Locations of senders and receiver
  - Application workload
  - Buffer size and queuing discipline
- Specify an *objective function*.
- Synthesize protocol automatically.
- Evaluate on a *testing scenario* inside ns-2

Automated protocol synthesis

- Find best protocol, given an imperfect network model.
- Unfortunately, problem is NEXP-complete.

Tractable Attempts at Optimal

- Rely on Remy [?] to produce Tractable Attempts at Optimal (TAO) congestion-control protocols.
- Approaches upper bounds on throughput and lower bounds on delay.

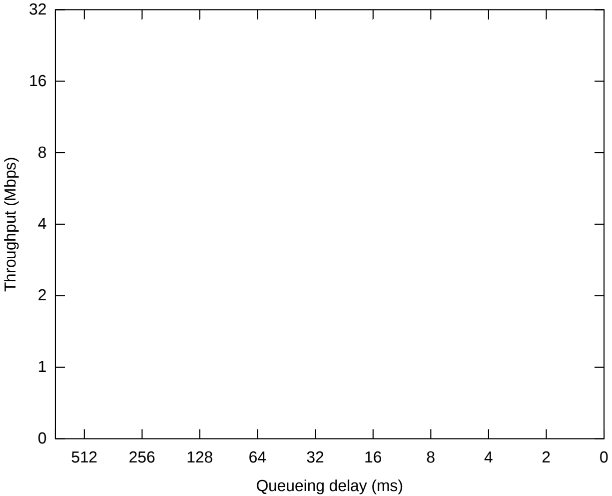
How far off is Remy from the optimal?

- Training scenario:

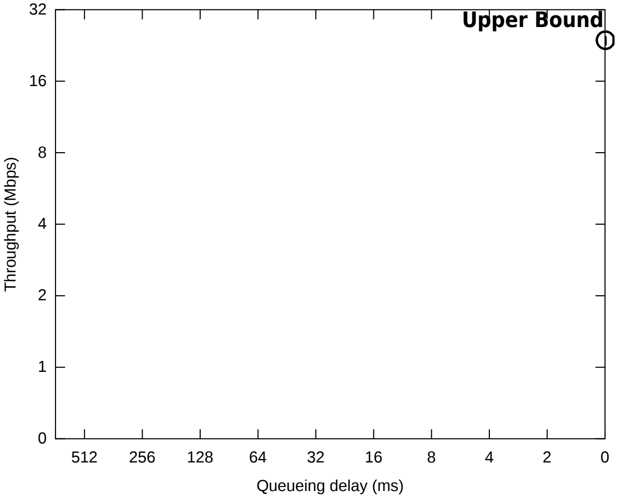
Link speed	32 Mbits/sec
minimum RTT	150 ms
Topology	Dumbbell
Number of senders	2
Workload	1 sec ON/OFF times
Buffer size	5 BDP
Objective function	$\sum$ log(throughput) - log(delay)

- Testing scenario identical to training scenario

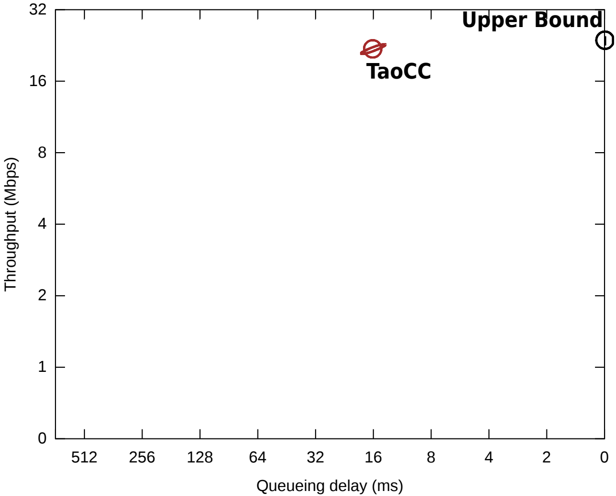
How far off is Remy from the optimal?



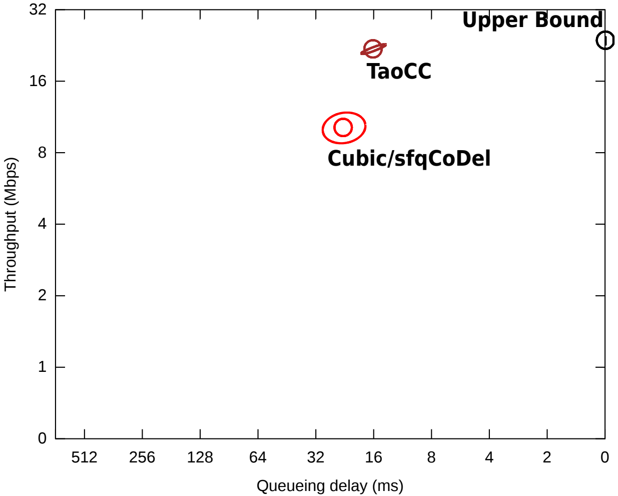
How far off is Remy from the optimal?



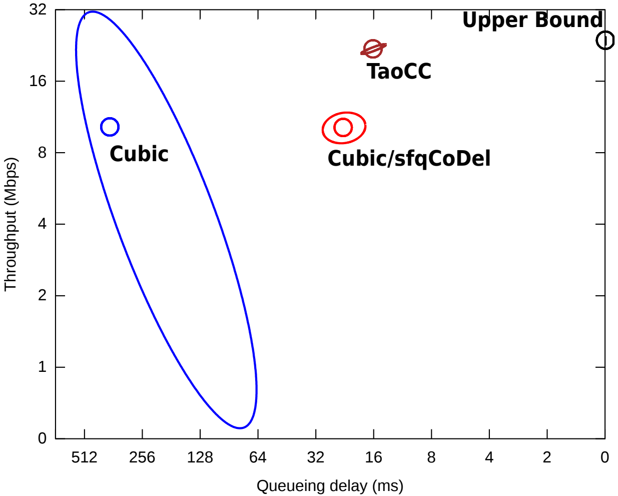
How far off is Remy from the optimal?



How far off is Remy from the optimal?



How far off is Remy from the optimal?

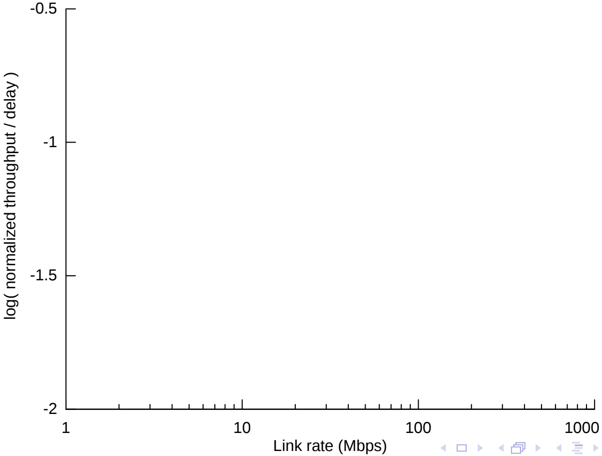


The cost of generality, or forwards-compatibility

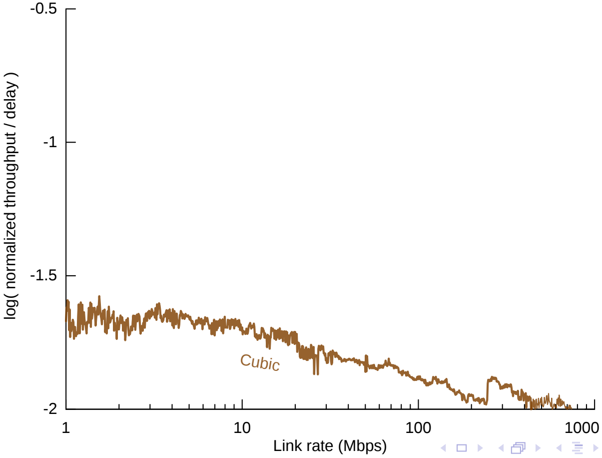
Tao	Link rates	RTT	Senders	ON/OFF time	Topology
1000x	1–1000 Mbps	150 ms	2	1 sec	Dumbbell
100x	3.2–320 Mbps	150 ms	2	1 sec	Dumbbell
10x	10–100 Mbps	150 ms	2	1 sec	Dumbbell
2x	22–44 Mbps	150 ms	2	1 sec	Dumbbell

Table : Training scenarios for forwards-compatibility experiment

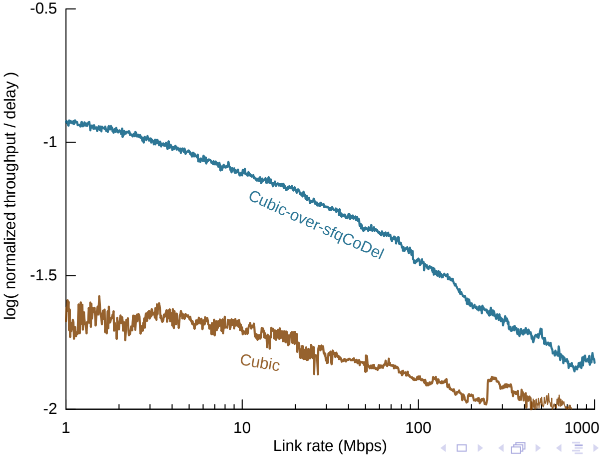
The cost of generality, or forwards-compatibility



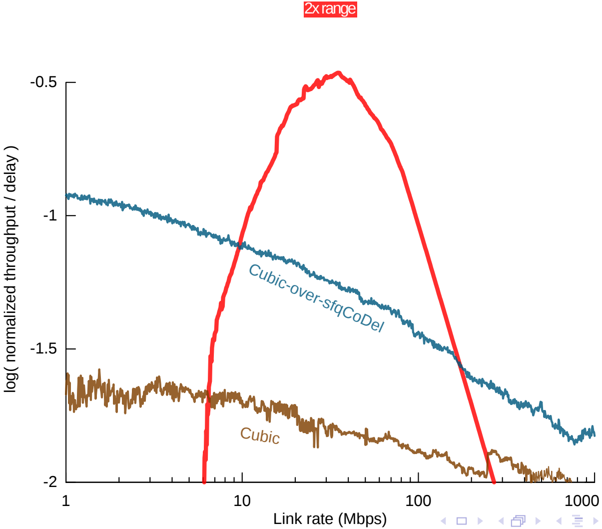
The cost of generality, or forwards-compatibility



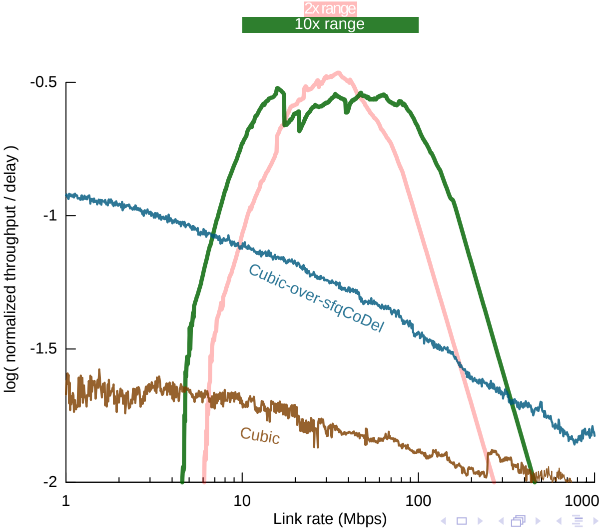
The cost of generality, or forwards-compatibility



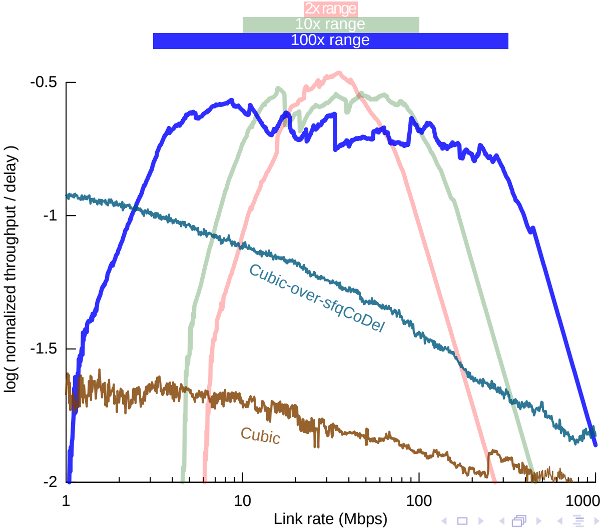
The cost of generality, or forwards-compatibility



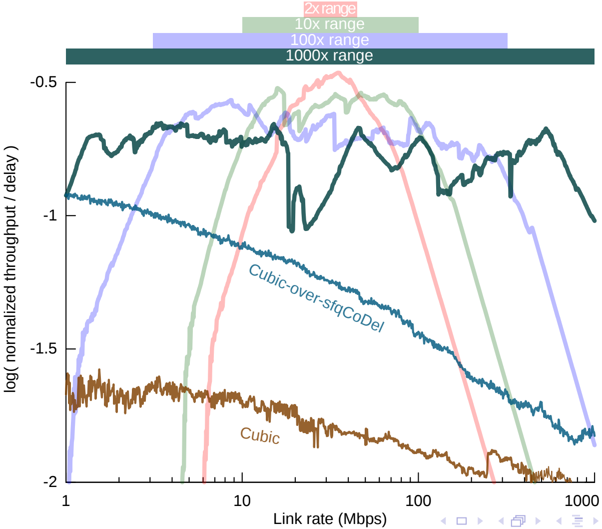
The cost of generality, or forwards-compatibility



The cost of generality, or forwards-compatibility

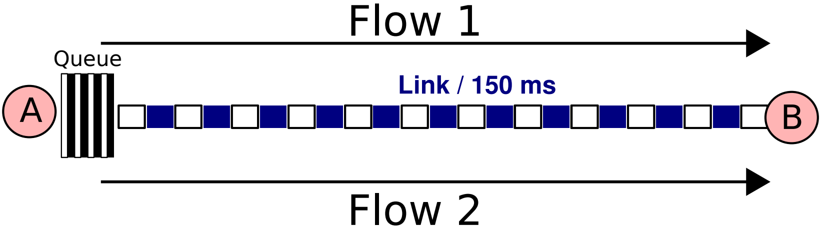


The cost of generality, or forwards-compatibility



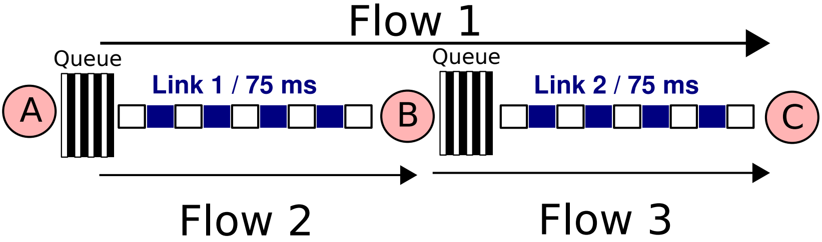
When the model is wrong about the topology

One bottleneck

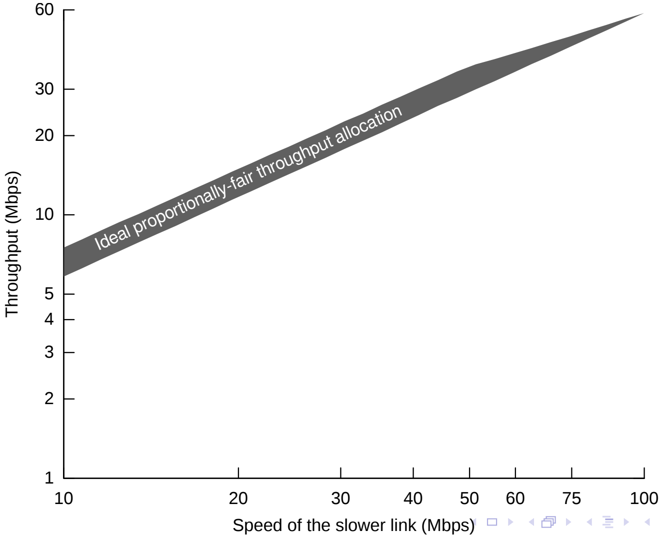


When the model is wrong about the topology

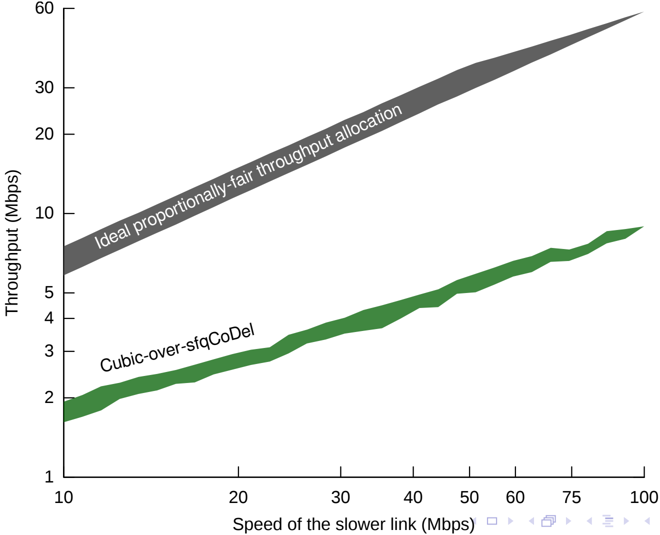
Two bottlenecks



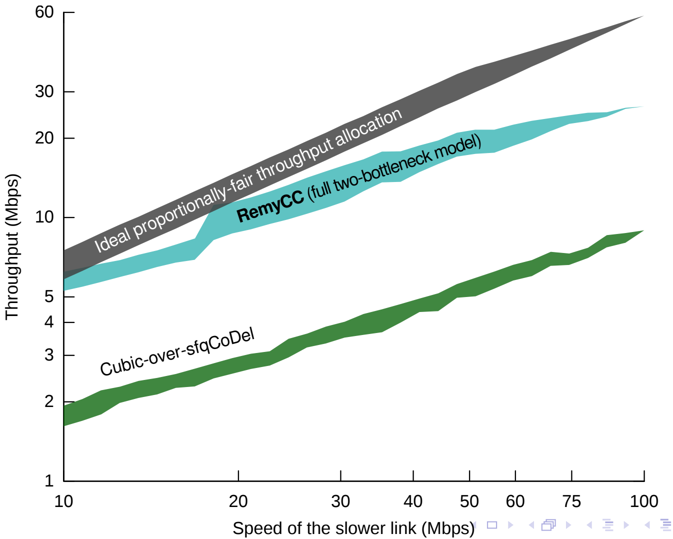
When the model is wrong about the topology



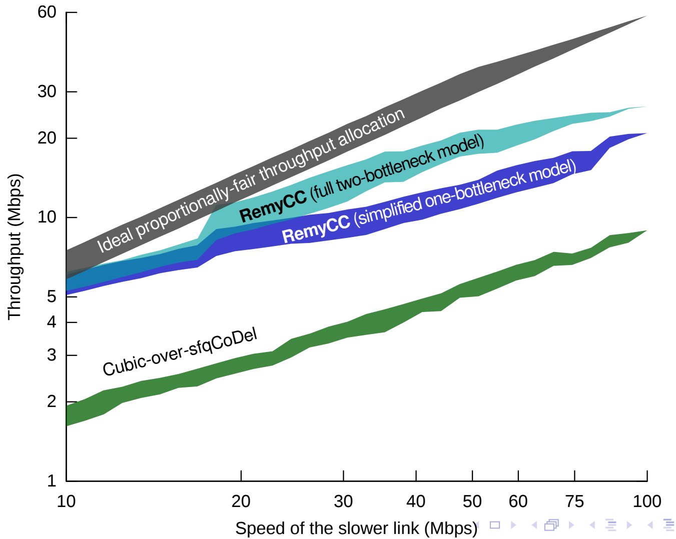
When the model is wrong about the topology



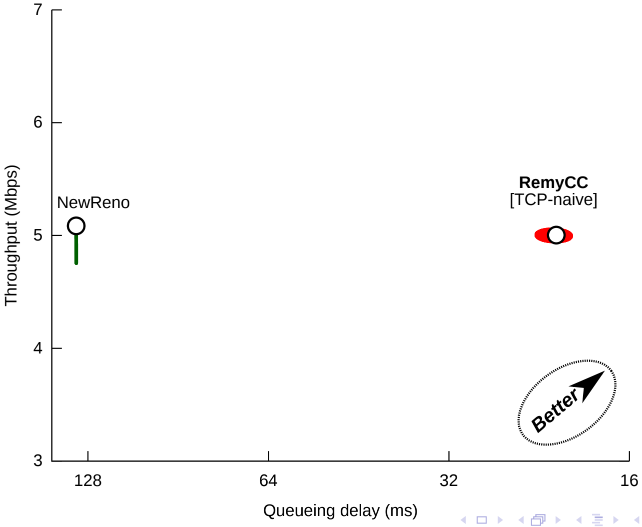
When the model is wrong about the topology



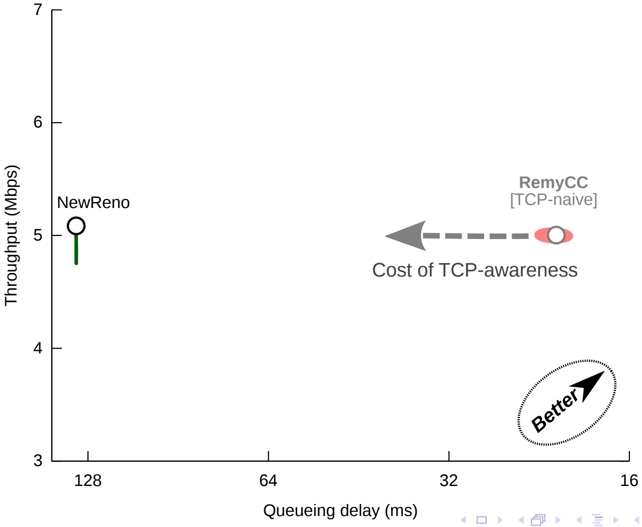
When the model is wrong about the topology



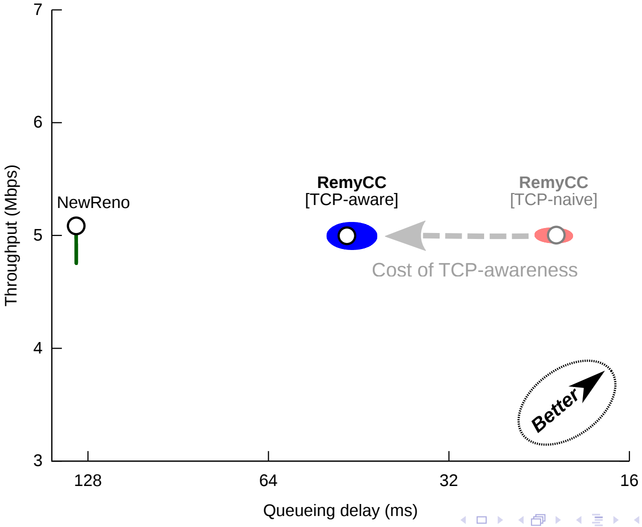
RemyCC competing against itself



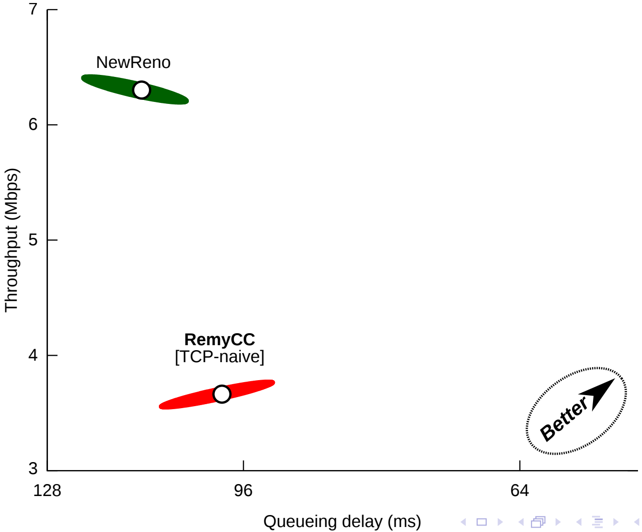
RemyCC competing against itself



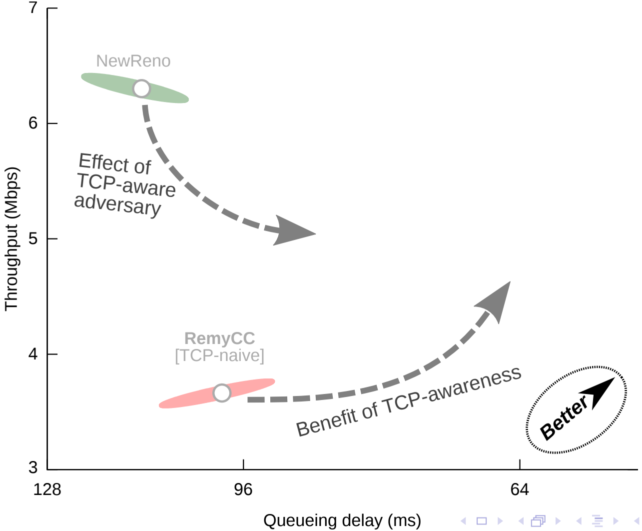
RemyCC competing against itself



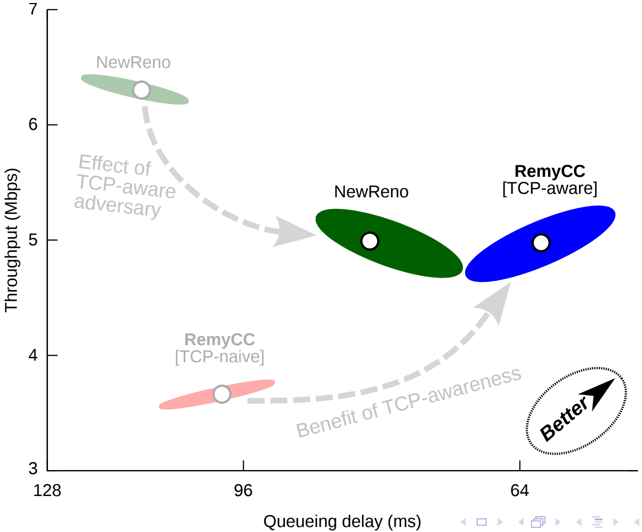
RemyCC competing against TCP NewReno



RemyCC competing against TCP NewReno



RemyCC competing against TCP NewReno



Can applications with different objectives coexist?

- Tpt. Sender: A throughput-intensive sender

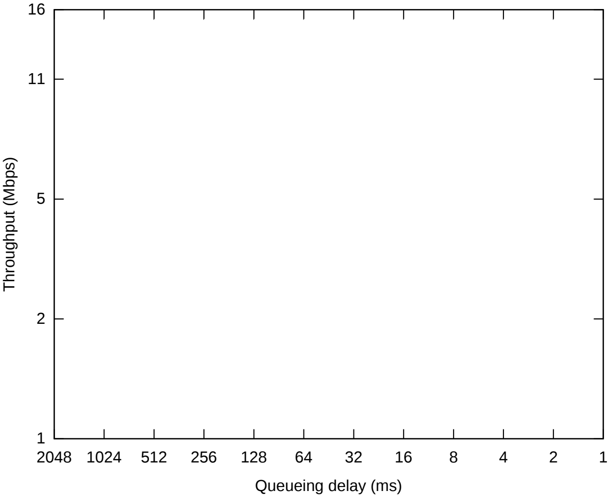
$$log(throughput) - 0.1 * log(delay) \qquad (1)$$

- Lat. Sender: A latency-sensitive sender

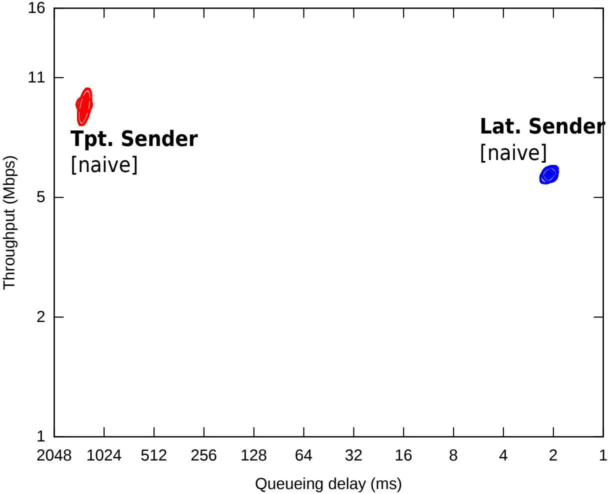
$$log(throughput) - 10.0 * log(delay) \qquad (2)$$

- Running over a FIFO queue

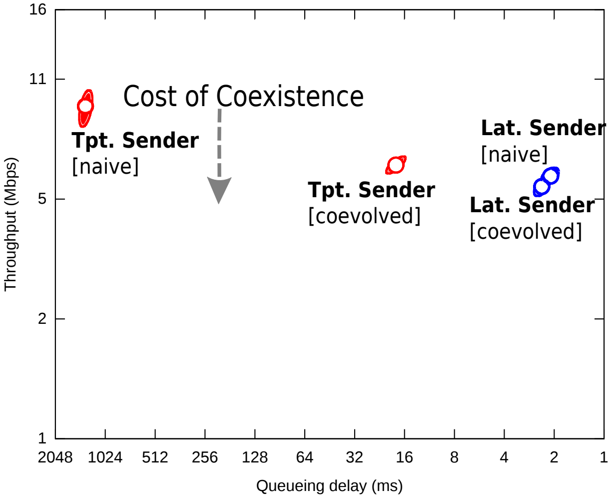
Training for diversity has a cost ...



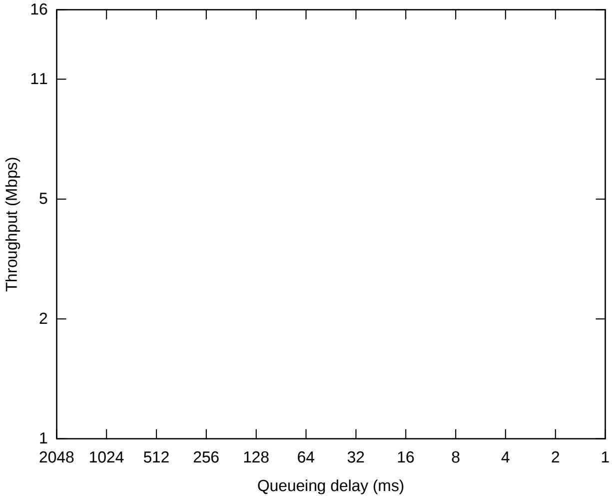
Training for diversity has a cost ...



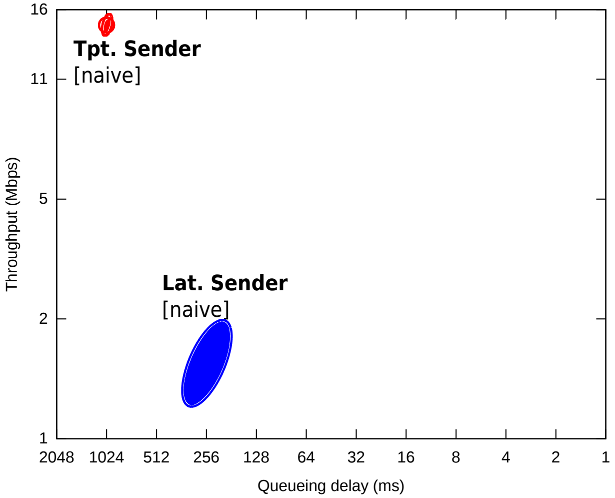
Training for diversity has a cost ...



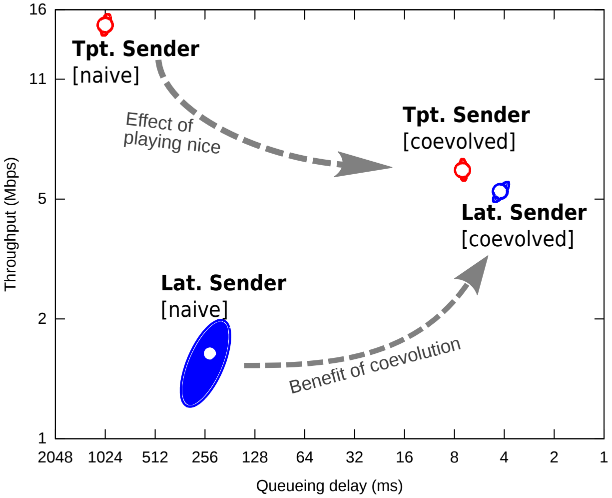
but, benefits the docile sender



but, benefits the docile sender



but, benefits the docile sender



Related Work

- Probably approximately correct learning
- Transfer learning
- Machine-generated congestion control

Limitations and future work

- Generalizability to more complex topologies?
- Better characterization of gap from optimal
- Do results change if we learn in-network behavior as well?
- Model mismatches between simulation and the real world