PCC: Performance-oriented Congestion Control

Michael Schapira

Hebrew University of Jerusalem

and

Compira Labs (co-founder and chief scientist)

Performance-oriented Congestion Control

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

האוניברסיטה העברית בירושלים
THE HEBREW UNIVERSITY OF JERUSALEM

“PCC: Re-architecting Congestion Control for Consistent High Performance” @ USENIX NSDI 2015
http://www.cs.huji.ac.il/~schapiram/PCC.pdf

“Vivace: Online-Learning Congestion Control” @ USENIX NSDI 2018
https://www.usenix.org/conference/nsdi18/presentation/dong
(to be posted shortly)

Performance-oriented Congestion Control

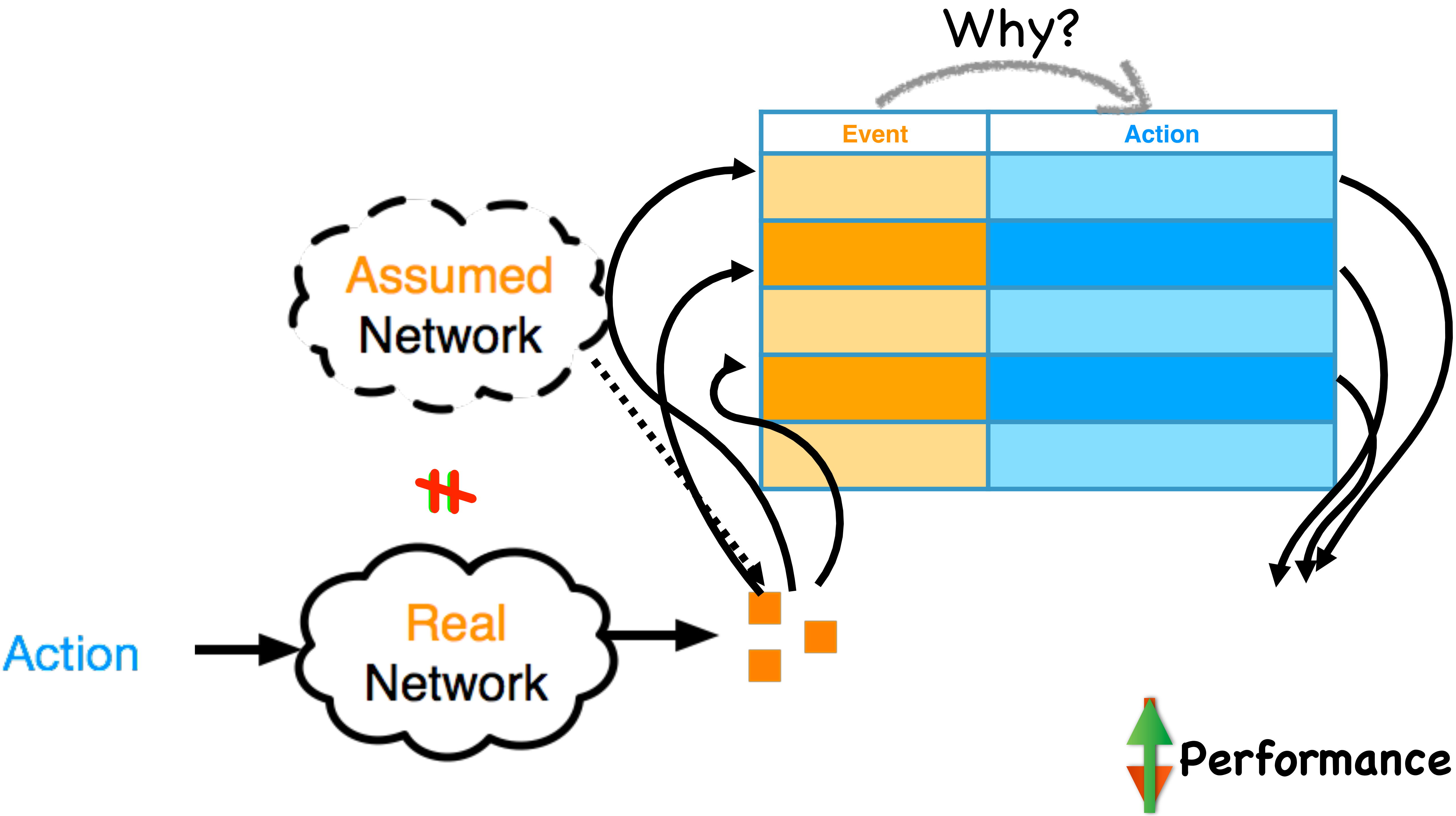
Currently being evaluated by

Google AT&T HUAWEI

and others

What is TCP?

	Event	Action
Reno		
Scalable		
Vegas		
FAST		
HTCP		


Why?
Event
Action
Assumed
Network
Real
Network
Action
Performance

Flow f sends at R, and then...

Event

Action

~~Increase R~~

Network

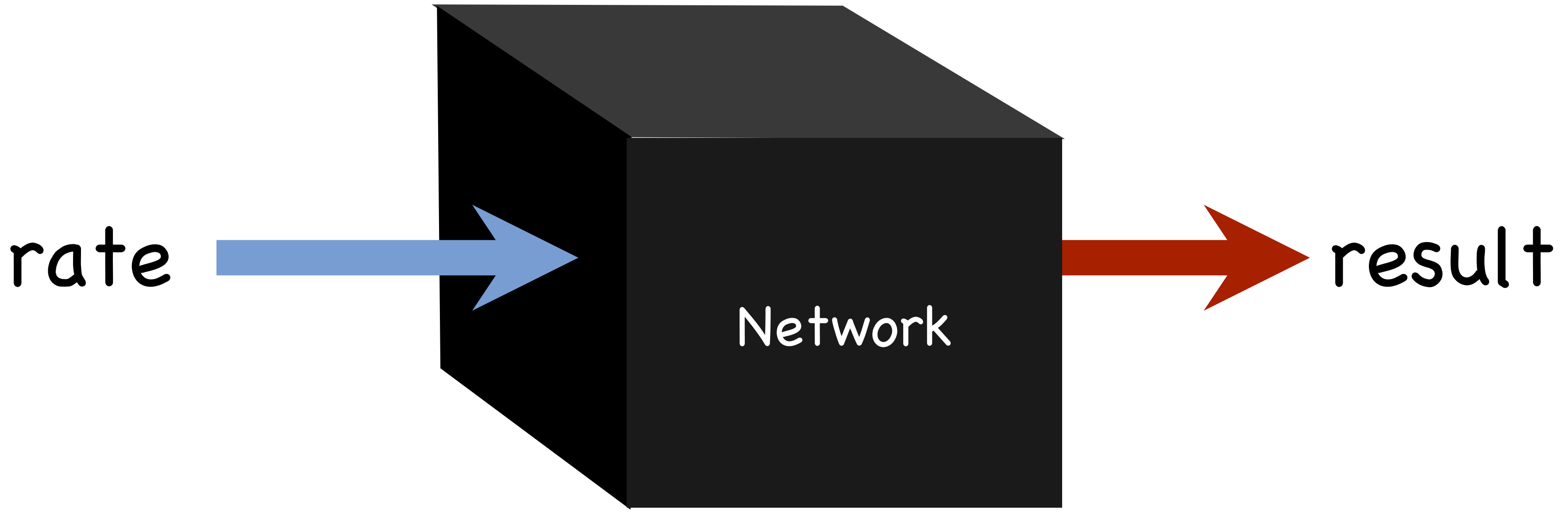
overflow

other high rate flow causing congestion

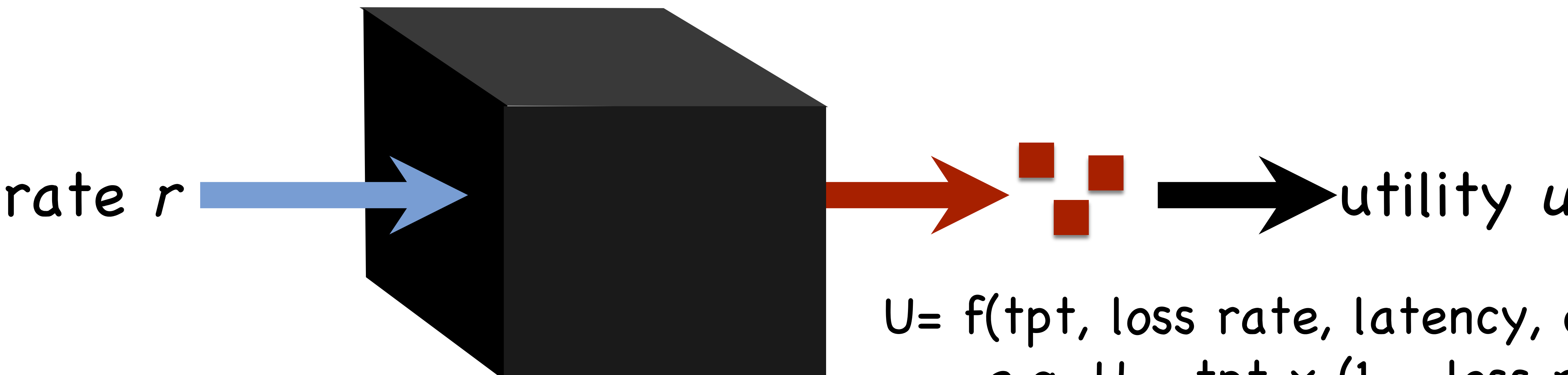
loss is random

No TCP-style event-control mapping optimal for all network scenarios

What is the right rate to send at?



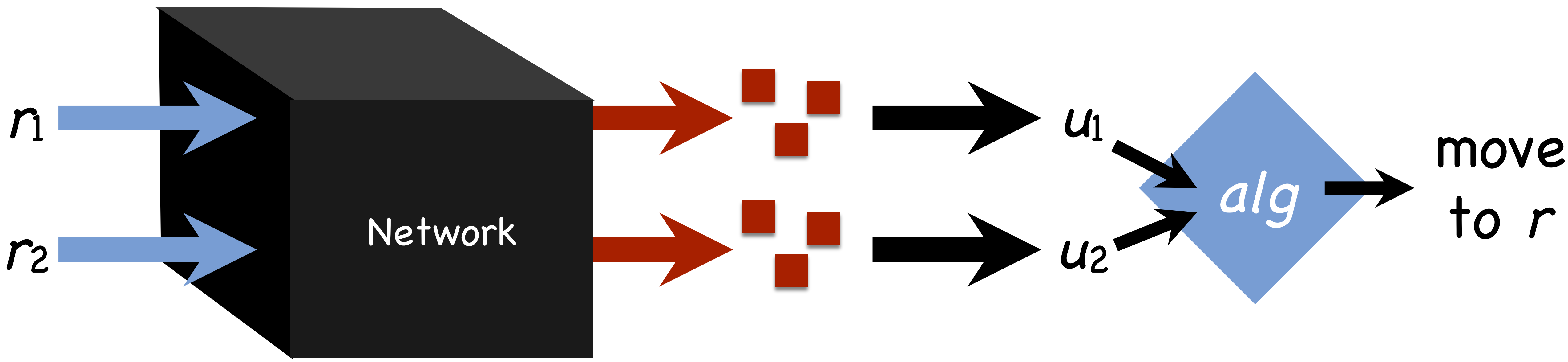
What is the right rate to send at?



U= f(tpt, loss rate, latency, etc.)

e.g. U = tpt x (1 – loss rate)

Performance-oriented Congestion Control



<u>**Learn**</u> real performance

=

Gather <u>**meaningful statistics**</u>

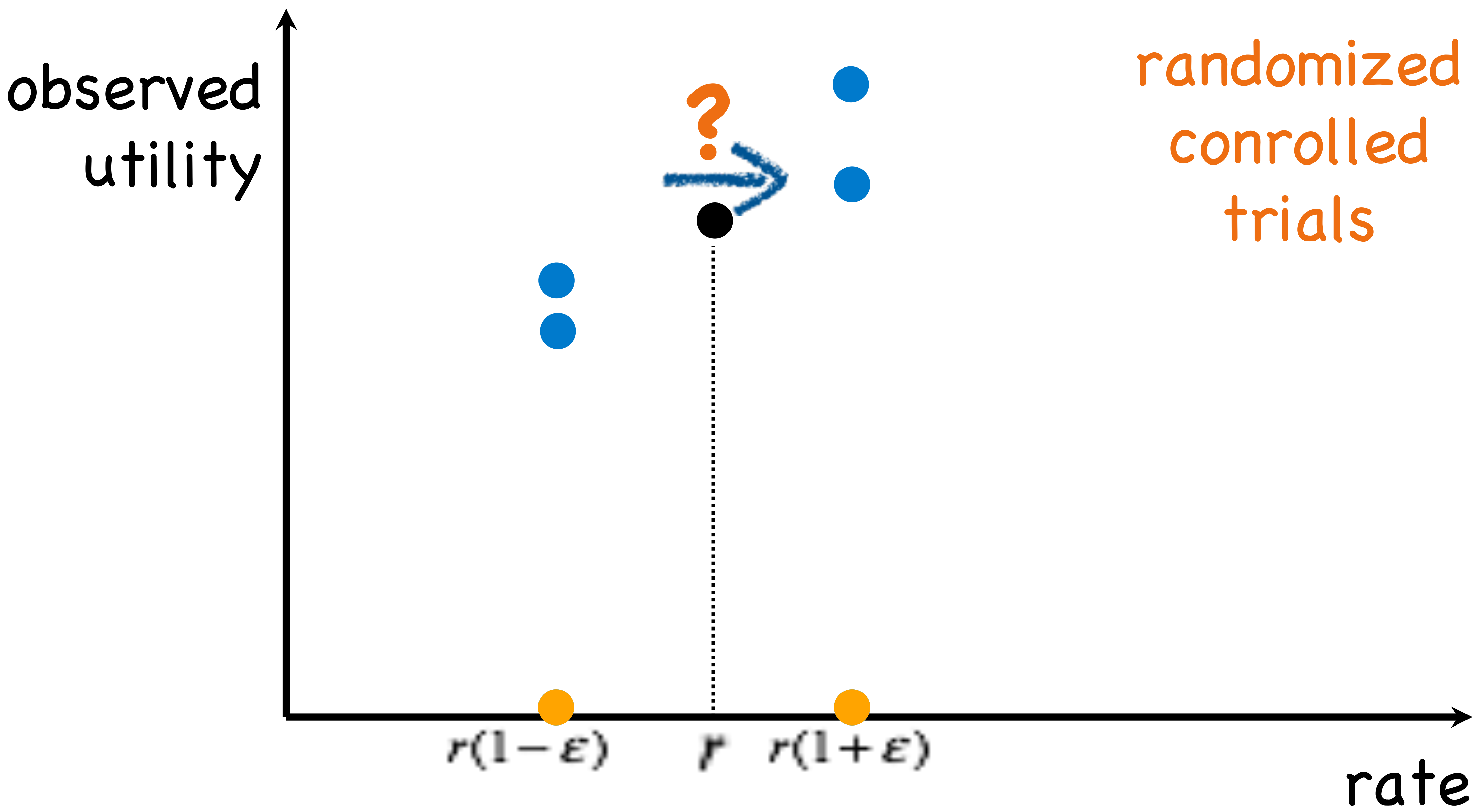
<u>**Control**</u> based on empirical evidence

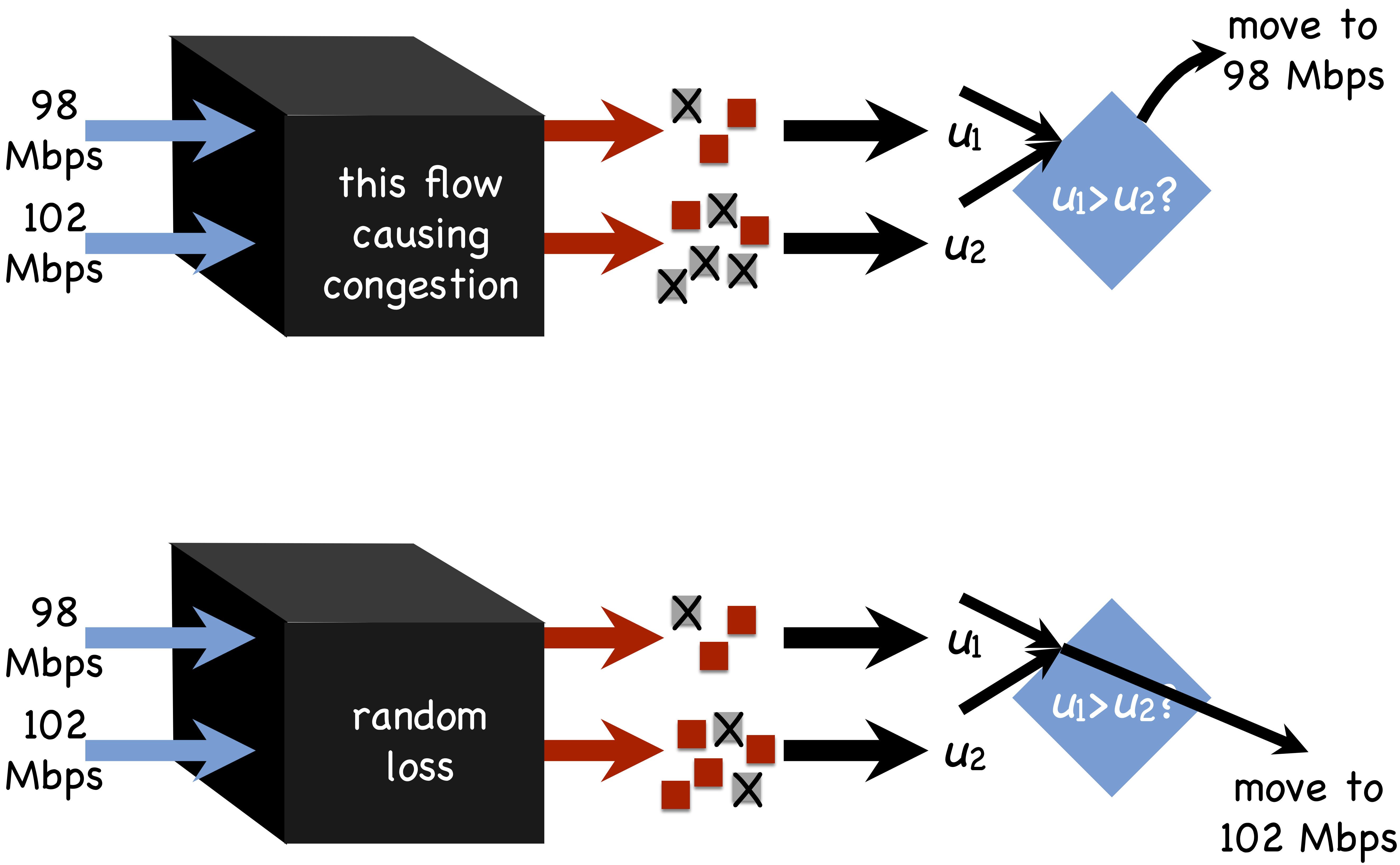
=

Apply <u>**online learning**</u> algorithm

yields **Consistent high performance**

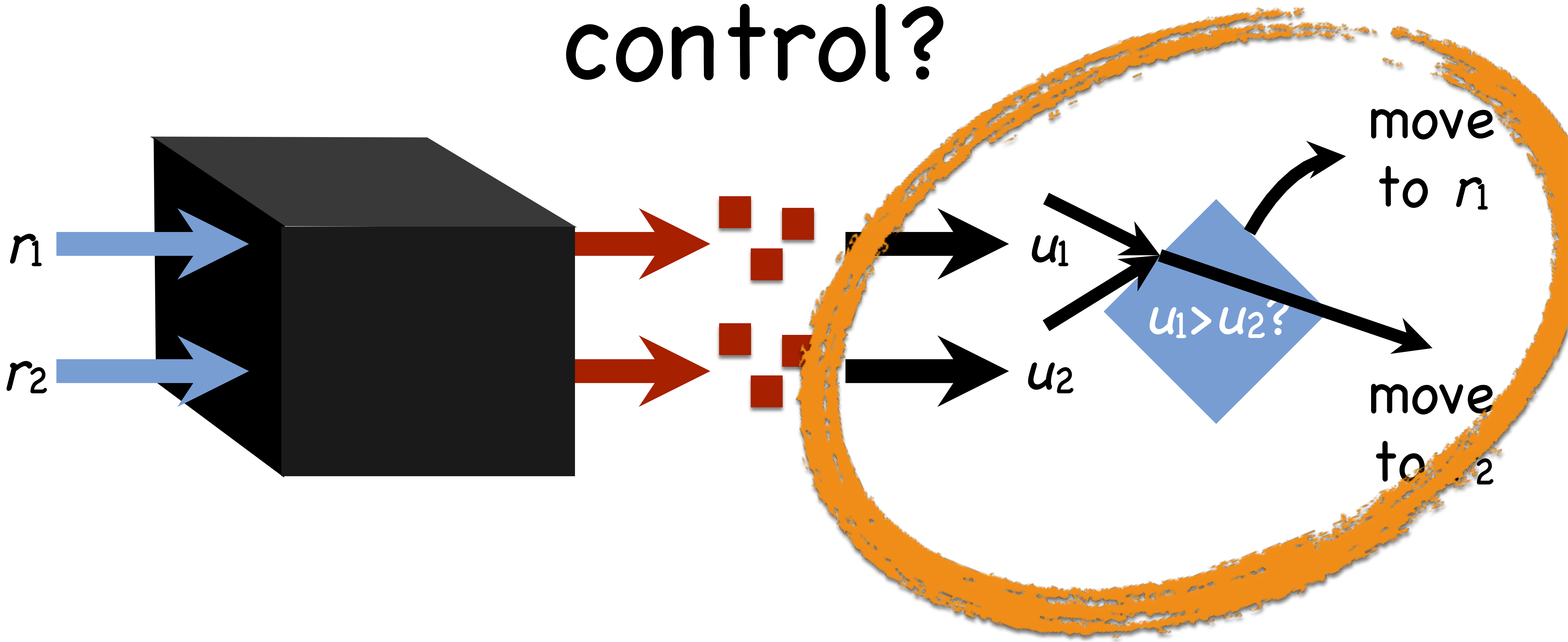
PCC Allegro (PCCv1, @ NSDI 15)




98 Mbps
102 Mbps
this flow causing congestion
u1
u2
u1>u2?
move to 98 Mbps
98 Mbps
102 Mbps
random loss
u1
u2
u1>u2?
move to 102 Mbps

But where's the congestion control?

But where's the congestion control?



Selfish utility-maximizing decision
=> non-cooperative game

What utility functions guarantee "good" Nash equilibrium?

Congestion Control via Game Theory

Find a utility function that:

- has an unique and "nice" NE under FIFO queueing
- expresses a generic data transmission objective
- maintains consistently high performance

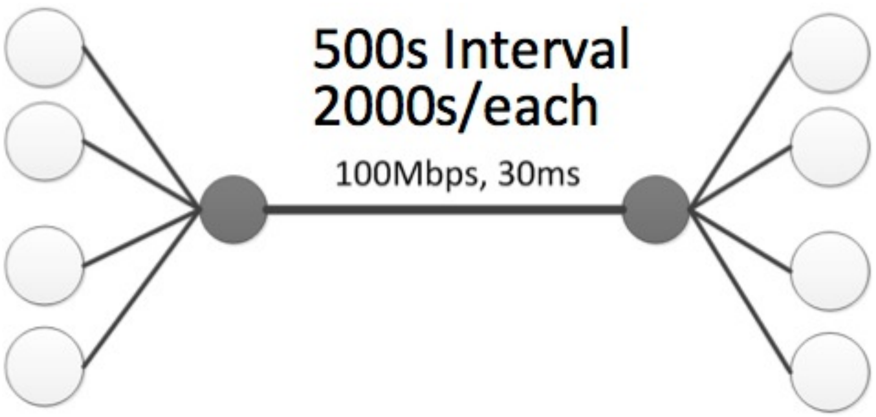
$$u_i(x) = T_i \quad - x_i * L_i$$

x_i is sending rate

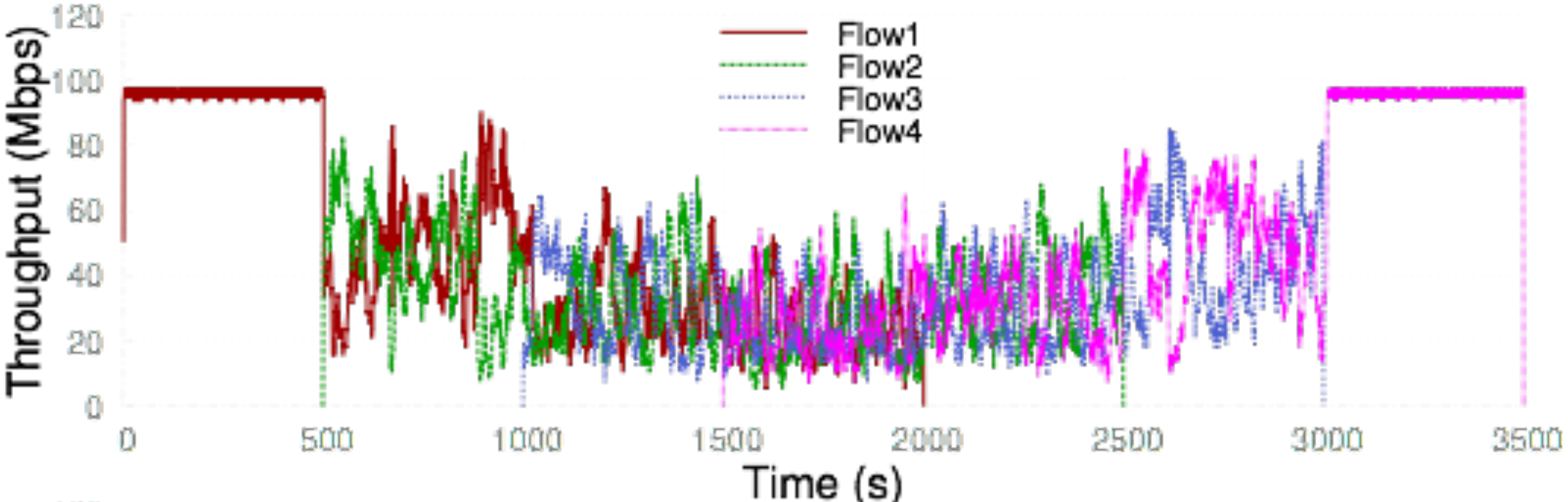
L_i is the observed loss rate

$T_i = x_i * (1 - L_i)$ is throughput

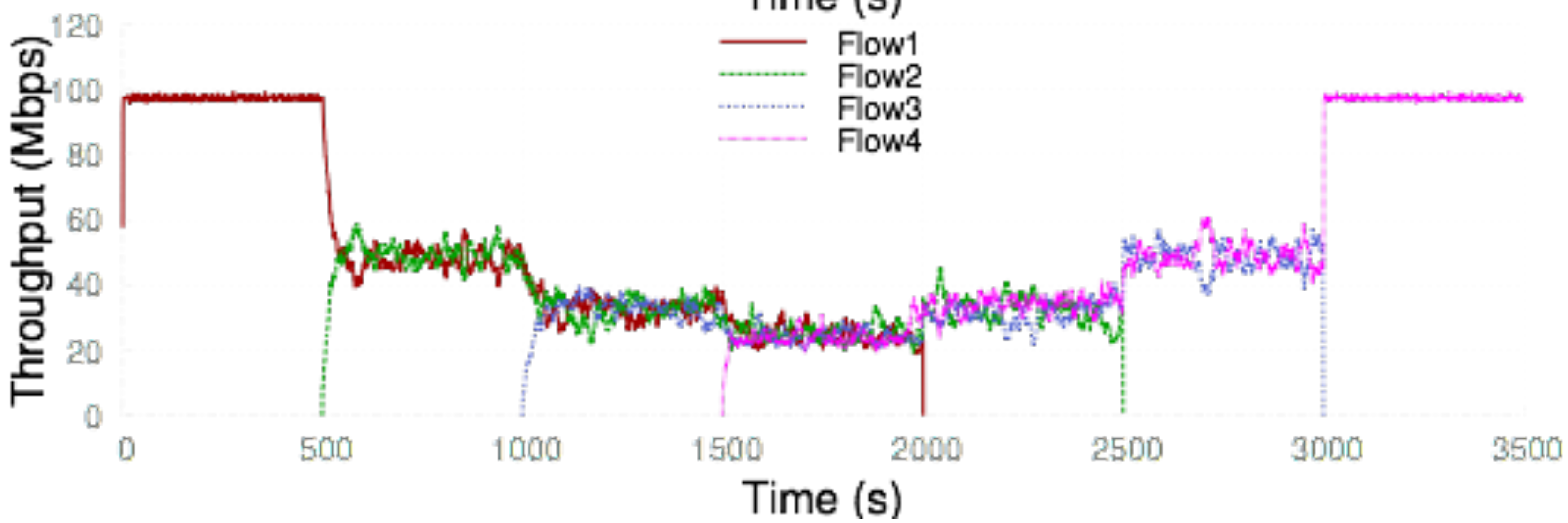
Convergence



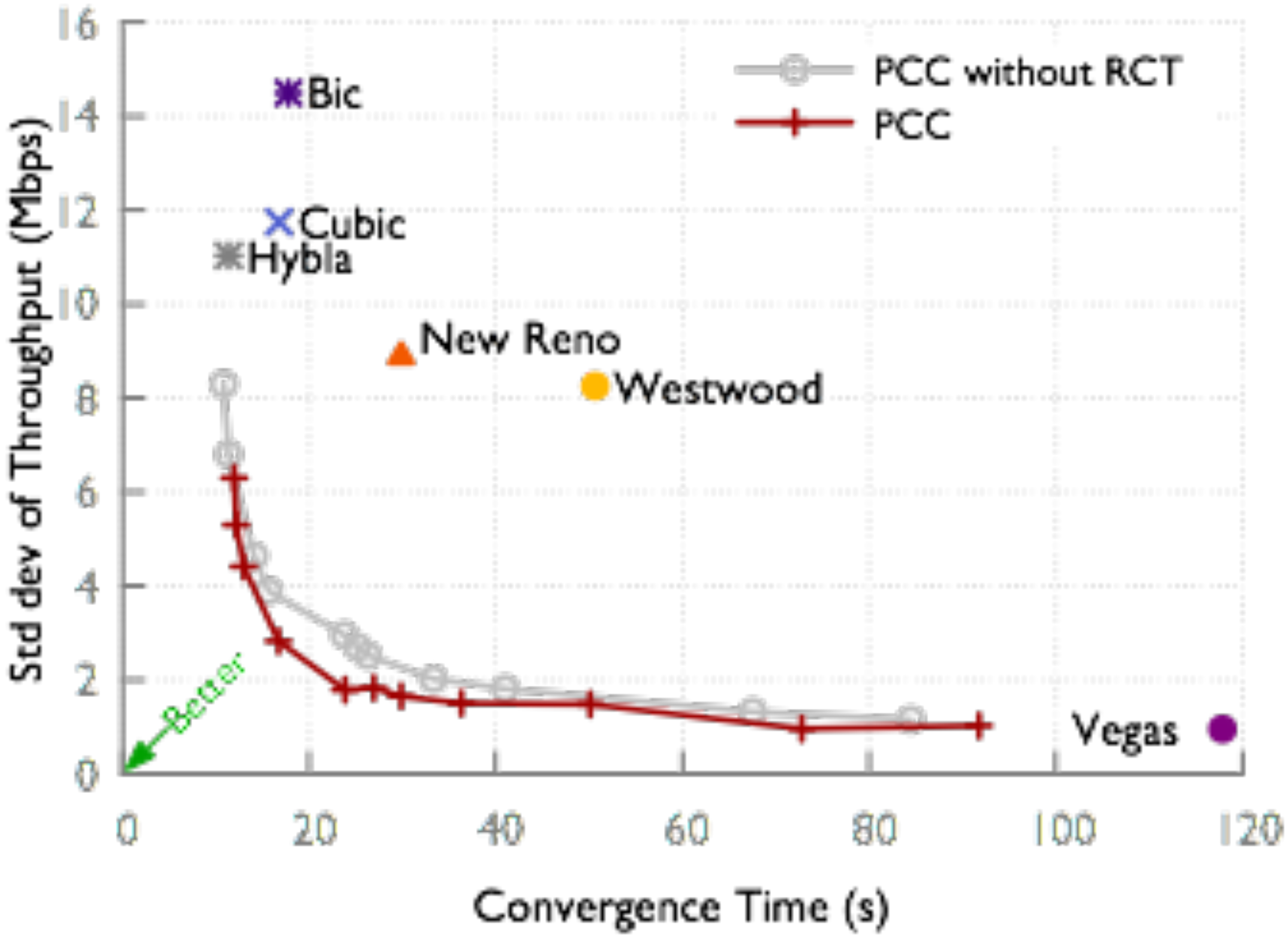
TCP



PCC



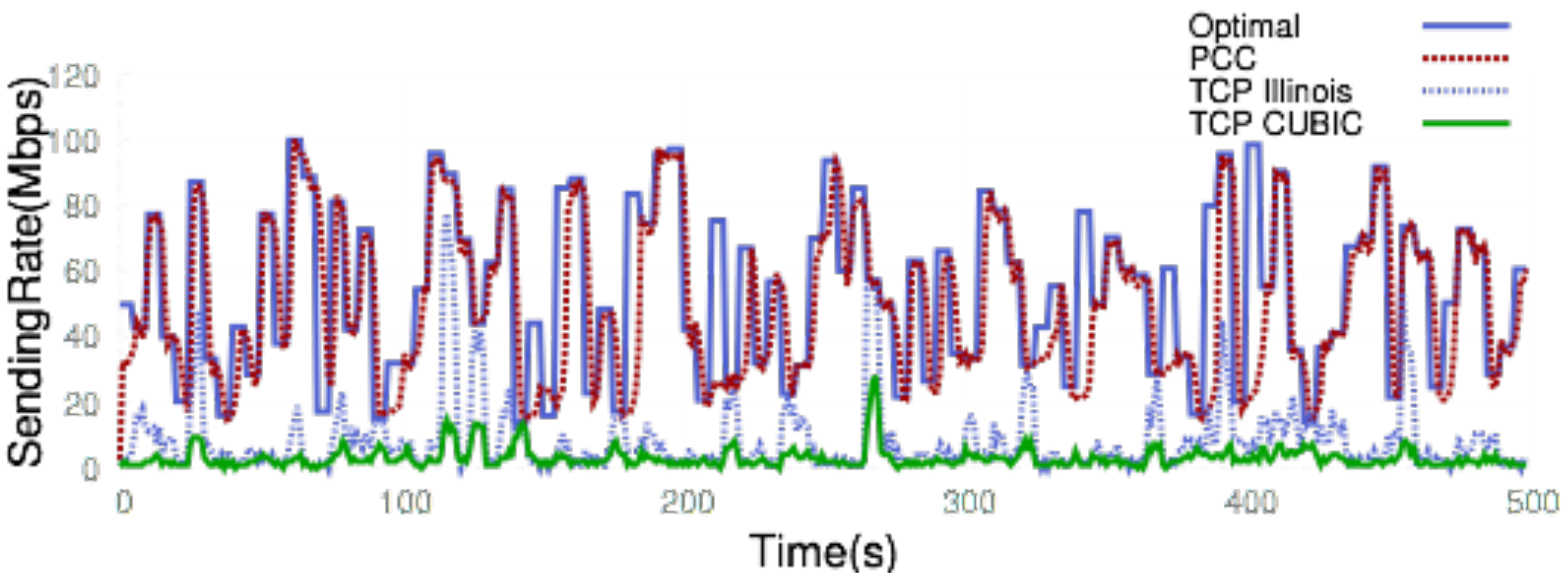
Reactiveness-stability trade-off



Consistently High Performance

Rapidly Changing Networks

BW: 10-100Mbps; RTT: 10-100ms; Loss Rate: 0-1%
Change every 5 seconds

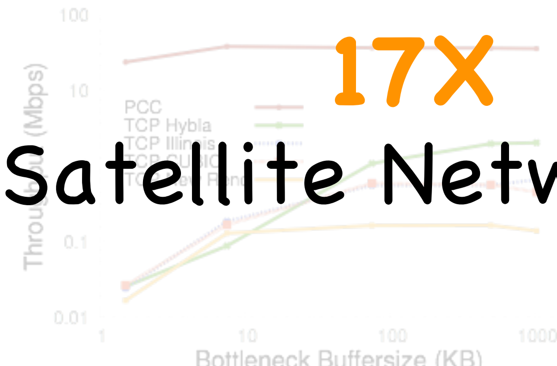


Consistently High Performance

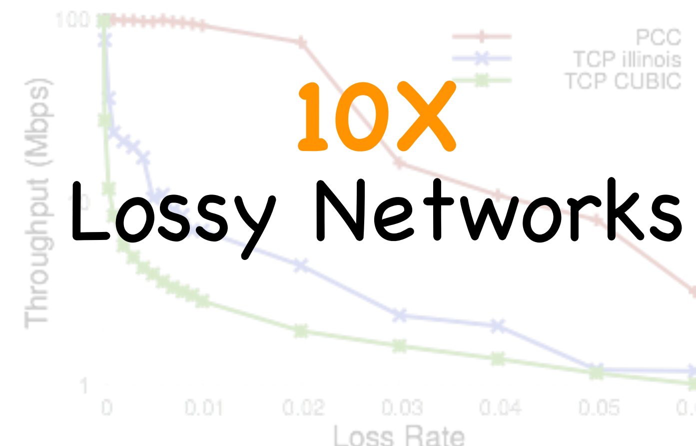
Table 1: PCC significantly outperforms TCP in inter-data center environments. RTT [illegible]; throughput in Mbps.

Transmission Pair	RTT	PCC	SABUL	CUBIC	Illinois
GPO → NYSERNet	[illegible]	[illegible]	[illegible]	129	326
GPO → Missouri	[illegible]	[illegible]	[illegible]	80.7	90.1
GPO → Illinois	35.4	766	664	84.5	102
NYSERNet → Missouri	47.4	816	662	108	109
Wisconsin → Illinois	9.01	801	700	547	562
GPO → Wisc.	38.0	783	487	79.3	120
NYSERNet → Wisc.	38.3	791	673	134	134
Missouri → Wisc.	20.9	807	698	259	262
NYSERNet → Illinois	36.1	808	674	141	141

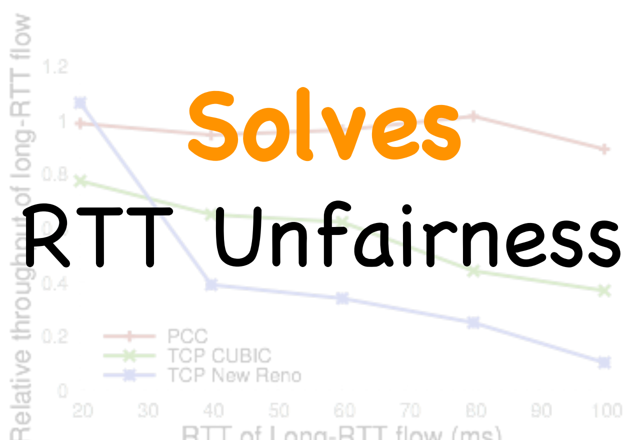
4X
InterDC



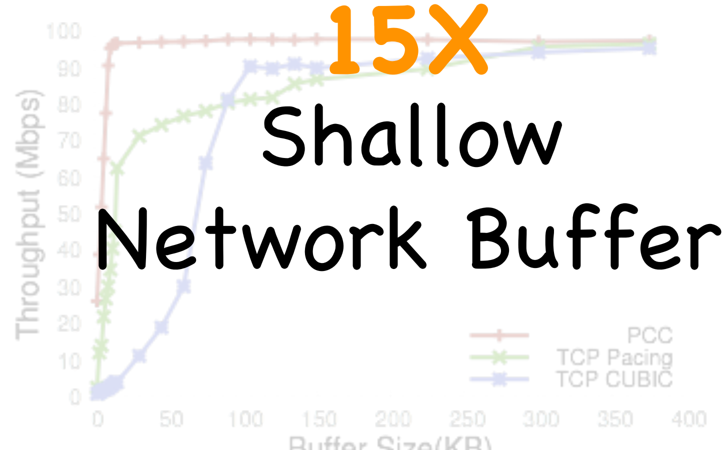
17X
Satellite Networks



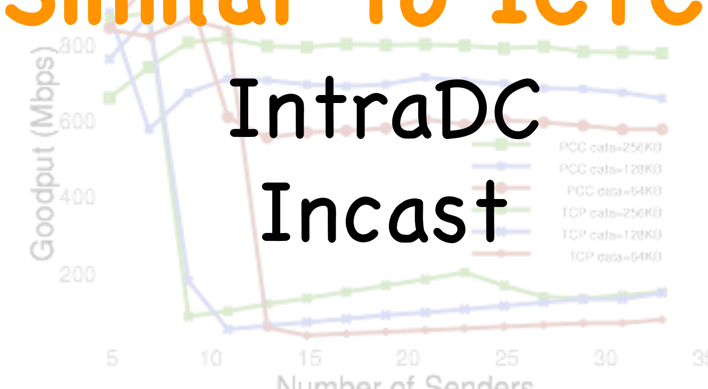
10X
Lossy Networks



Solves
RTT Unfairness



15X
Shallow Network Buffer



Similar to ICTCP
IntraDC Incast

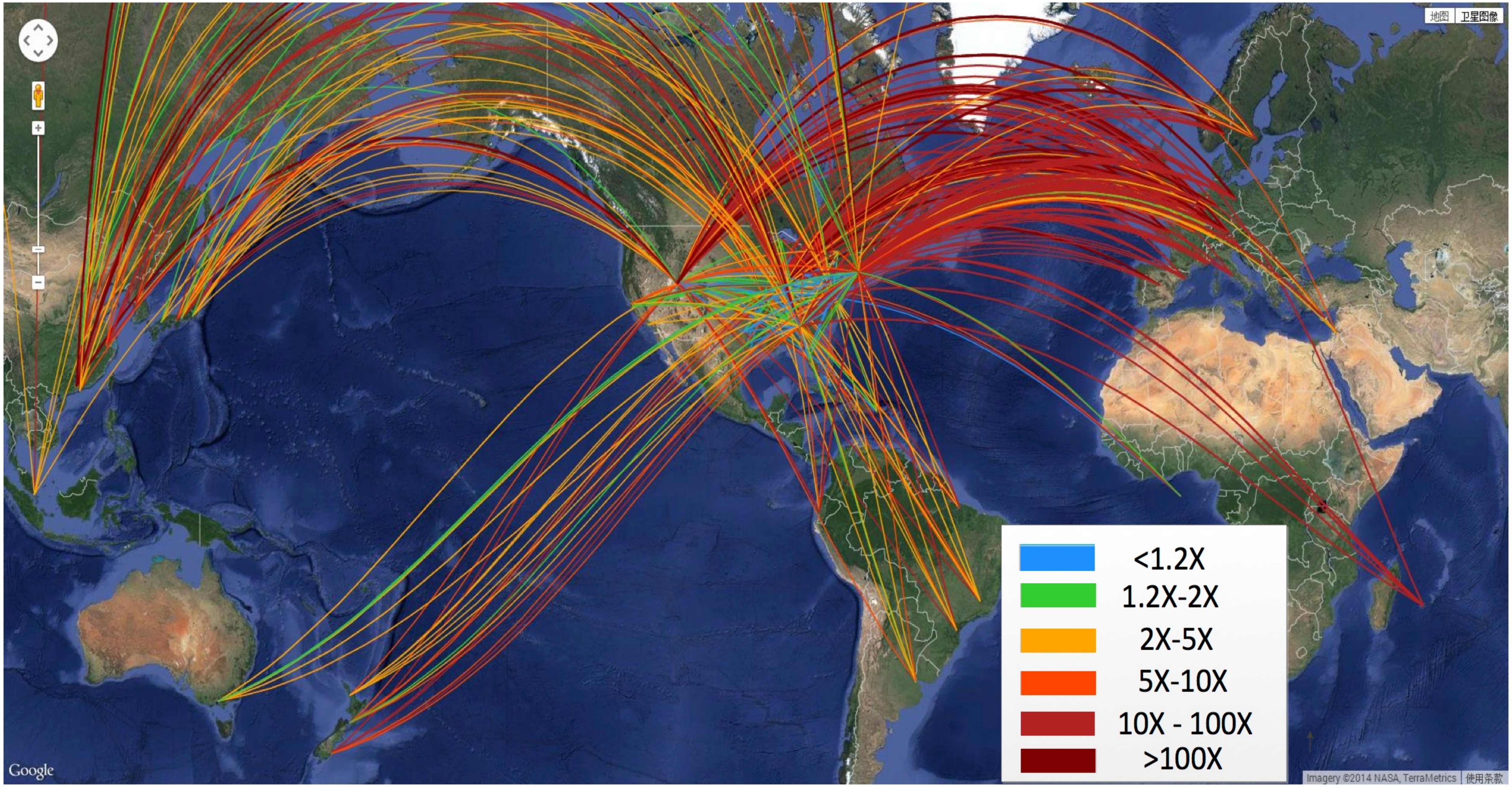
Close to Optimal
Rapidly Changing Networks

5X in median
Global Commercial Internet

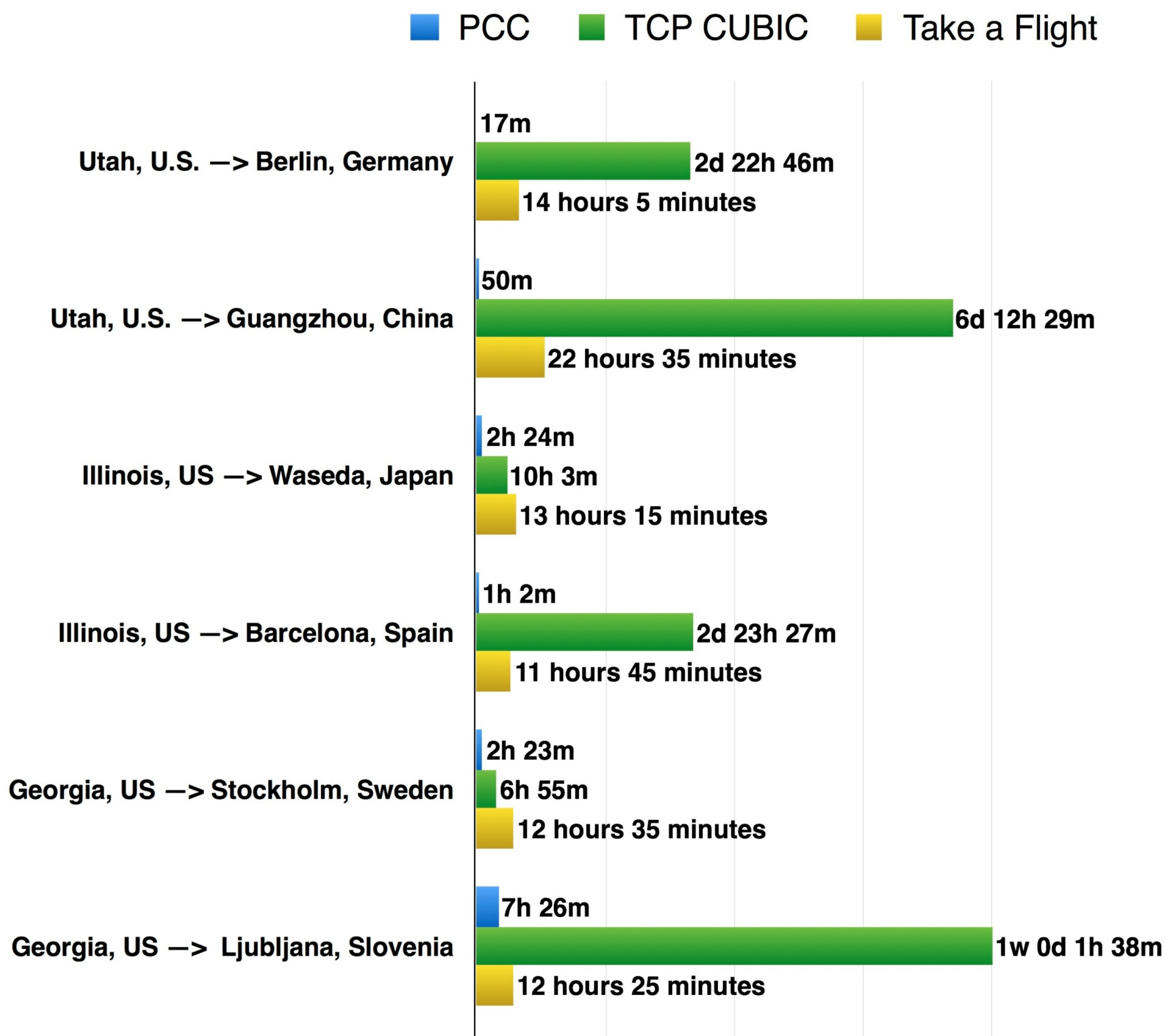
"PCC: Re-architecting Congestion Control for Consistent High Performance" @ USENIX NSDI 2015. http://www.cs.huji.ac.il/~schapiram/PCC.pdf

Consistently High Performance

Global Commercial Internet



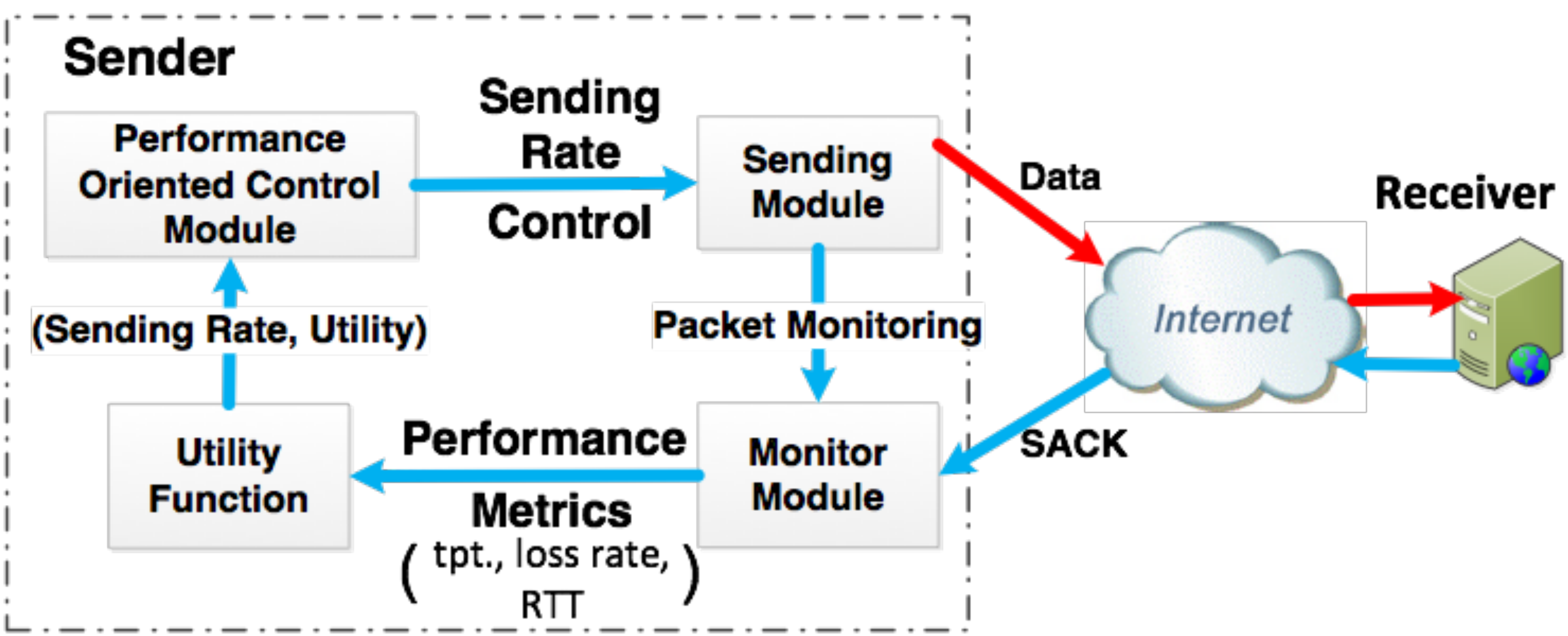
Delivering 100GB data…



Deployability and Deployment

Deploying PCC involves

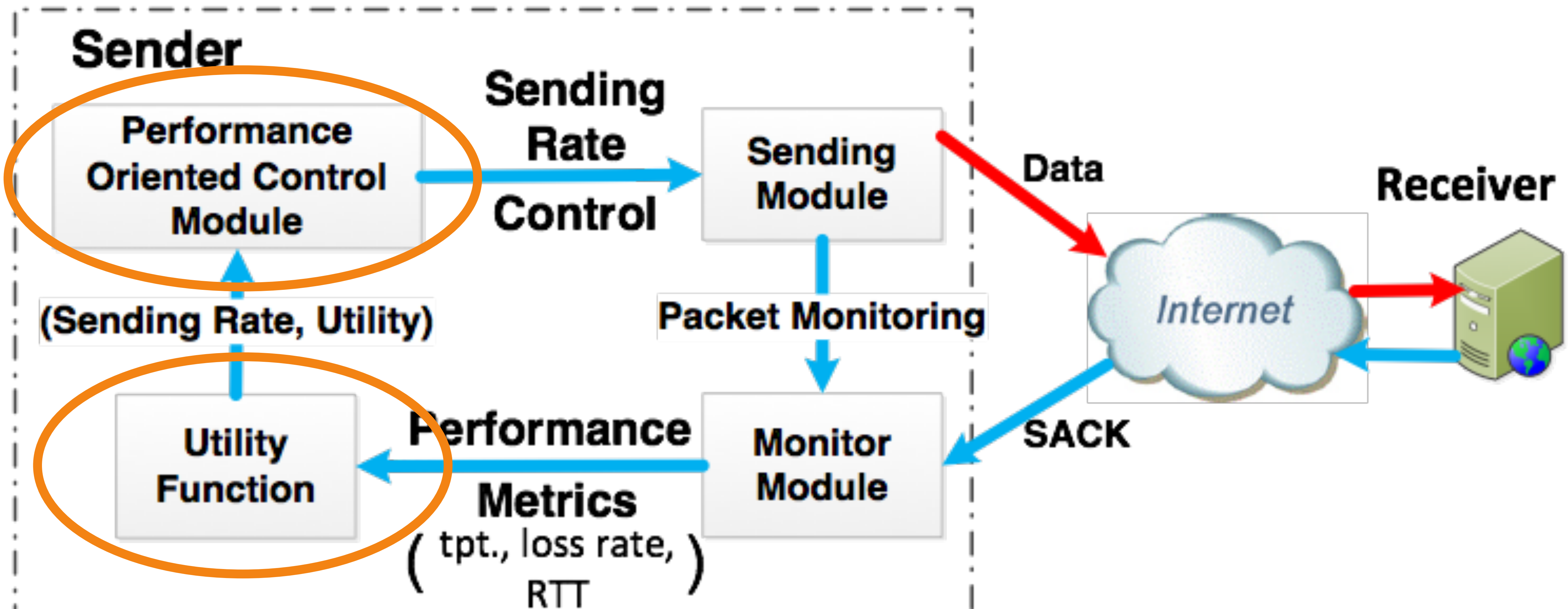
- software changes only
- no changes to the application layer
- and no changes to the (legacy TCP) receiver



…but PCC Allegro still far from perfect…

- Suboptimal convergence rate
- Little experimentation and no analysis of latency-based utility functions
- Bad performance in mobile networks
- Suboptimal QoE in OTT media delivery
- …

PCC Vivace (PCCv2 @ NSDI 18): Same architecture, new components



"Vivace: Online-Learning Congestion Control" @ USENIX NSDI 2018
https://www.usenix.org/conference/nsdi18/presentation/dong
(to be posted shortly)

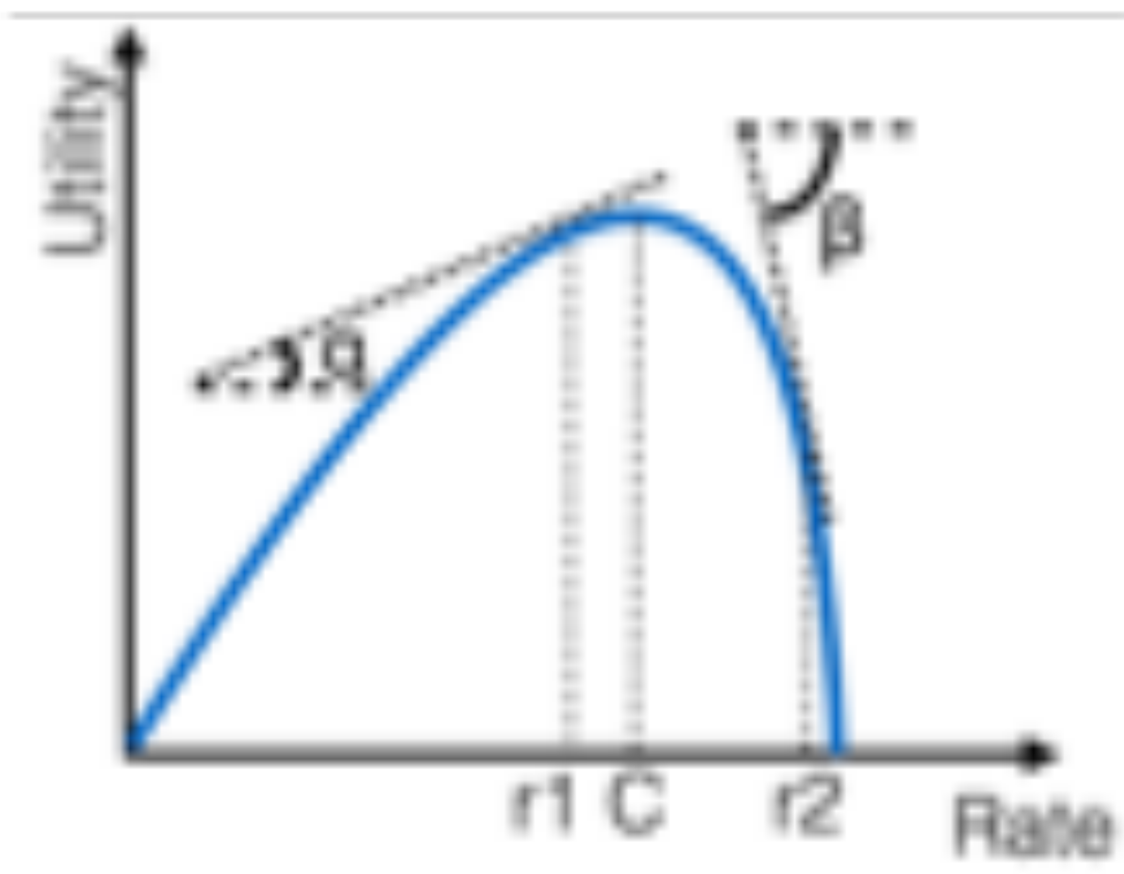
New utility function framework

- Incorporates latency
- Provable convergence, better convergence
- Can tailor different utility functions to different senders!
    - without compromising on convergence
    - while being able to reason about the resulting equilibrium

New online learning algorithm

Idea: gradient ascent on utility function

- Leverages provable results from online learning theory and game theory
- Additional techniques to contend with unreliable statistics



Comparison to BBR

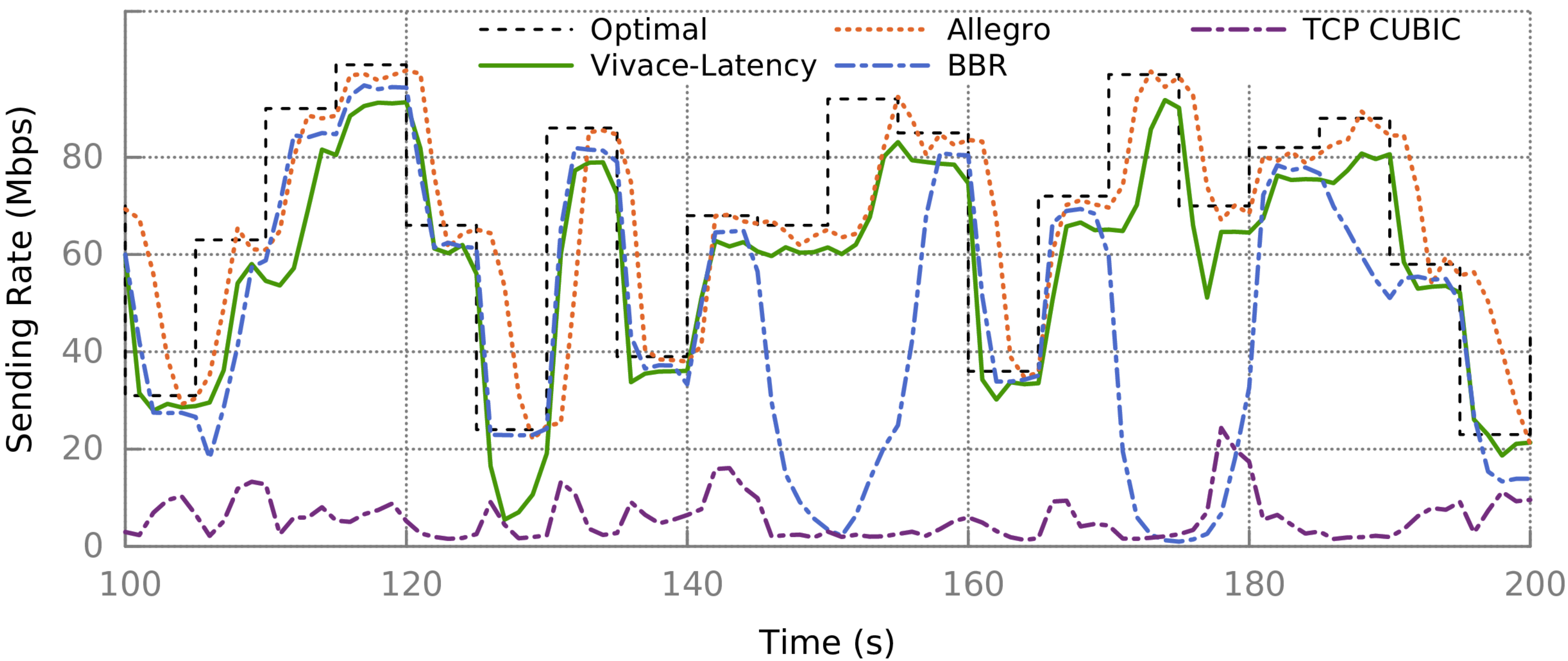
BBR:

- **Model the network pipe** as a single link
- Track the **bottleneck bandwidth**

PCC:

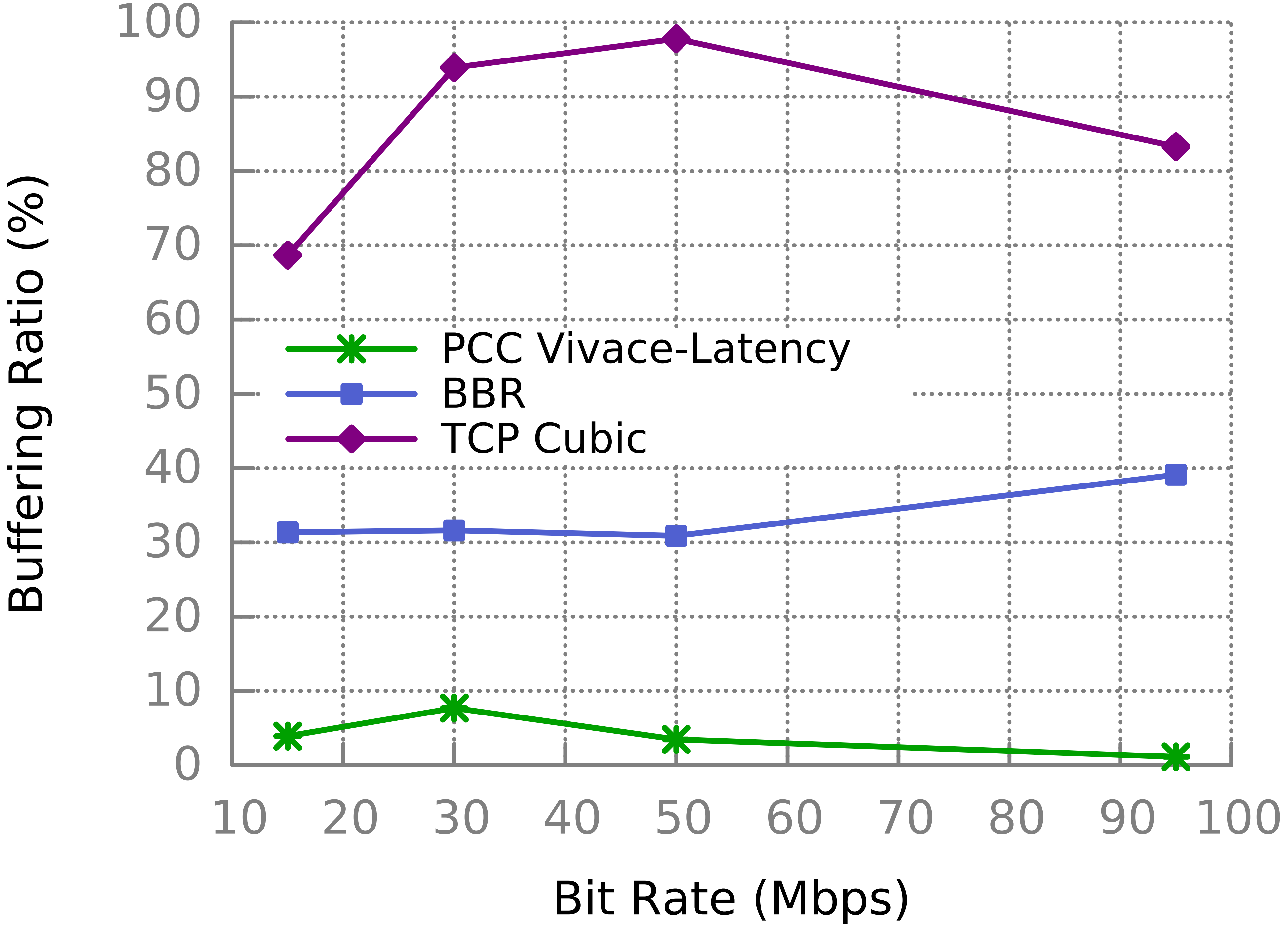
- Associate rate with **utility** value
- Apply **online learning** to adapt rate in direction/pace that empirically yields higher performance

PCC Reacts Better to Network Changes



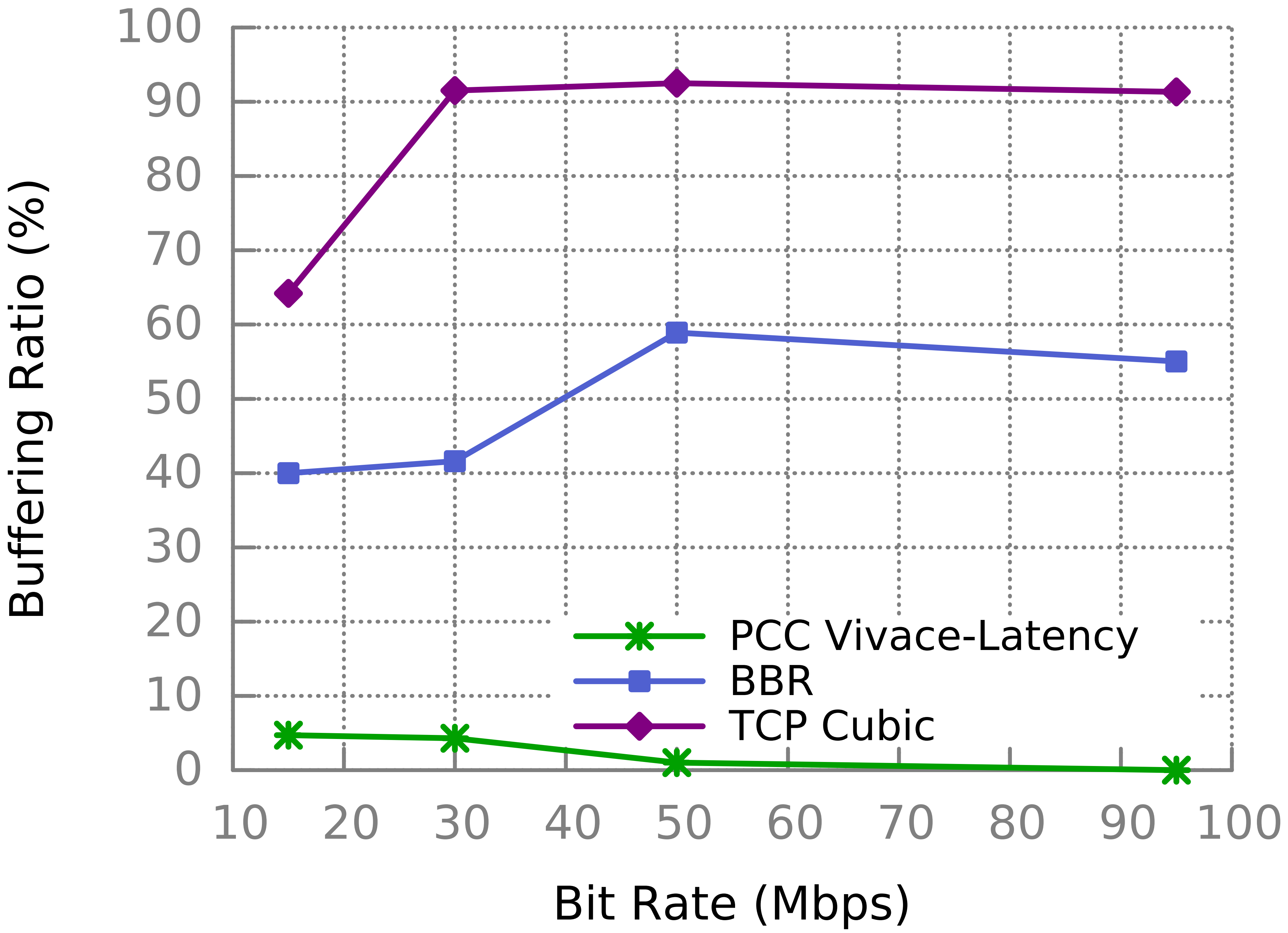
BW: 10-100Mbps; RTT: 10-100ms; Loss Rate: 0-1%
Change every 5 seconds

PCC Exhibits Improved Buffering Ratio for Streaming Video



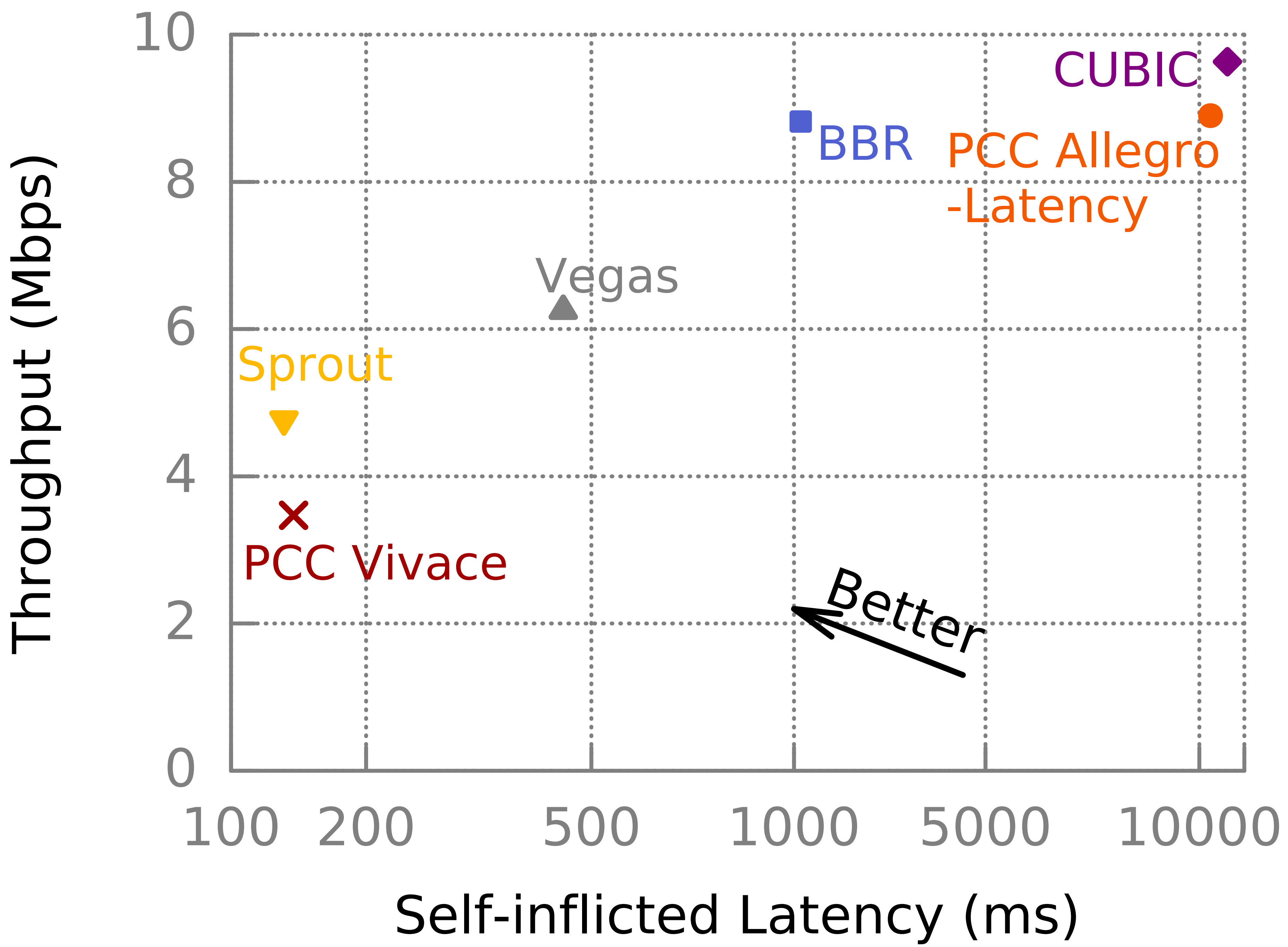
RTT changes between 10ms to 100ms in every five seconds. Link with 300KB buffer and 0.01% random loss rate. Network bandwidth at least 10% more than the required bitrate

PCC Improves Buffering Ratio (also for Multiple Video Streams)



3 competing streaming flows on bottleneck link with 75KB buffer, 100ms RTT, 0.01% random loss, and adequate bandwidth

PCC Achieves Better Throughput-Latency Tradeoffs in LTE-like Environments



Mahimahi used to replay Verizon-LTE trace

Demo

https://www.youtube.com/watch?v=Y3IzuCdwdUo

https://www.youtube.com/watch?v=4lt0JkumL-M

Also related

Congestion control throwdown
(with Keith Winstein from Stanford)
http://www.cs.huji.ac.il/~schapiram/Congestion_Control_Throwdown%20(5).pdf

https://www.youtube.com/watch?v=T1DCoNoVvRM

Ongoing Efforts

- Better online learning and utility frameworks
- PCC for future mobile networks
- Video-oriented PCC
- Open-source consortium
(center around kernel implementation and QUIC implementation of PCC)

See papers for …

- More stories about the fact that **TCP is broken**
- Proof of **fairness of Nash Equilibrium and Convergence**
- **Implementation** of PCC
  - Performance monitoring
  - Details of learning control algorithms
  - Implementation designs and optimizations
- **Performance Evaluation**
  - Inter data center networks
  - small buffer networks
  - Reactiveness and stability tradeoff
  - Jain index fairness
  - Benefit of Randomized Control Trials
  - Details of TCP friendliness evaluation
  - Emulated satellite networks
  - Emulated datacenter networks
  - Cure RTT unfairness
  - Does not fundamentally harm short flow FCT
  - Evaluation in the wild vs non-TCP protocols
  - …
- Flexibility by **pluggable utility function**

And more…

Thank You