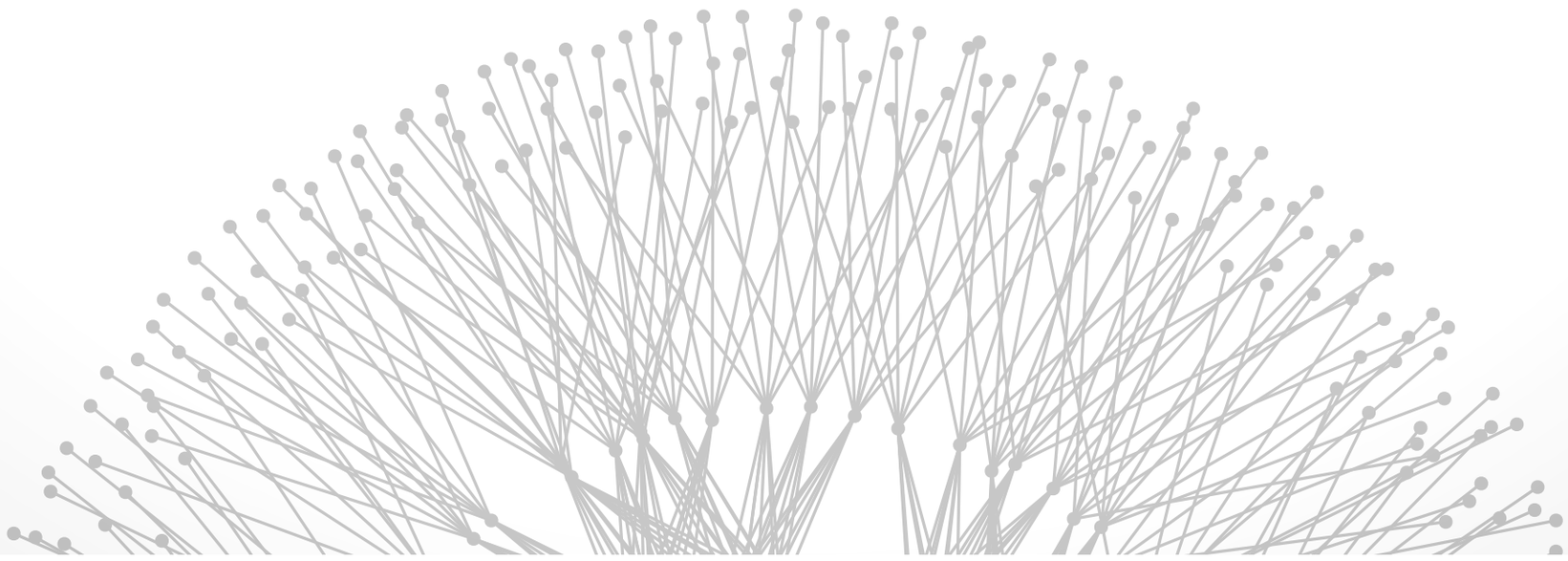
Modern Congestion Control

Mo Dong

CS 538 February 13 2017



Isn’t Congestion Control a done deal?

Congestion control with help from the network

All Good Stuff

0 Deployment


are
can be not happy with TCP ?

High BDP	Wireless	Satellite	Inter-DC
BIC			
H-TCP	Westwood	Hybla	
Compound	Vegas	STAR	Illinois
CUBIC	Veno		SABUL
FAST TCP			
10X	10X	17X	4X

Unstable, RTT Unfair, Bufferbloat, Crash on Changing Networks, ……

Point Solutions

+

Performance

Far from Optimal

CC Goals

Consistent High Performance

Fast and Stable Convergence

Why is it so hard?

Possible Answer No. 1

Not leveraging all available insights and capability of specific kinds of networks

Possible Answer No. 1

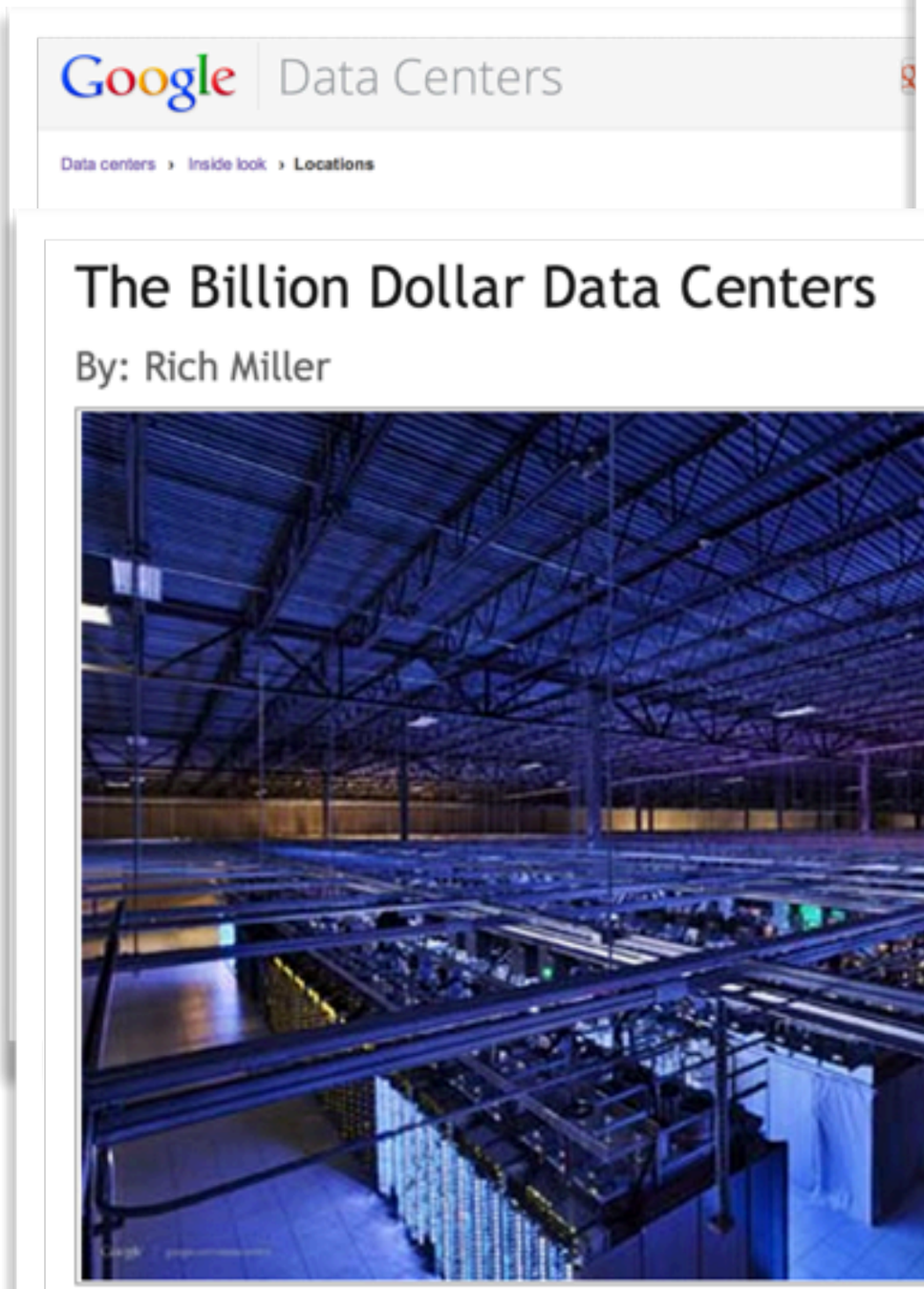
Data center networks (DCTCP, ICTCP, TIMELY)

- Insights to network properties
- Specific traffic patterns
- Full control of network infrastructure

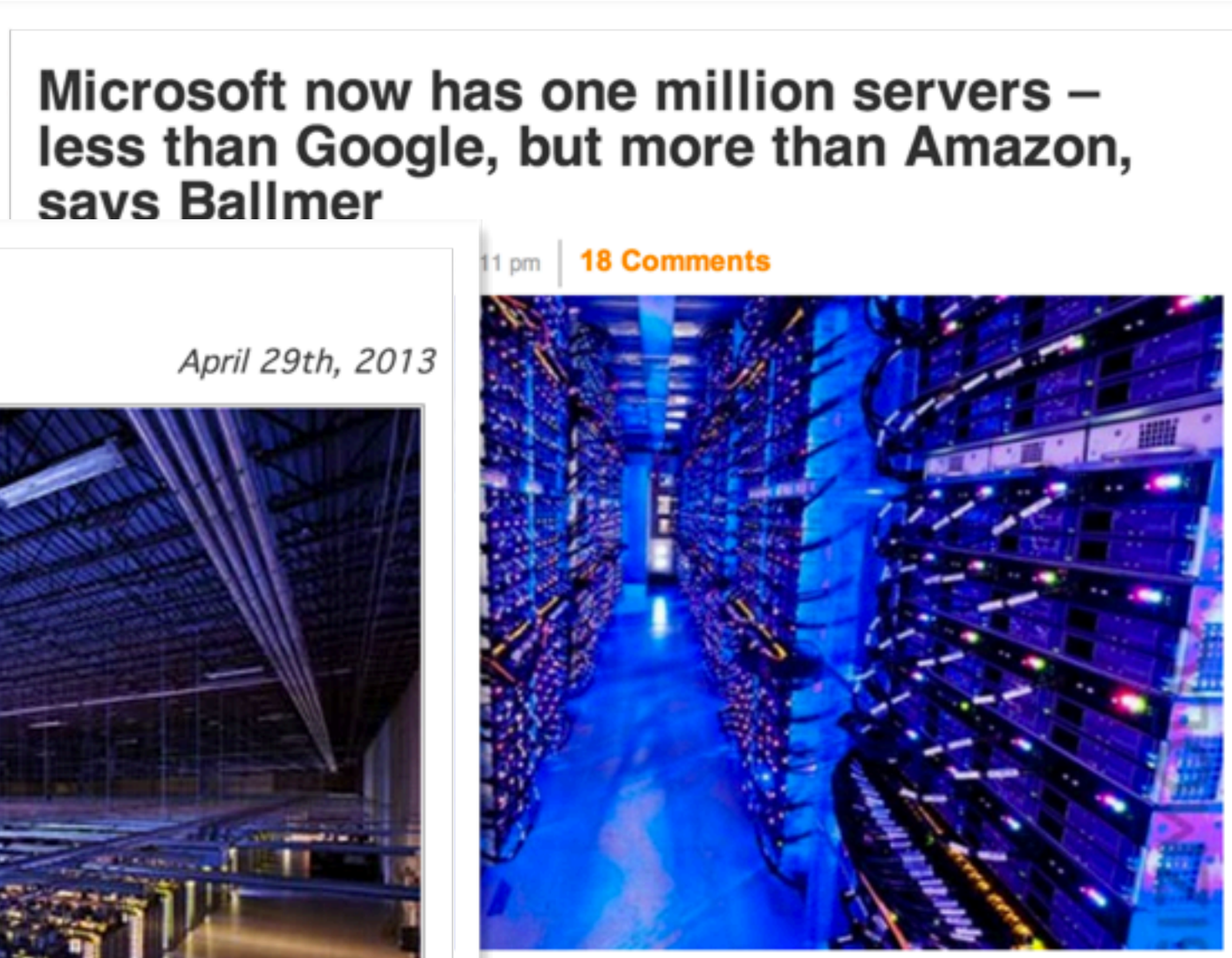
Cellular networks (Sprout, Verus)

- Insights to network properties

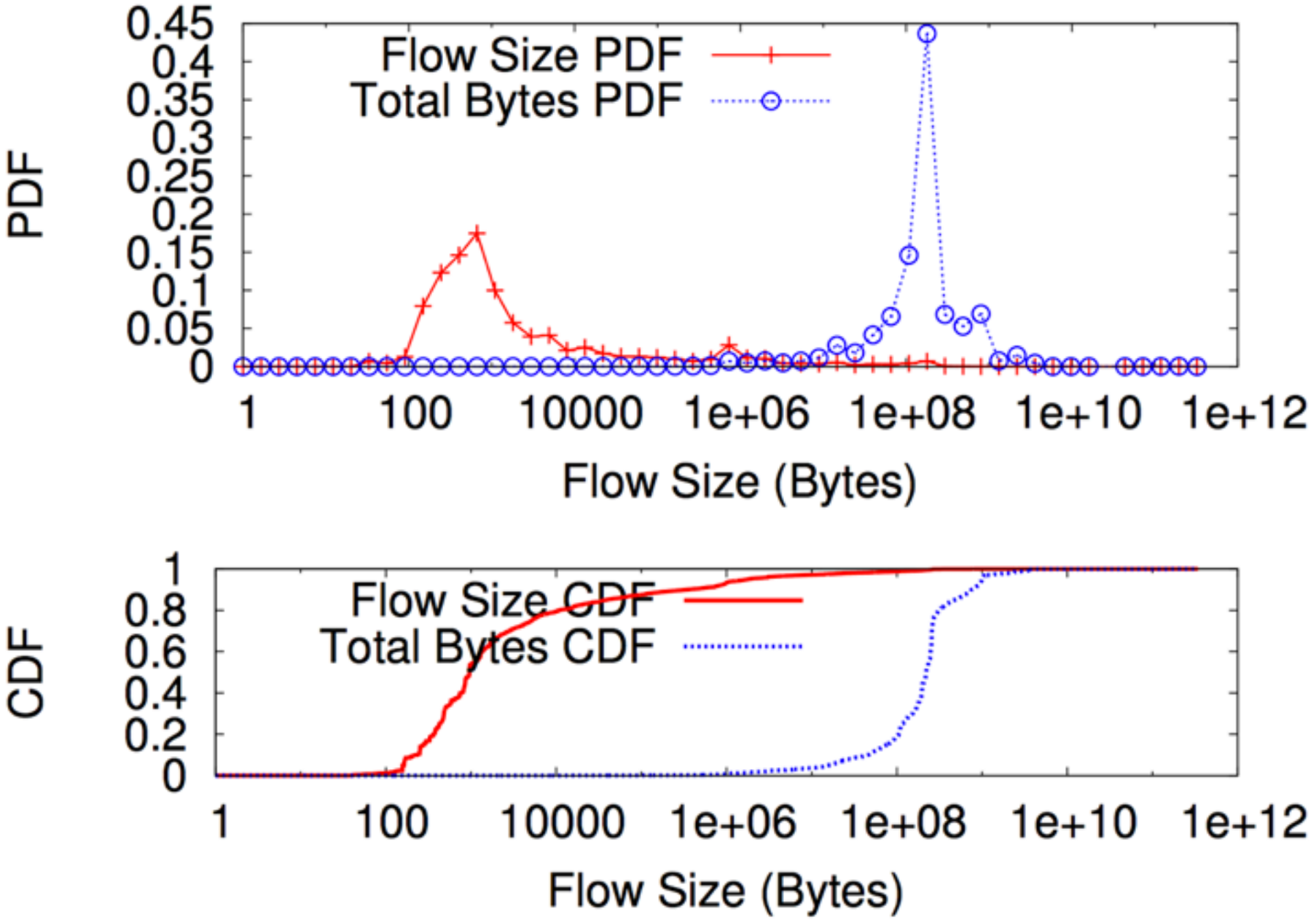
Data Center Networks



An overhead view of the server infrastructure in Google's data center in Council Bluffs, Iowa, where the company has invested more than $1 billion. (Photo: Connie Zhou for Google)



Data center traffic characteristics



[VL2, SIGCOMM'09]

What do we want?

Short flows

complete flows before their deadlines

Long flows

no deadline, but still preferable to finish earlier

Low latency is the key

YAHOO!

400 ms slowdown resulted
in a traffic decrease of 9%

[Yslow 2.0; Stoyan Stefanov]

Google

100 ms slowdown reduces
searches by 0.2-0.4%

[Speed matters for Google Web Search; Jake Brutlag]

AOL

Users with lowest 10% latency viewed 50% more
pages than those with highest 10% latency

[The secret weapons of the AOL optimization team; Dave Artz]

mozilla Firefox

2.2 sec faster web response
increases 60 million more Firefox
install package downloads per year

[Firefox and Page Load Speed; Blake Cutler]

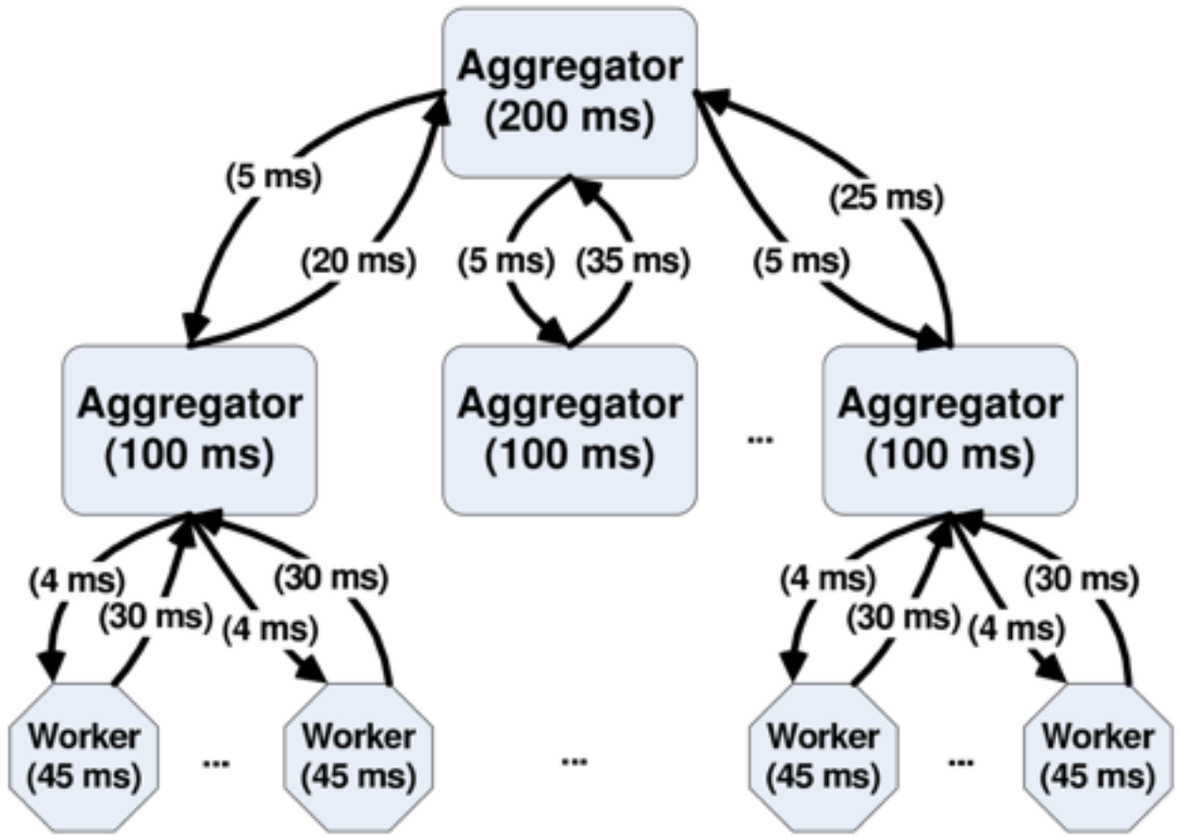
Walmart

Users with 0-1 sec load time have
2x conversion rate of 1-2 sec

[Is page performance a factor of site
conversion? And how big is it; Walmart Labs]

Improving latency in data centers

Server side optimization: Parallel computation



partition aggregate model

3 impairments [DCTCP]

- Incast
- Queue buildup
- Buffer pressure

Incast

What is TCP Incast problem?

- Synchronized flows overflow the switch buffer

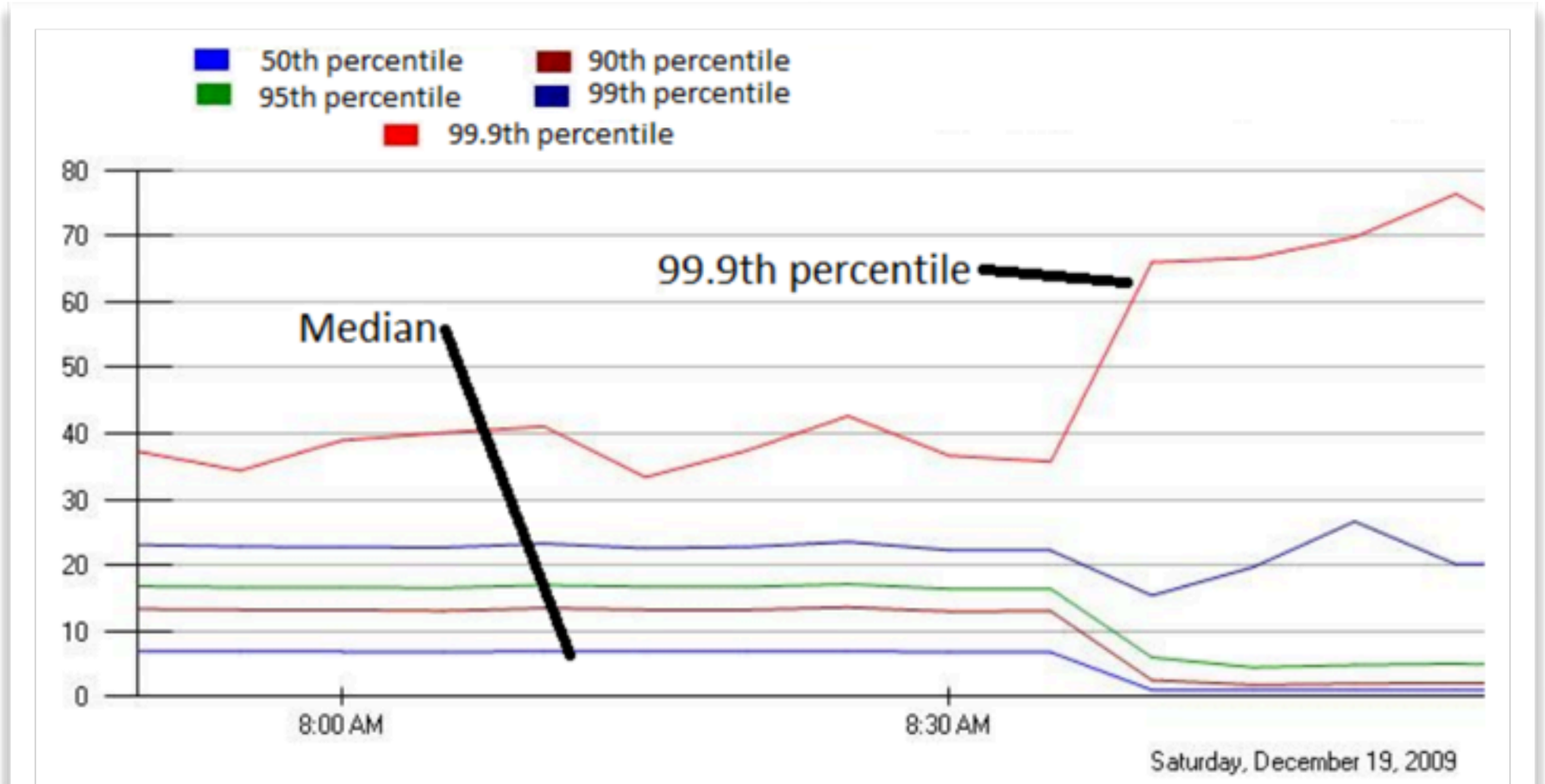
Causes?

- (Barrier) synchronized many-to-one traffic pattern
- Short flows (10s KB to 100s KB)
- Small queue buffer (4 to 8 MB shared memory)
- Large default RTO (300 ms)

Fixing TCP Incasts

- Use larger switch buffers
- Decrease RTOmin
- Desynchronize flows (random delay ~10ms)

Query completion time [ms]



Queue buildup and buffer pressure

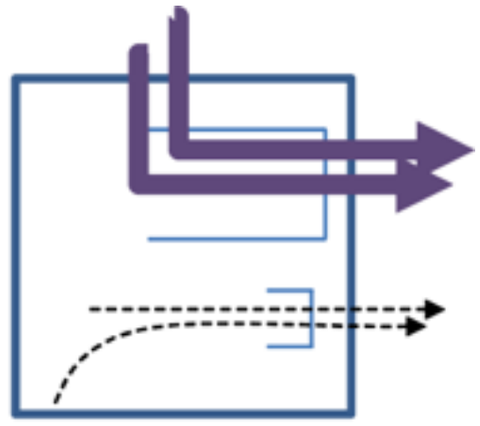
Causes: Long TCP flows occupy switch buffer

Queue buildup: short flow experiences increased delay

90%: RTT < 1ms --- (Bing's DC)

10%: 1 ms < RTT < 15 ms

Buffer pressure: 4 MB shared memory, i.e., how much buffer per port is not a constant



Many solutions to Incast do not apply here...

DCTCP

[Alizadeh et al., SIGCOMM'10]

(adapted from Alizadeh's slides)

DCTCP: Two goals

Goal #1: Low latency and high burst tolerance

- Ensuring low queue occupancy

Goal #2: Still having high throughput for long flows

- Using most of the network bandwidth

Achieve either goal is not hard; what's hard is to achieve both

Explicit Congestion Notification

Switches mark packet's ECN bit *before* buffer overflows

TCP sender treats ECN signals as if a single packet is dropped — but packets are not actually dropped

More useful for short flows — avoid packet drop, therefor avoid RTO timeout.

Well supported by today's commodity switches and end-hosts

DCTCP: Two Key ideas

1. React in proportion to the extent of congestion, not its presence

ECN Marks	TCP	DCTCP
1011110111	cut window by **50%**	cut window by **40%**
0000000001	cut window by **50%**	cut window by **5%**

2. Mark based on instantaneous queue length

   - Fast feedback to better deal with bursts

DCTCP Algorithm

Switch side:

- mark packet iff queue length > K

Sender side:

- maintain running avg of fraction of marked pkts

In each RTT:

$$F = \frac{\#\ of\ marked\ ACKs}{Total\ \#\ of\ ACKs} \qquad \alpha \leftarrow (1-g)\alpha + gF$$

- adaptive window decreases: $Cwnd \leftarrow (1 - \frac{\alpha}{2})Cwnd$

Why does it work?

Small buffer occupancies

→ bursts fit

→ low queueing delay

Aggressive marking when queue buffer builds up

→ fast react before packet drops

Adaptive window reduction

→ high throughput

Discussion

- Can we leverage more capability and information in date center environment? Switch features? Traffic patterns? etc..

- Can we use DCTCP in wide area networks?

- Can we use other switch features to improve the performance?

- Short flow performance in general settings

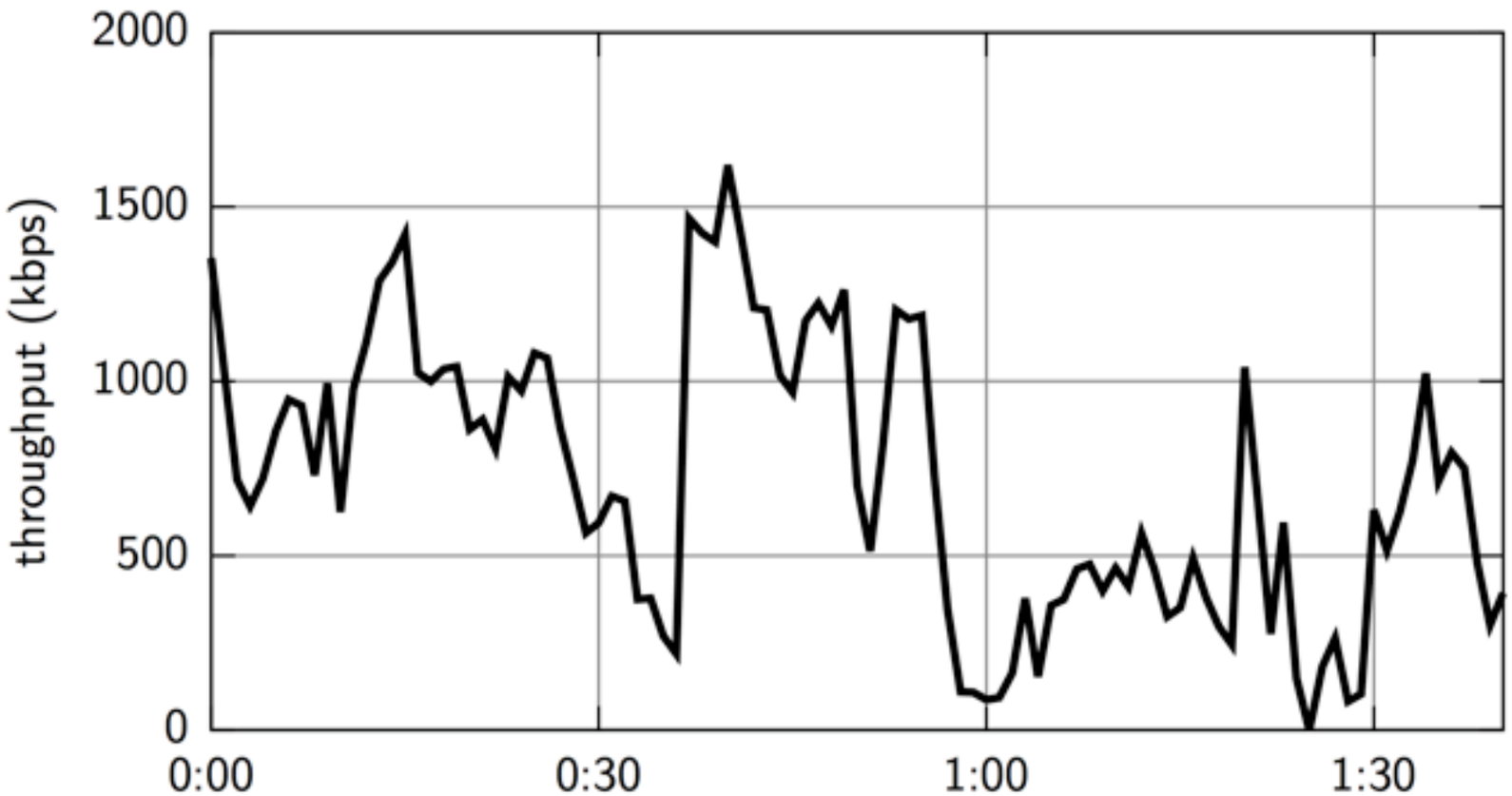
Sprout: Stochastic Forecasts Achieve High Throughput and Low Delay over Cellular Networks

[Winstein et al., NSDI'13]

(adapted from Winstein's slides)

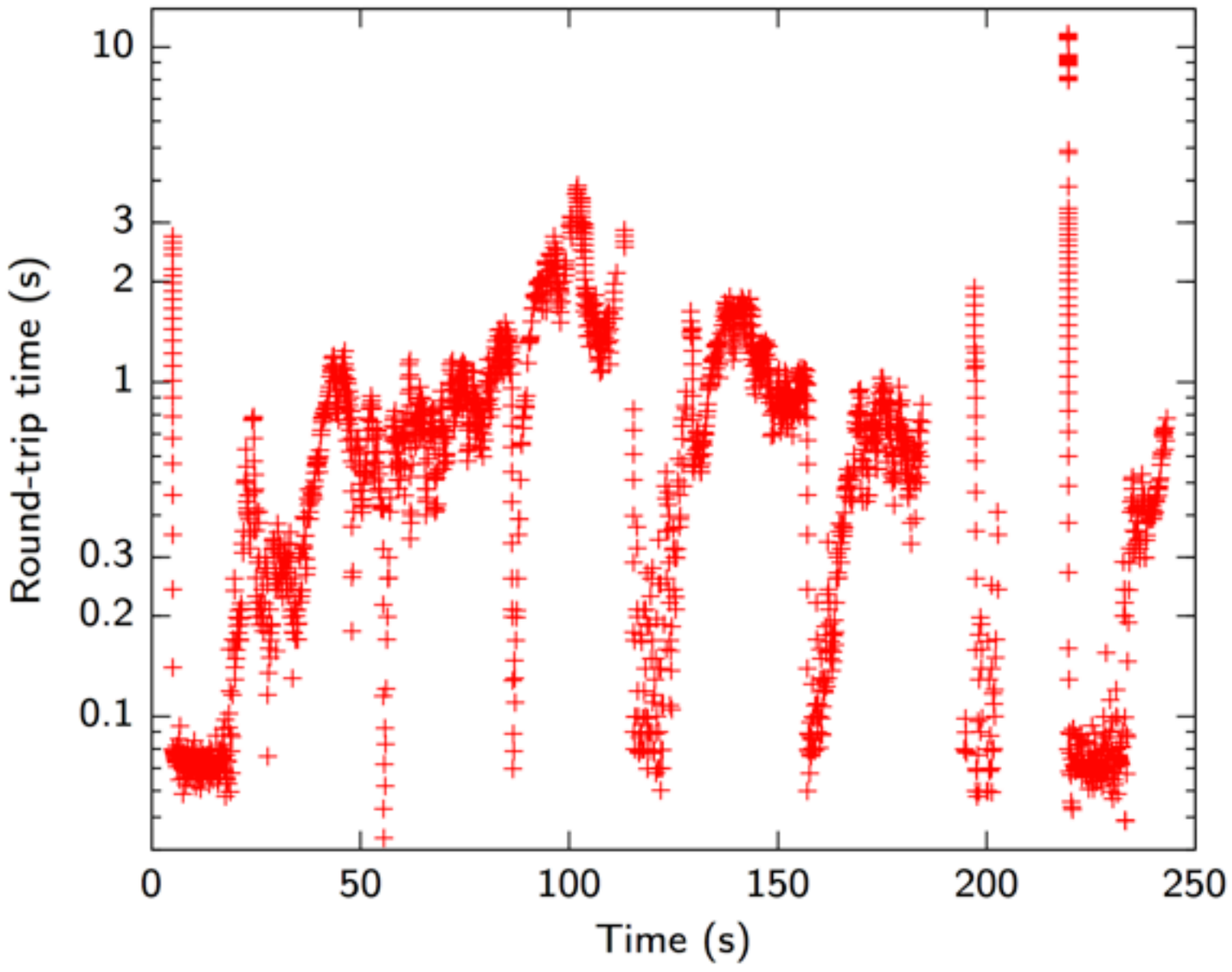
Cellular Networks

- Highly dynamic network condition



Cellular Networks

- Dedicated channel abstraction without packet loss



Cellular Networks

- TCP works poorly because
  - Existing schemes react to congestion signals.
    - Packet loss.
    - Increase in round-trip time.
  - Feedback comes too late.
  - The killer: **self-inflicted queueing delay.**

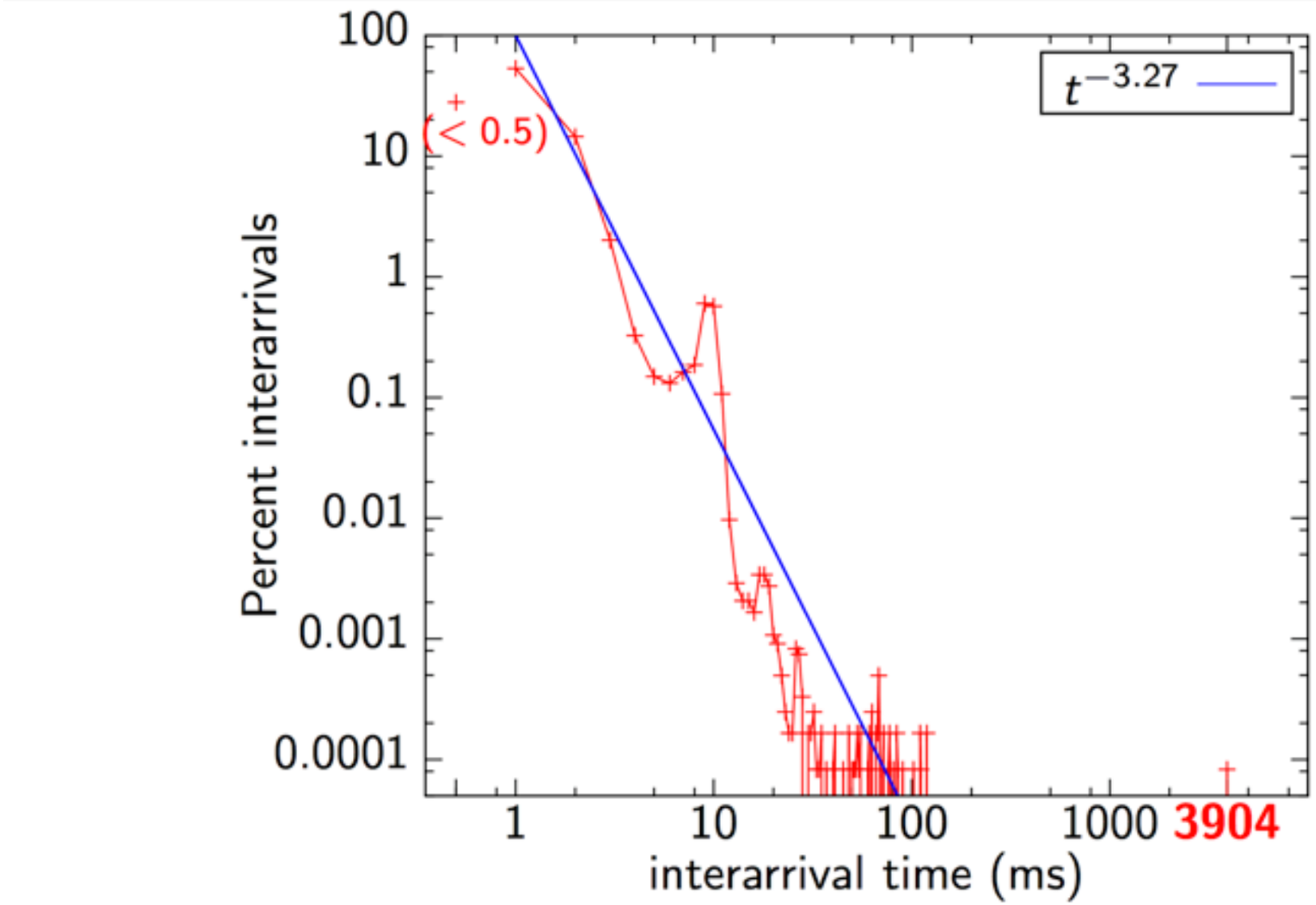
Cellular Networks

- Can we fix TCP to achieve
  - Most throughput
  - Bounded risk of delay > 100 ms

Sprout

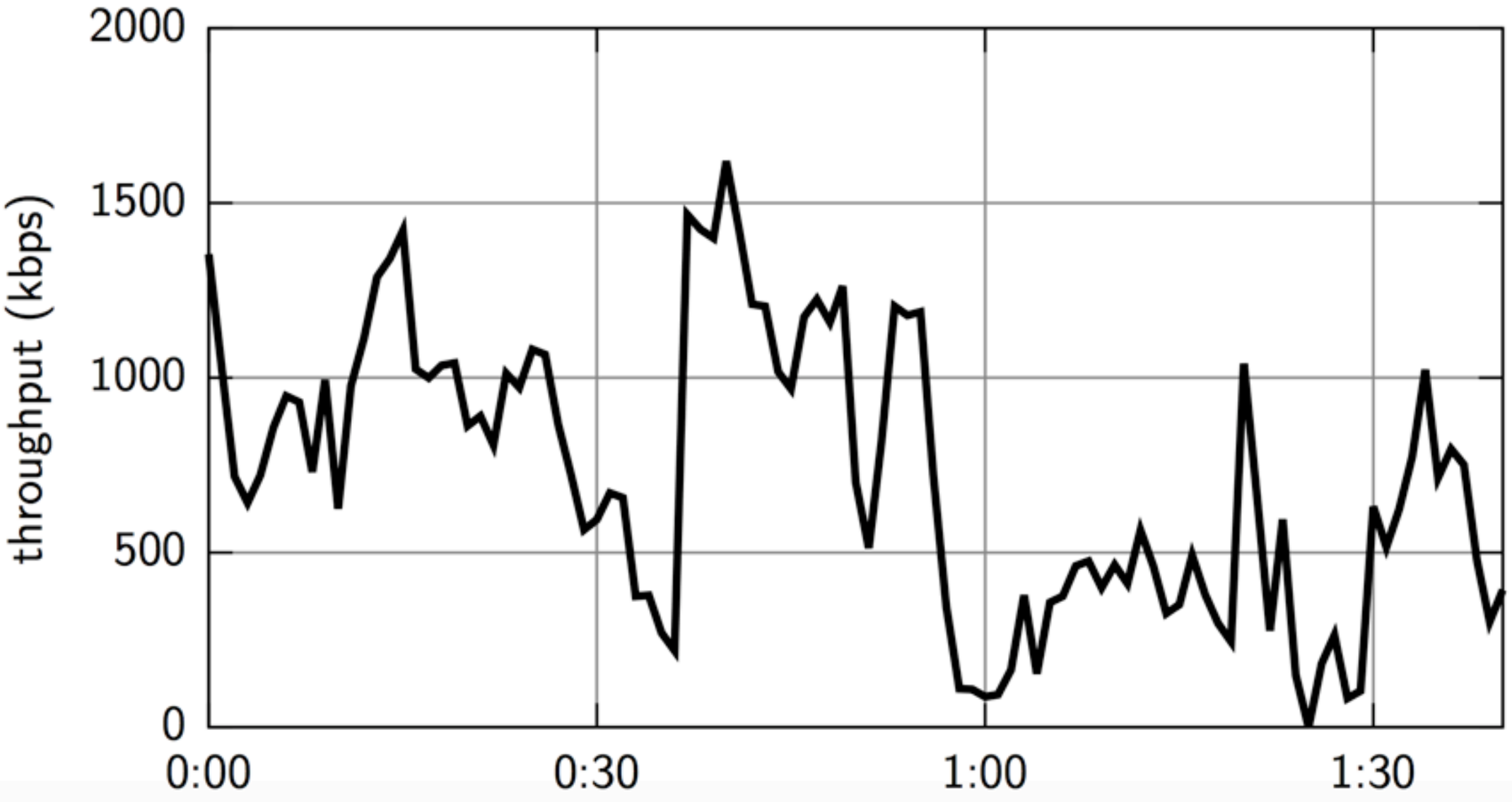
- **Model** variation in link speed
- **Infer** current link speed
- **Predict** future link speed
  - Don’t wait for congestion
- **Control**: Send as much as possible, but require:
  - 95% chance all packets arrive within 100 ms

Model packet deliveries looks like flicker noise

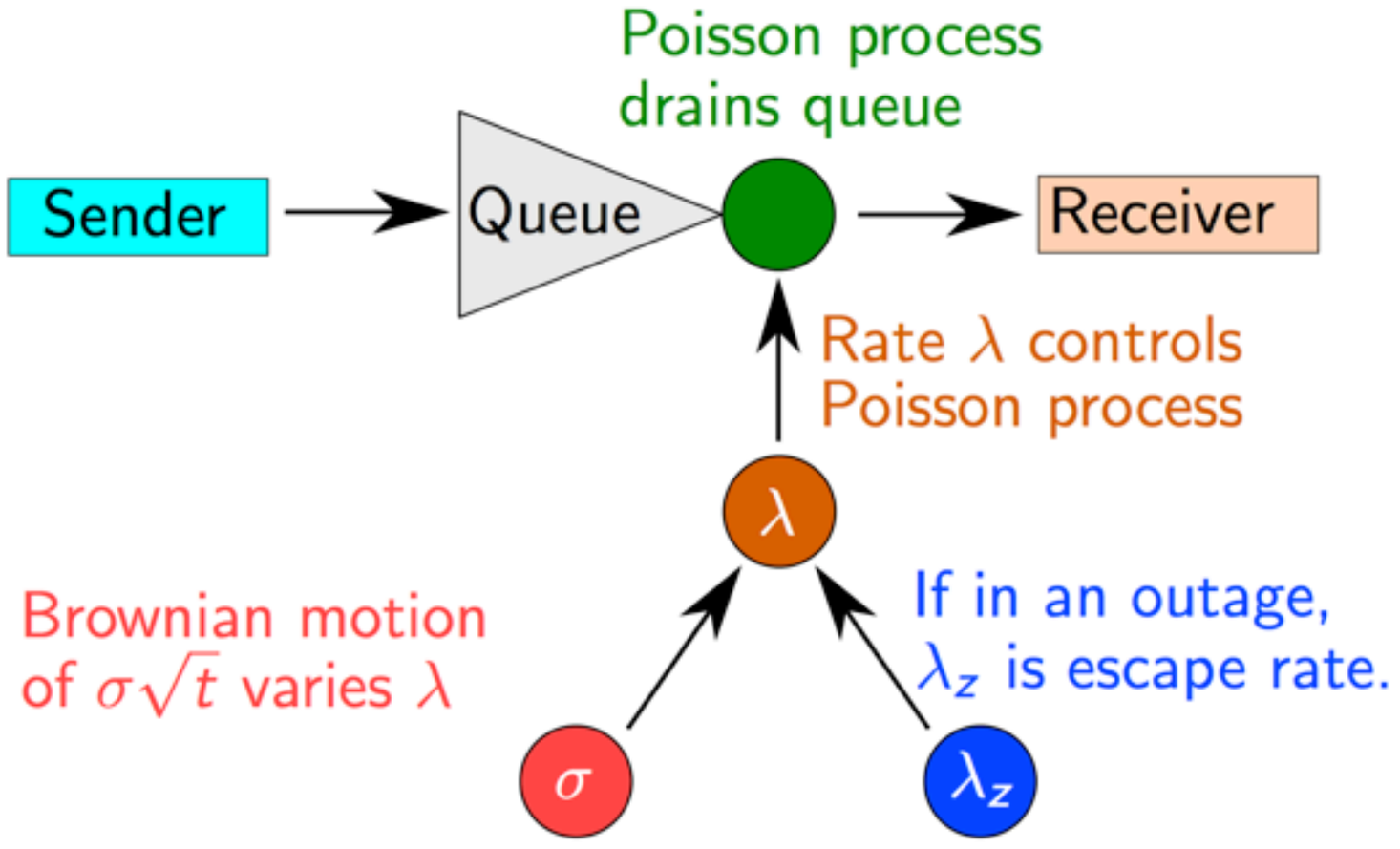


(Verizon LTE, phone stationary.)

Model: average rate looks like random walk



Sprout: Model Cellular Networks



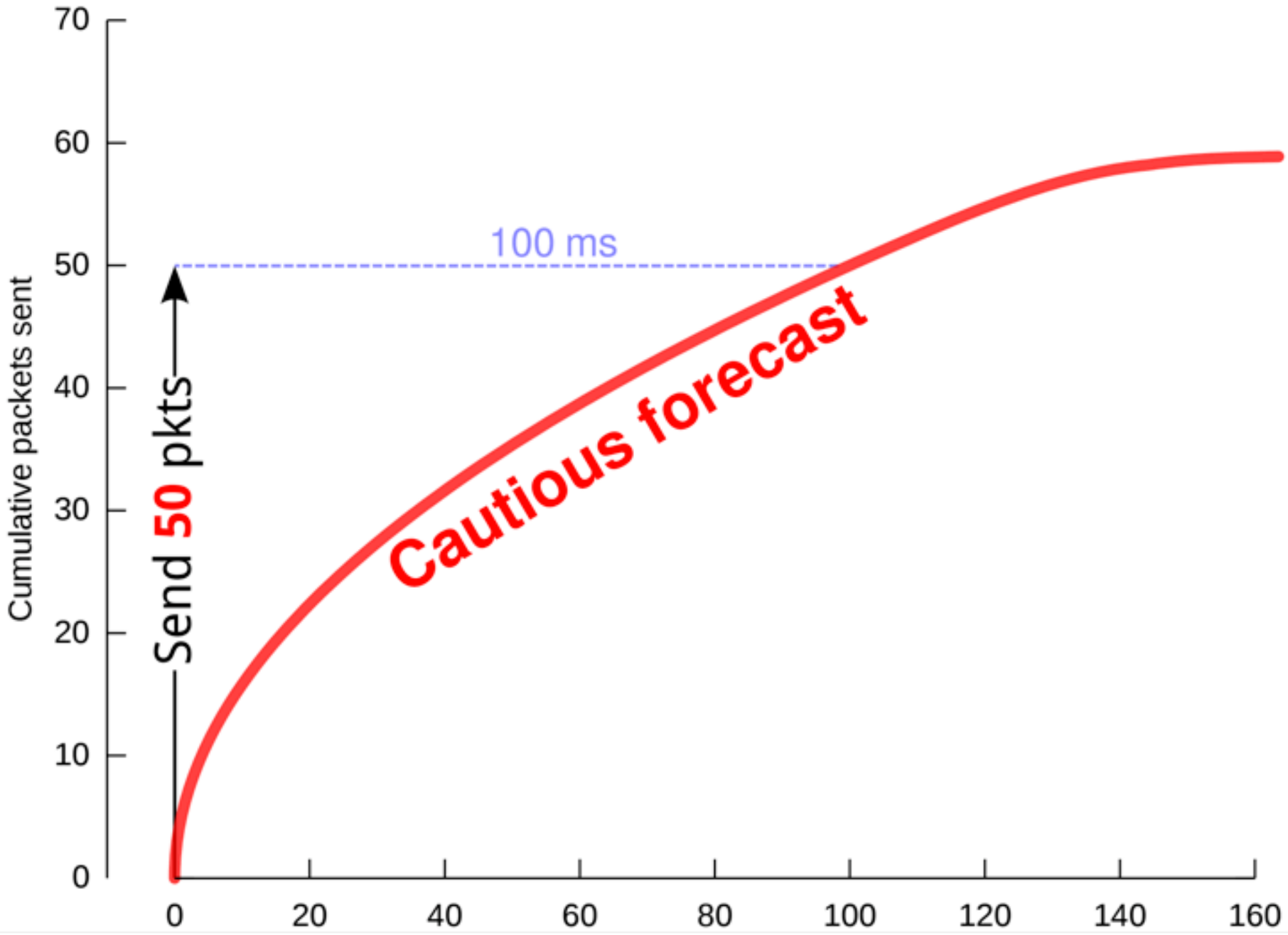
Infer: Current Link Speed

- **Observe** packets received every τ
  - receiver feedback
- **Update** P(λ)

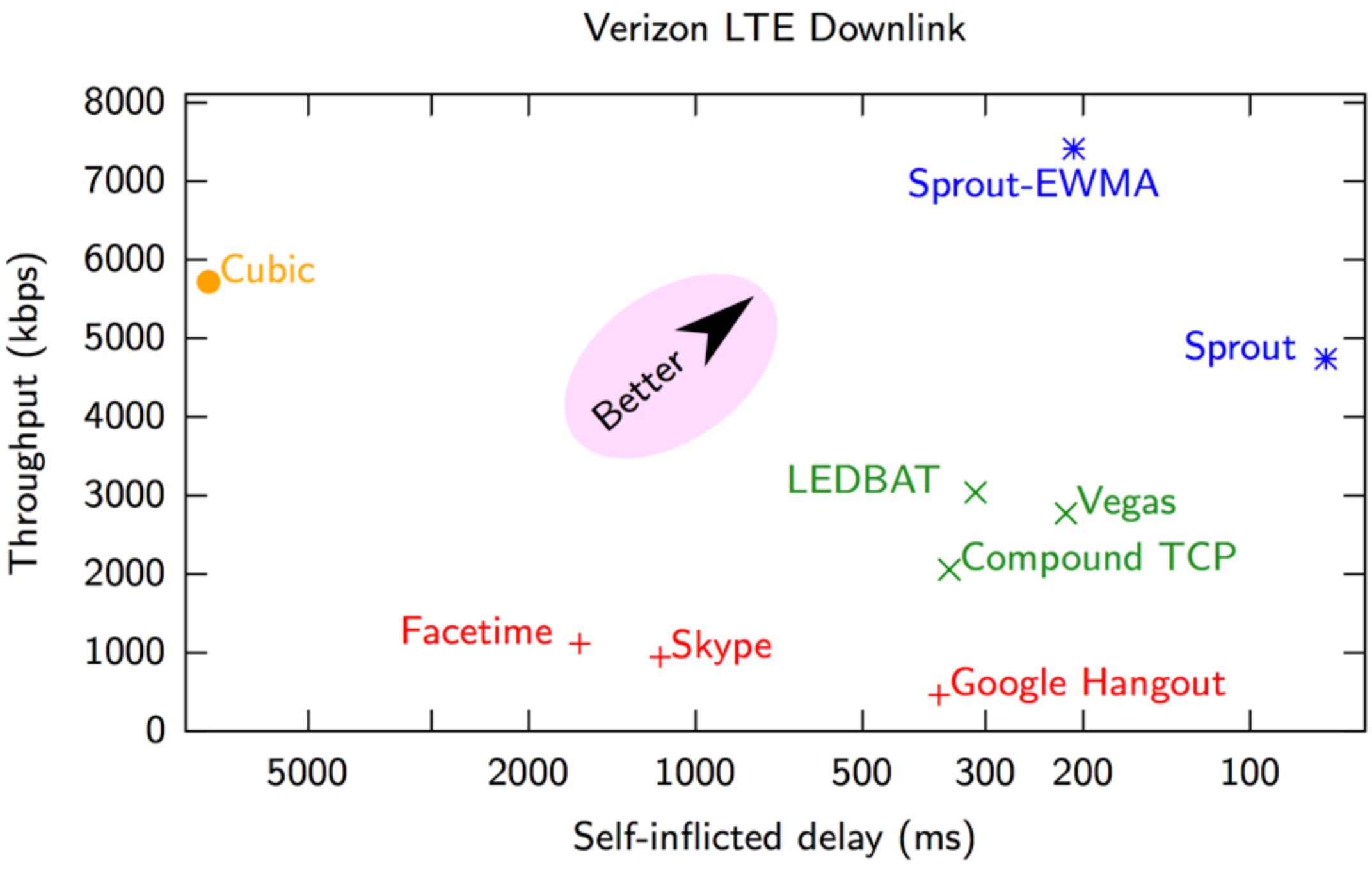
Predict: future link speed

- **Evolve** model forward
  - receiver feedback
- **Predict** with a certain horizon

Control: fill up 100 ms forecast window



Evaluation: LTE Verizon Downlink



Discussion:

High BDP	Wireless	Satellite	Inter-DC	Intra-DC
BIC H-TCP Compound CUBIC FAST TCP	Westwood Vegas Veno	Hybla STAR	Illinois SABUL	DCTCP
10X	10X	17X	4X	

Unstable, RTT Unfair, Bufferbloat, Crash on Changing Networks, ……

Point Solutions

+

Performance

Far from Optimal

Possible Answer No.2

Replace human from the loop

TCP

ex machina

[Winstein et al., SIGCOMM'13]

http://exmachina-movie.com/

Machine learning based CC

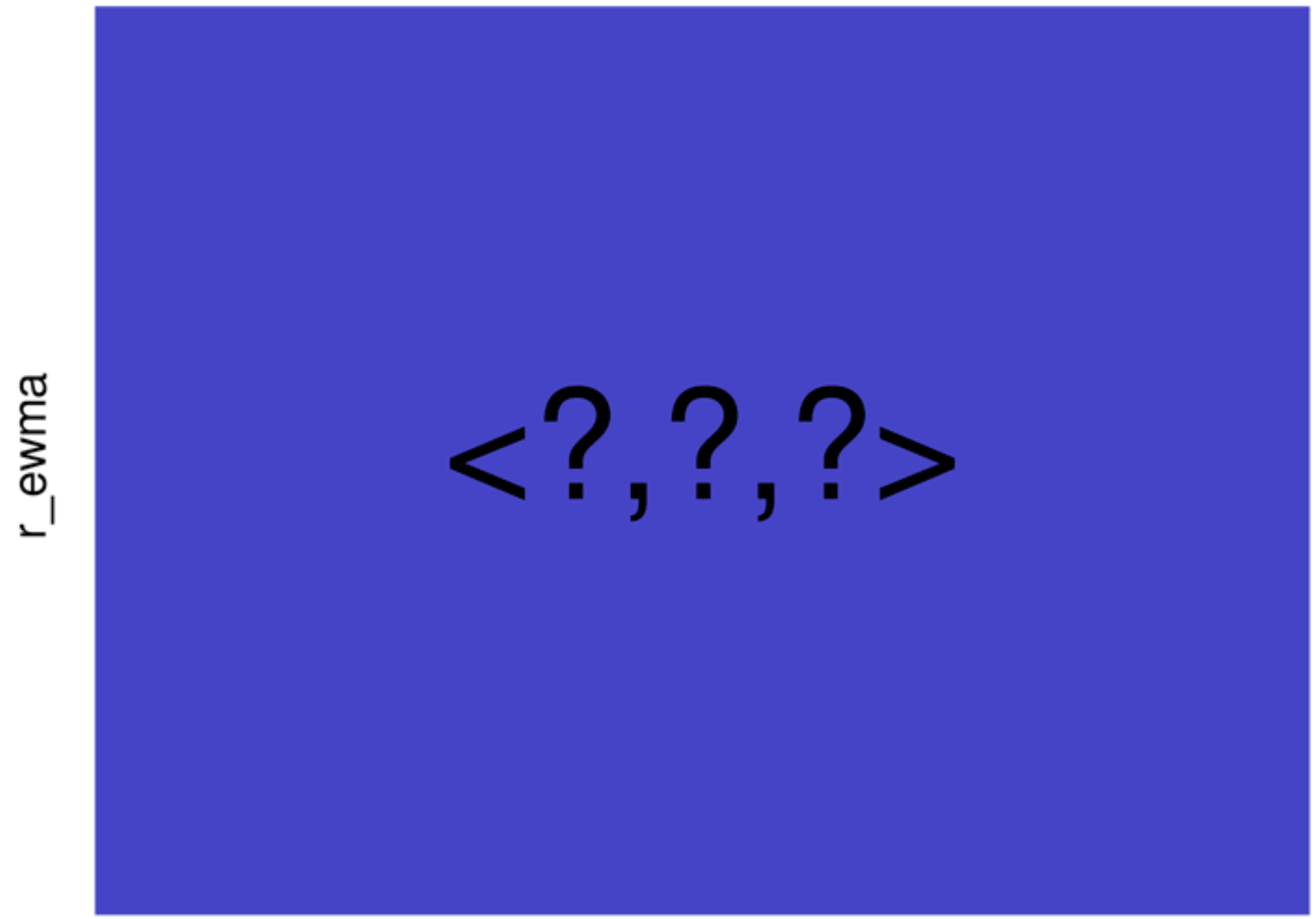
- Given a range of possible network conditions
  - Bandwidth, RTT, number of senders
- Using a set of congestion control signal
  - r_ewma, s_ewma, rtt_ratio

Machine learning based CC

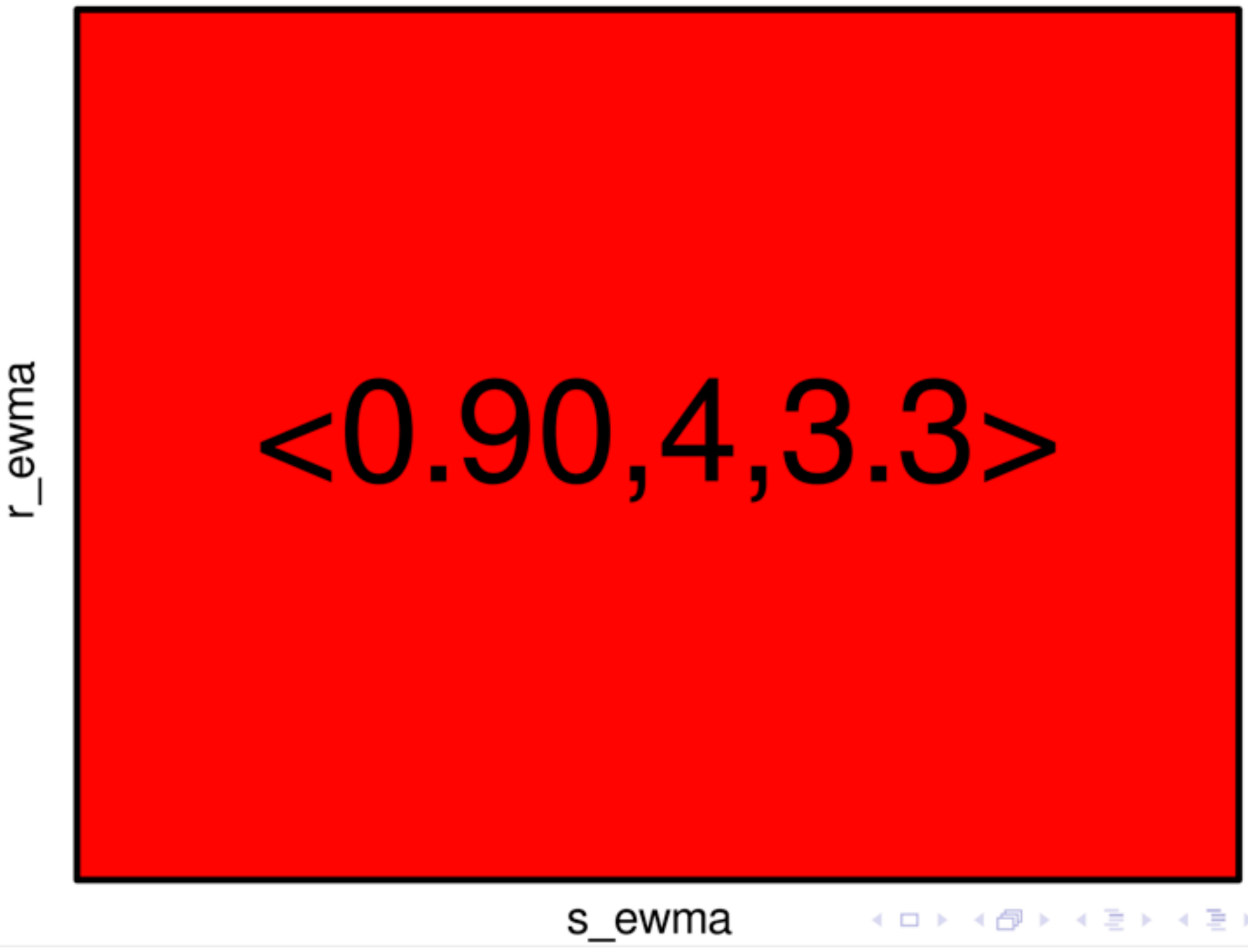
- Use offline machine learning to train a map
  - Rule(r_ewma, s_ewma, rtt_ratio) → <m, b, τ>

m	Multiple to congestion window
b	Increment to congestion window
τ	Minimum interval between two outgoing packets

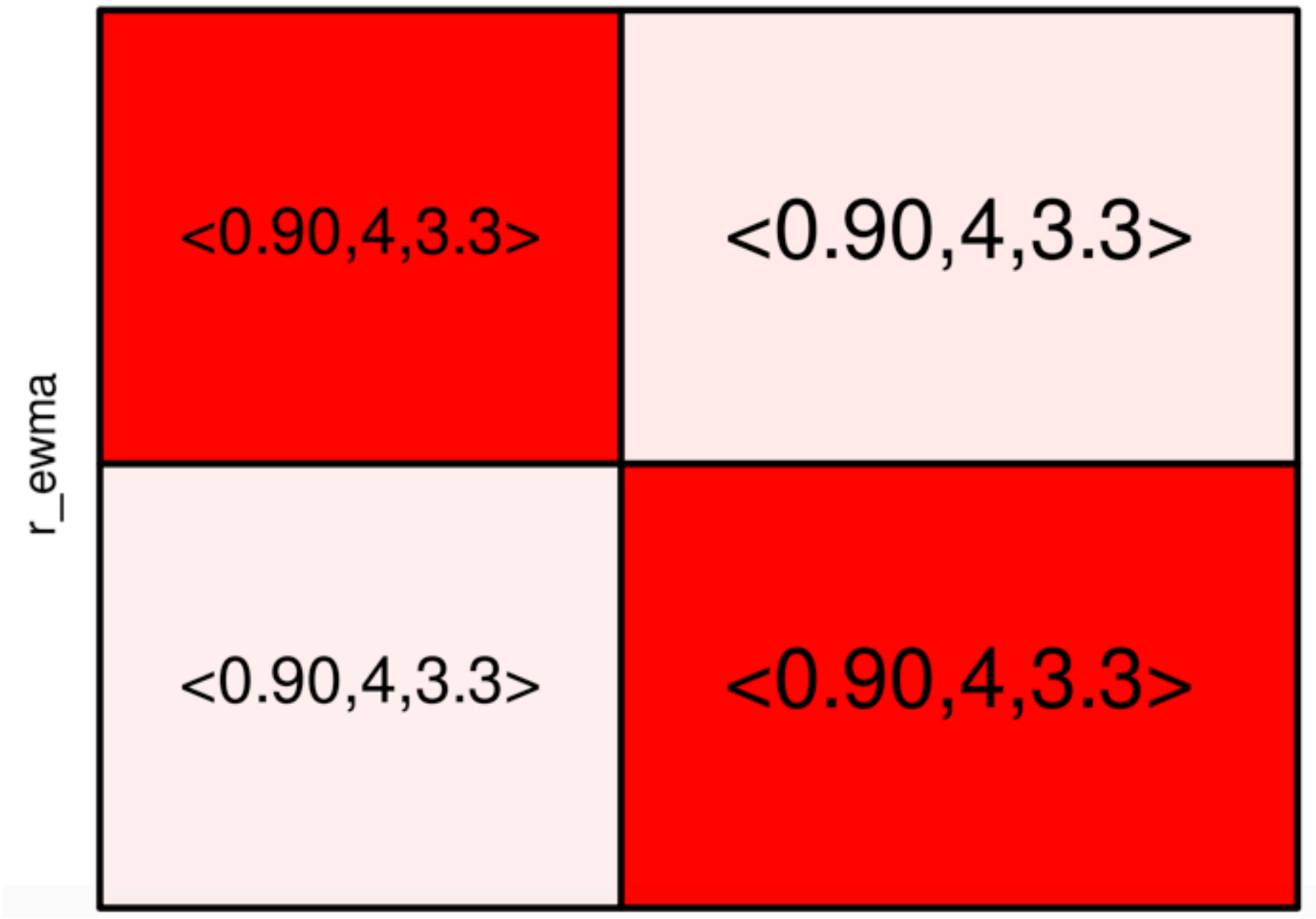
One action for all state



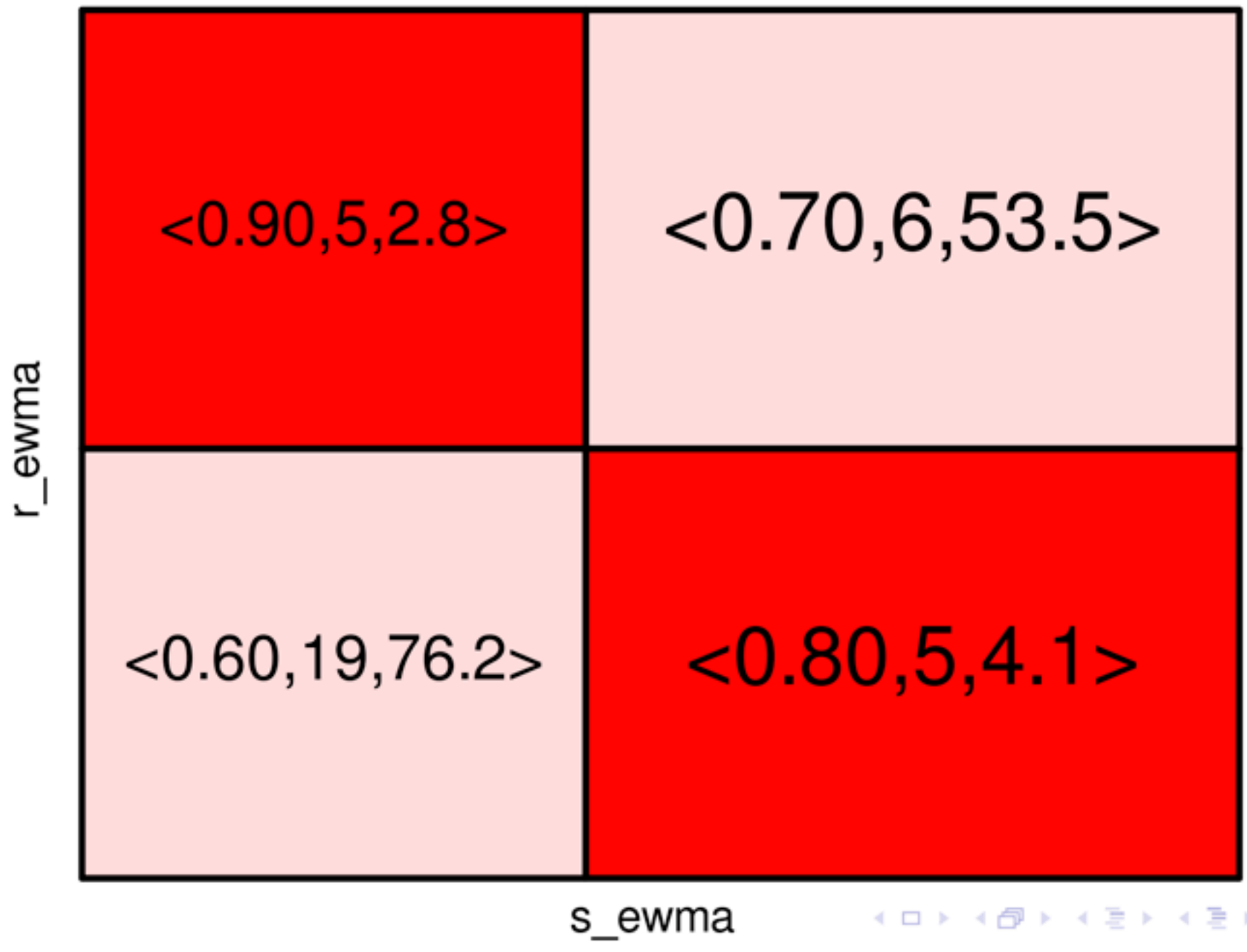
The best single action, split on median



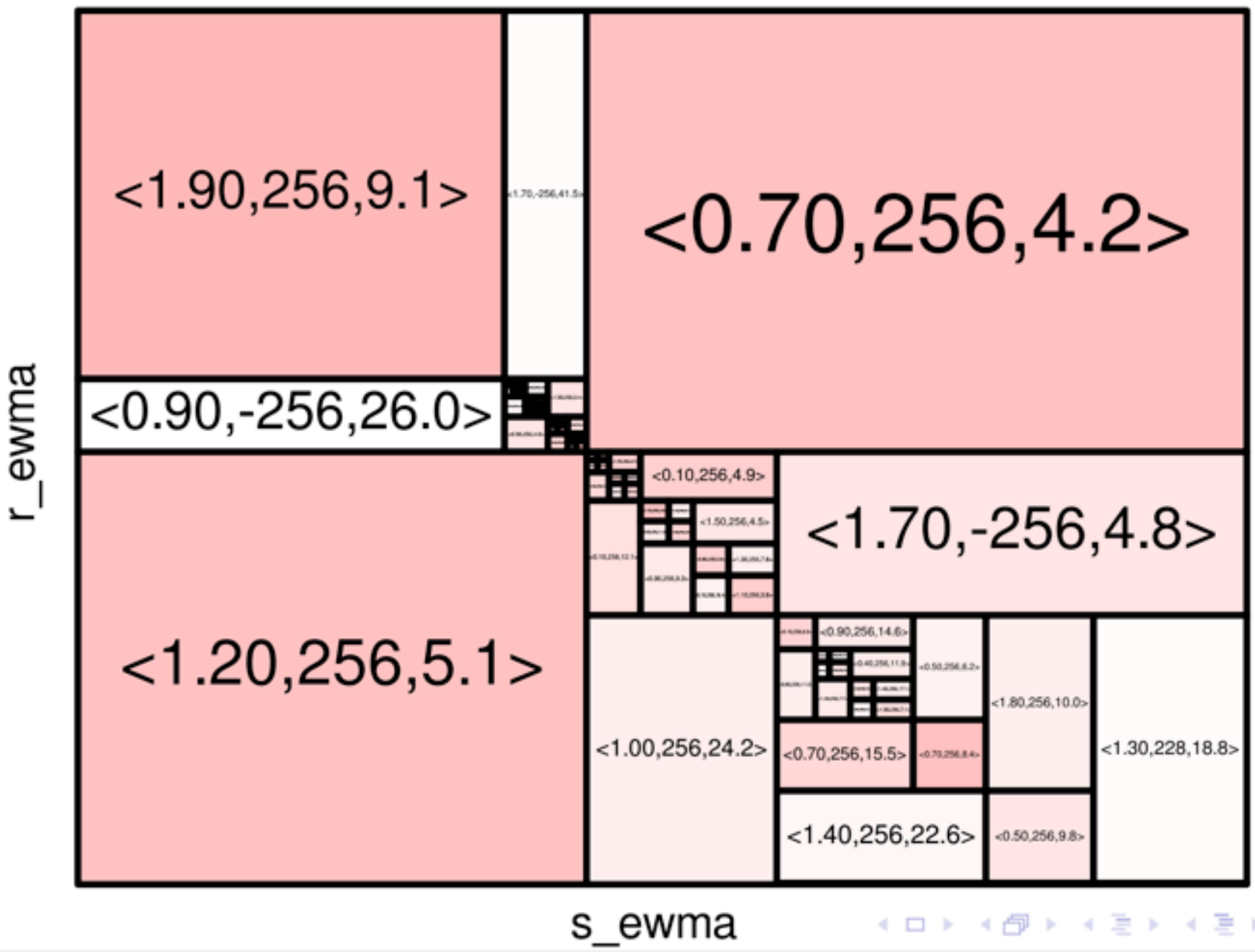
Optimize for each sub actions



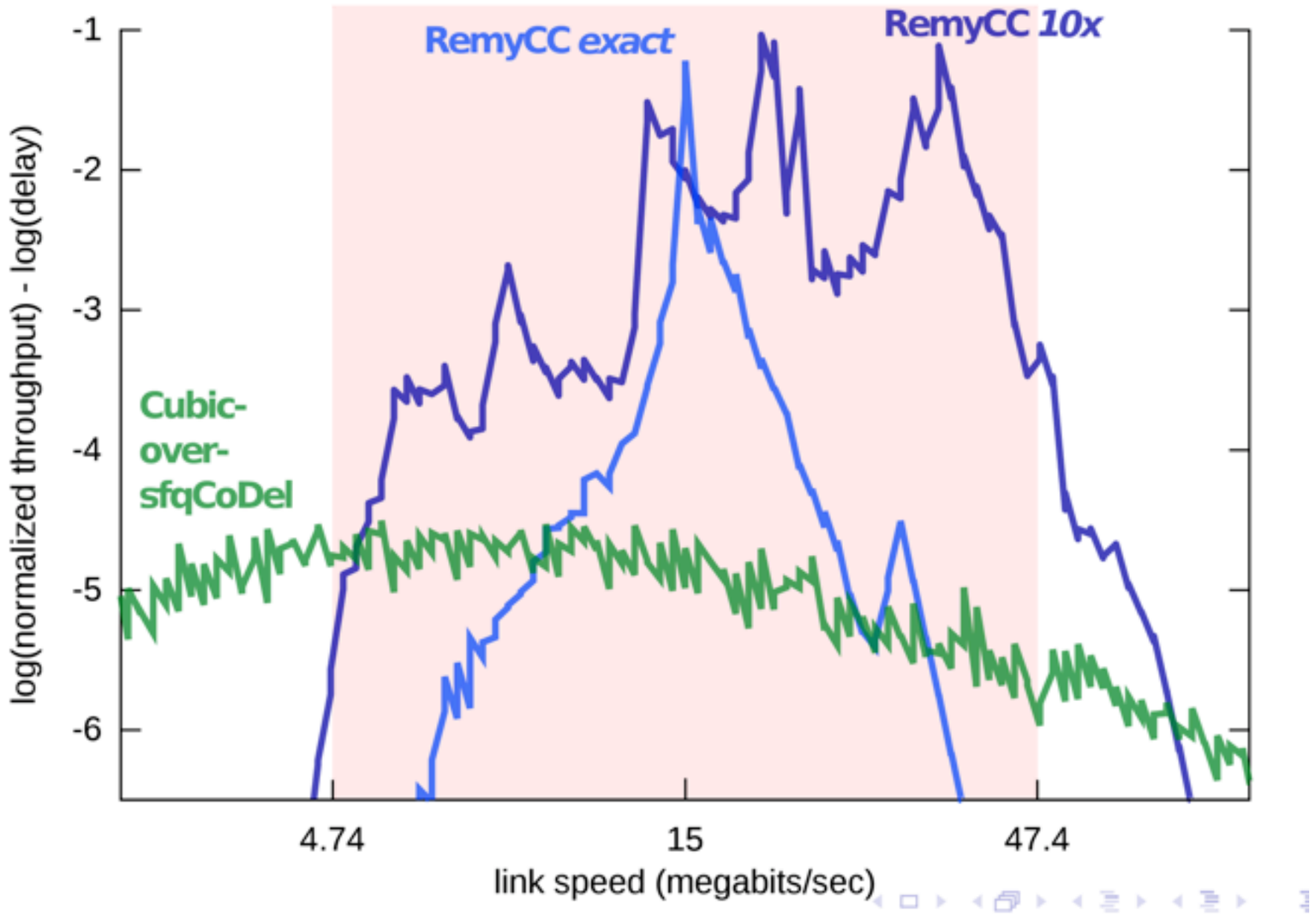
Split the most used rule



Iterate


<1.90,256,9.1>
<0.70,256,4.2>
<0.90,-256,26.0>
<0.10,256,4.9>
<1.50,256,4.5>
<1.70,-256,4.8>
<1.20,256,5.1>
<1.00,256,24.2>
<0.90,256,14.6>
<0.50,256,8.2>
<1.80,256,10.0>
<1.30,228,18.8>
<0.70,256,15.5>
<1.40,256,22.6>
<0.50,256,9.8>
r_ewma
s_ewma

Discussion

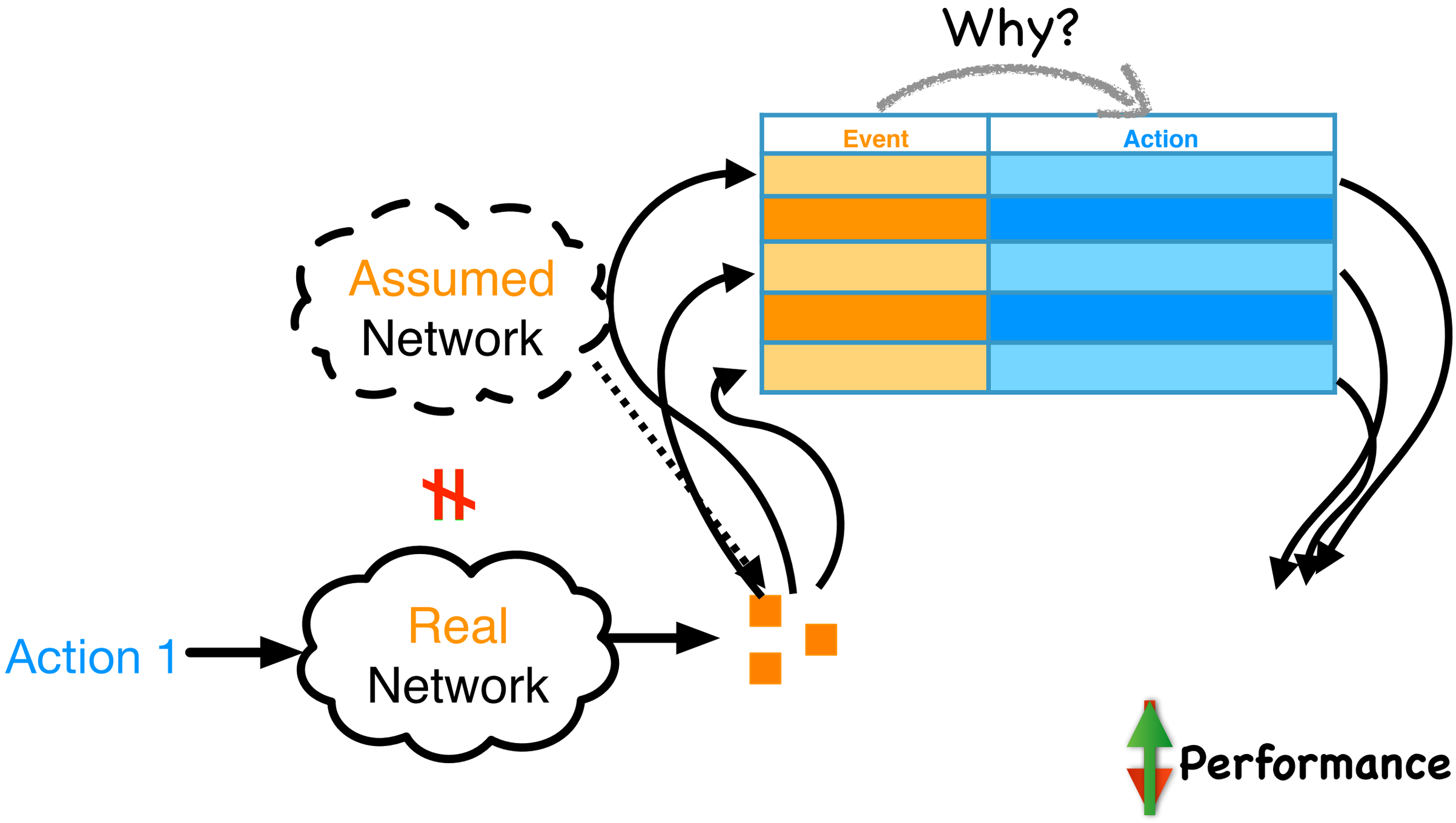

RemyCC exact
RemyCC 10x
Cubic-
over-
sfqCoDel
log(normalized throughput) - log(delay)
-1
-2
-3
-4
-5
-6
4.74
15
47.4
link speed (megabits/sec)

Possible Answer No.3

TCP's Architecture Sucks

Hardwired Mapping

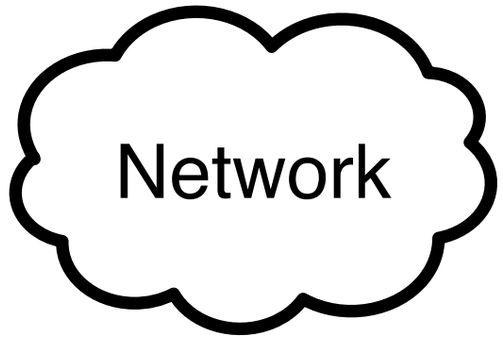
	Event	Action
Reno		
Scalable		
CUBIC		
FAST		
HTCP		


Why?
Event
Action
Assumed
Network
Real
Network
Action 1
Performance

Flow f sends at R

Event	Action
Pac	Dec R a lot Maintain R Increase R

No event-control mapping optimal for all network scenarios



f causes most congestion

fer

other high rate flow causing congestion

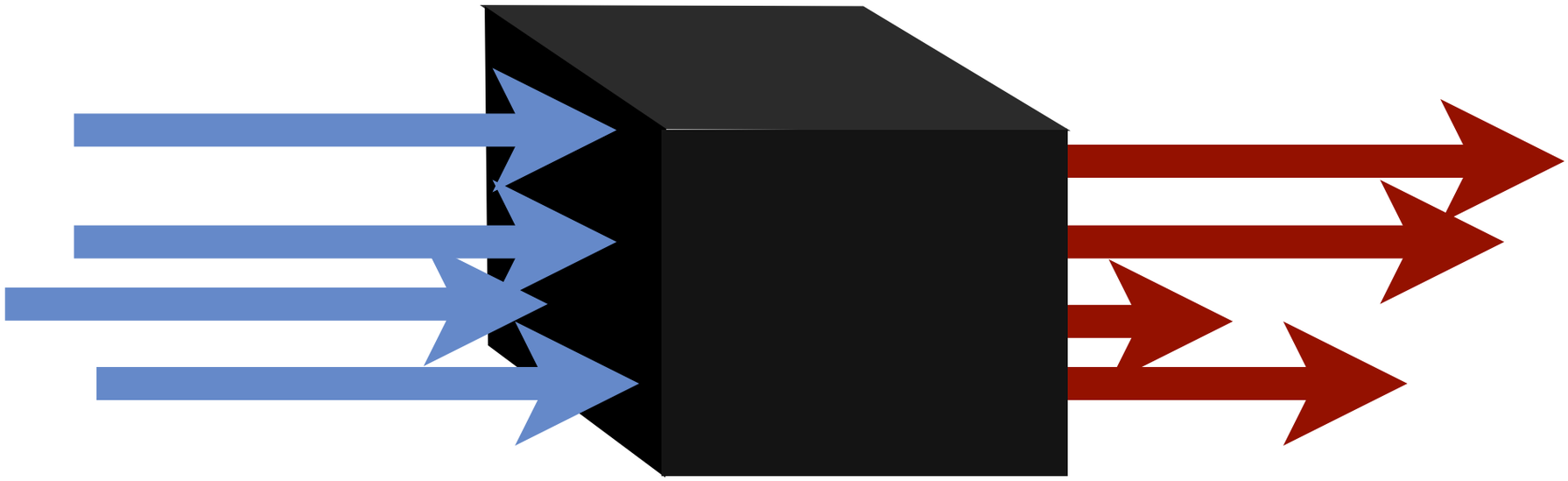
loss is random

PCC

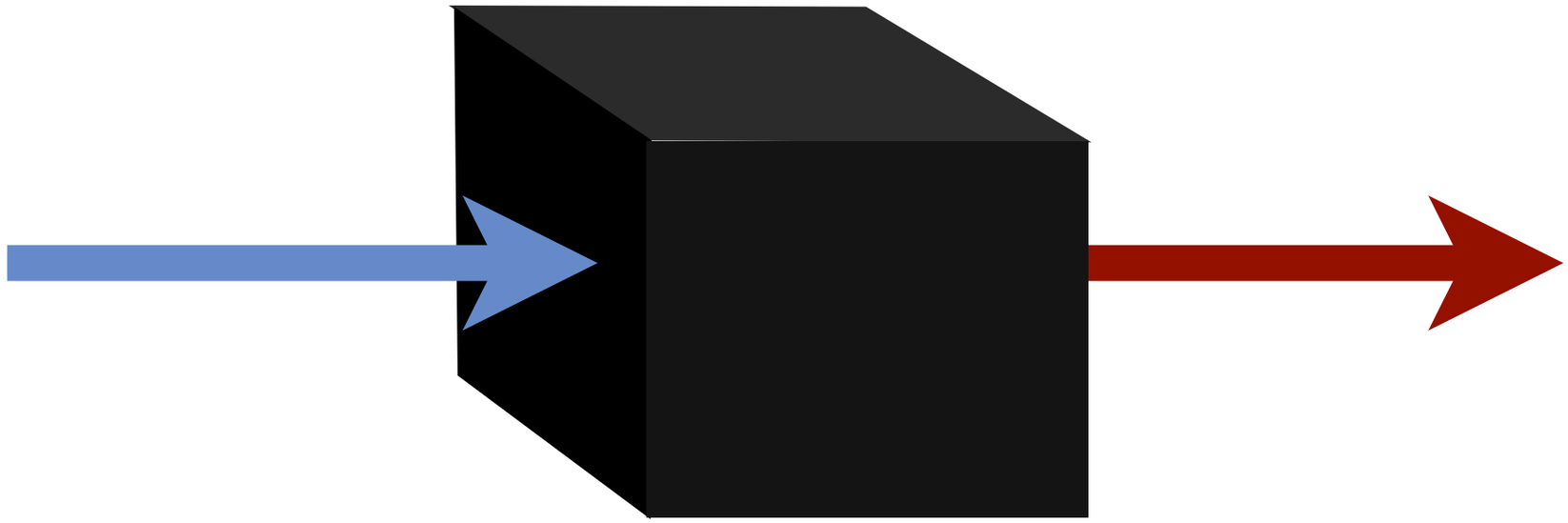
[Dong et al., NSDI'15]

(adapted from Dong's slides)

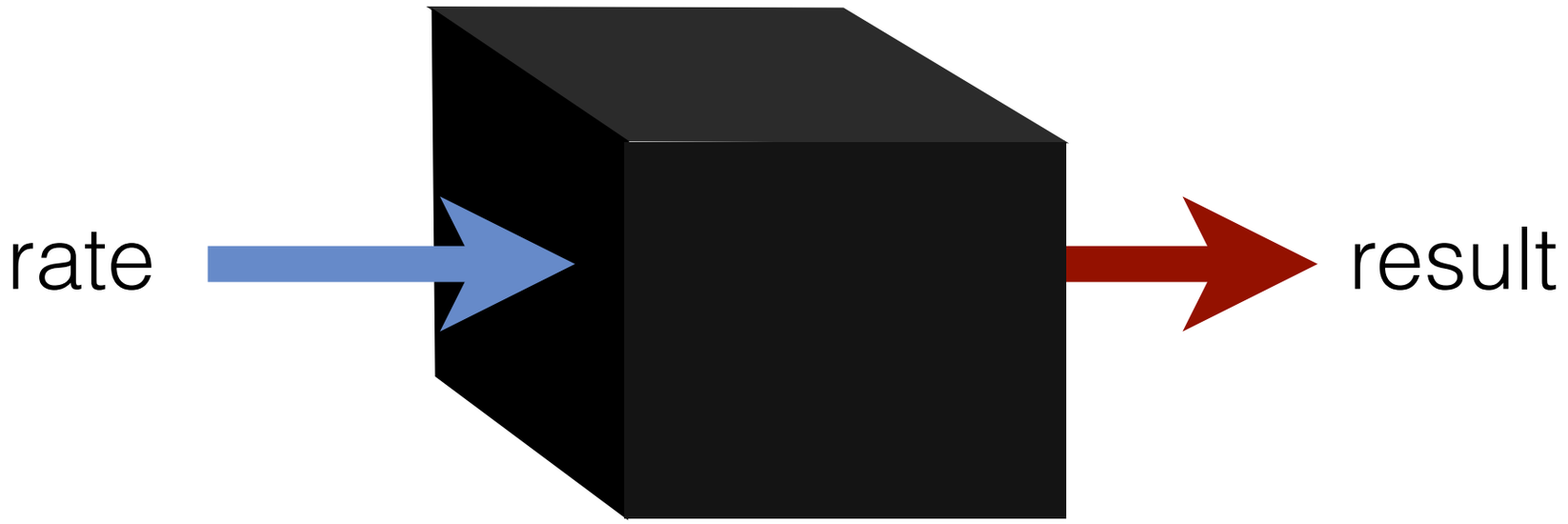
What is the right rate to send?



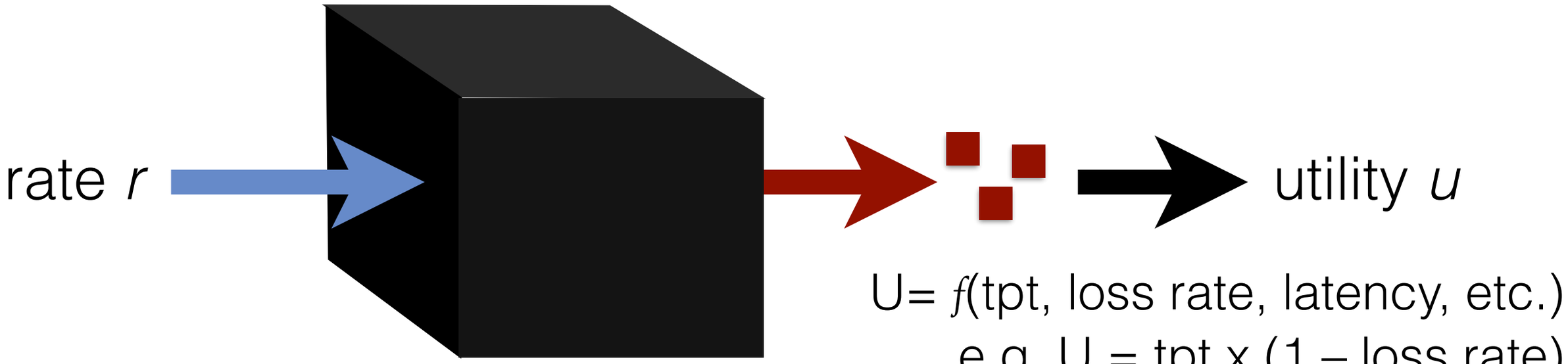
What is the right rate to send?



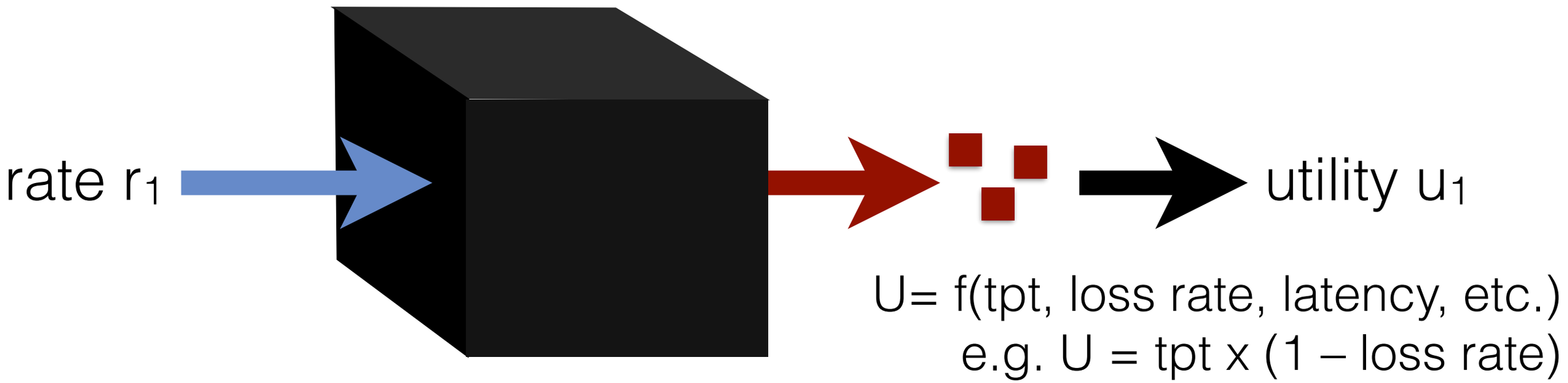
What is the right rate to send?



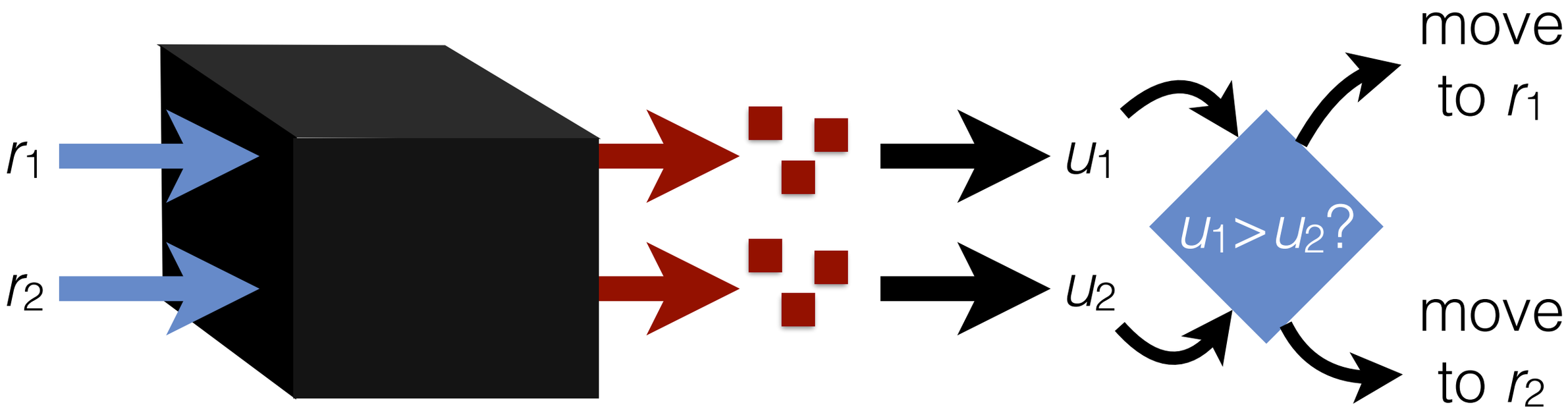
What is the right rate to send?



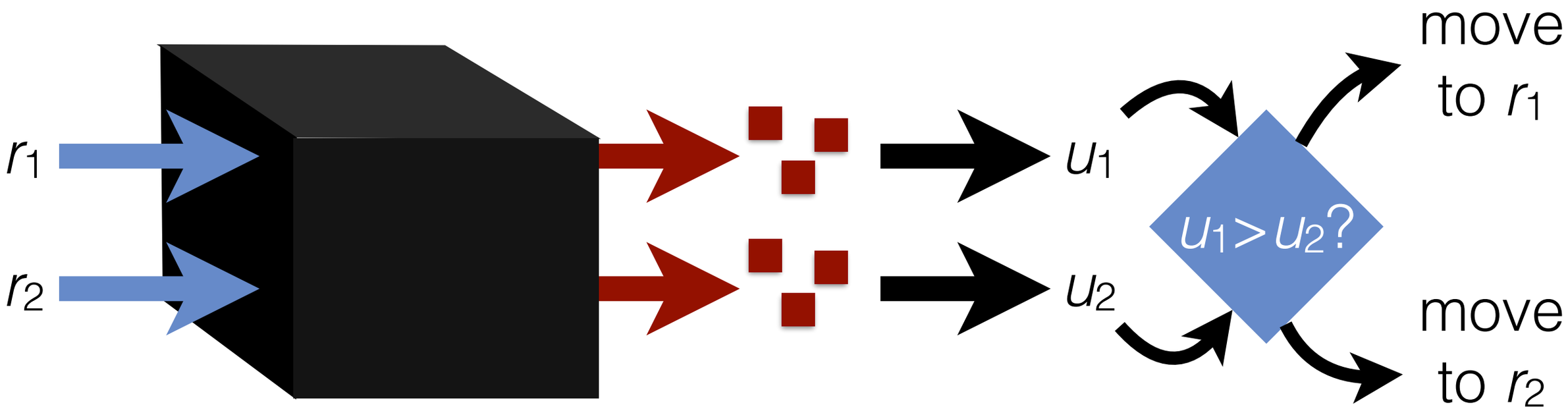
What is the right rate to send?



No matter how complex the network,
rate r —> utility u


r_1
r_2
u_1
u_2
$u_1 > u_2$?
move to r_1
move to r_2

Performance-oriented Congestion Control



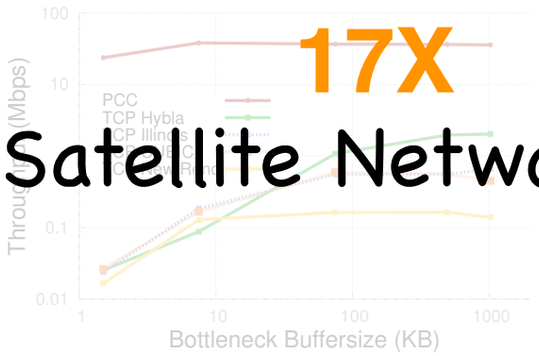
Observe real performance	Control based on empirical evidence	yields **Consistent high performance**

Consistent High Performance

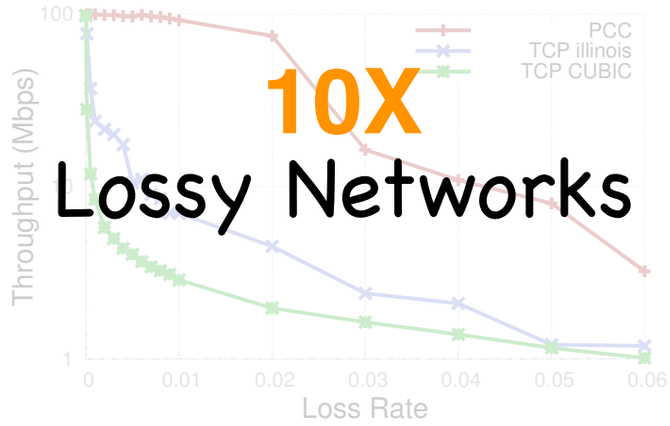
Table 1: PCC significantly outperforms TCP in inter-data center environments. RTT in msec; throughput in Mbps.

Transmission Pair	RTT	PCC	SABUL	CUBIC	Illinois
GPO → NYSERNet	12.1	791	674	129	326
GPO → Missouri	47.4	808	661	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

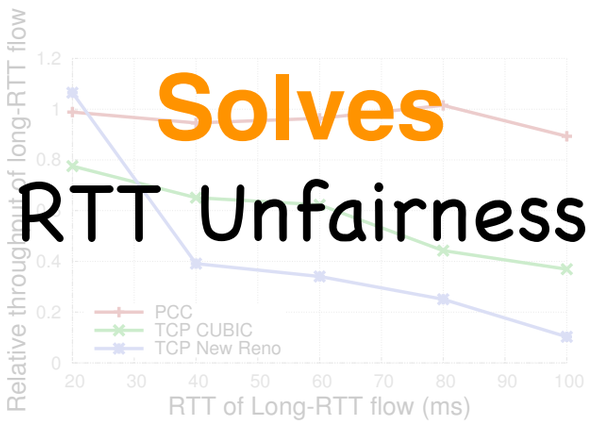
4X
InterDC



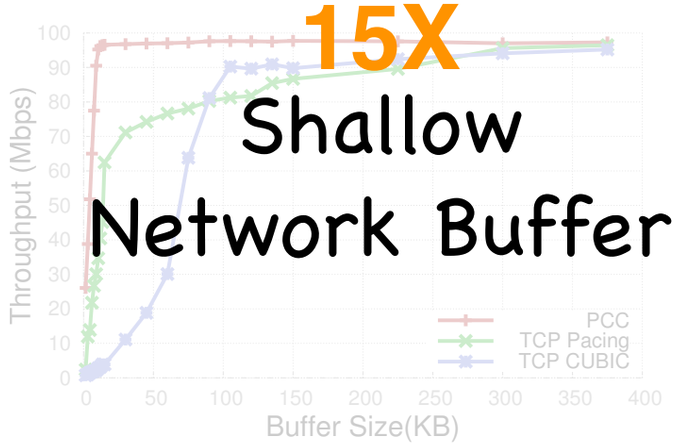
17X
Satellite Networks



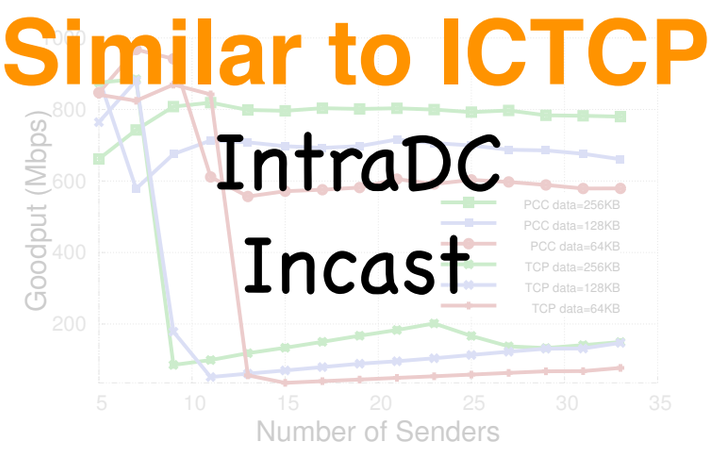
10X
Lossy Networks



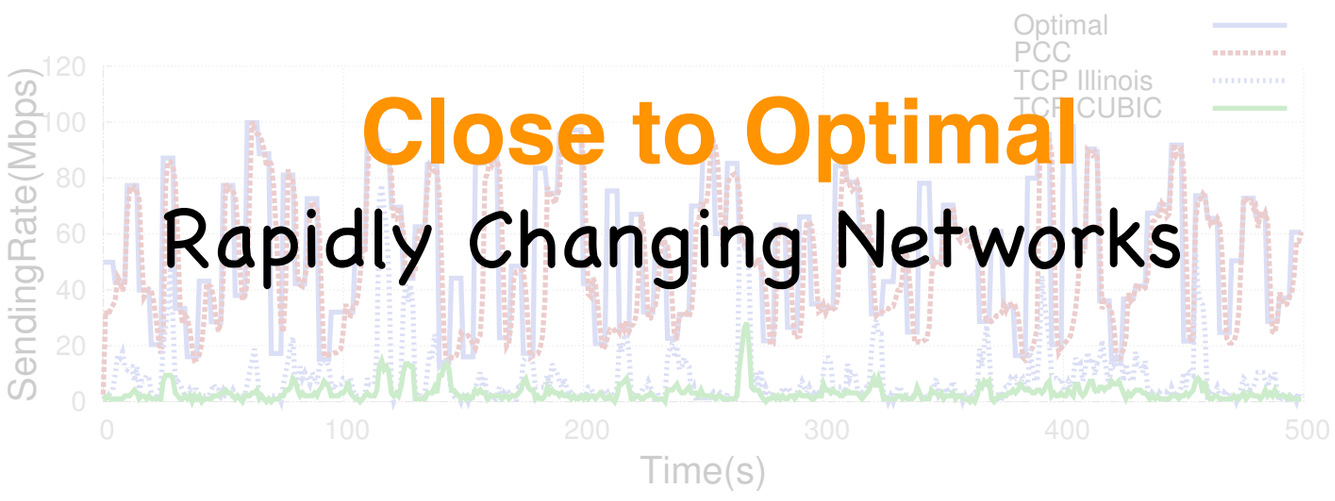
Solves
RTT Unfairness



15X
Shallow
Network Buffer



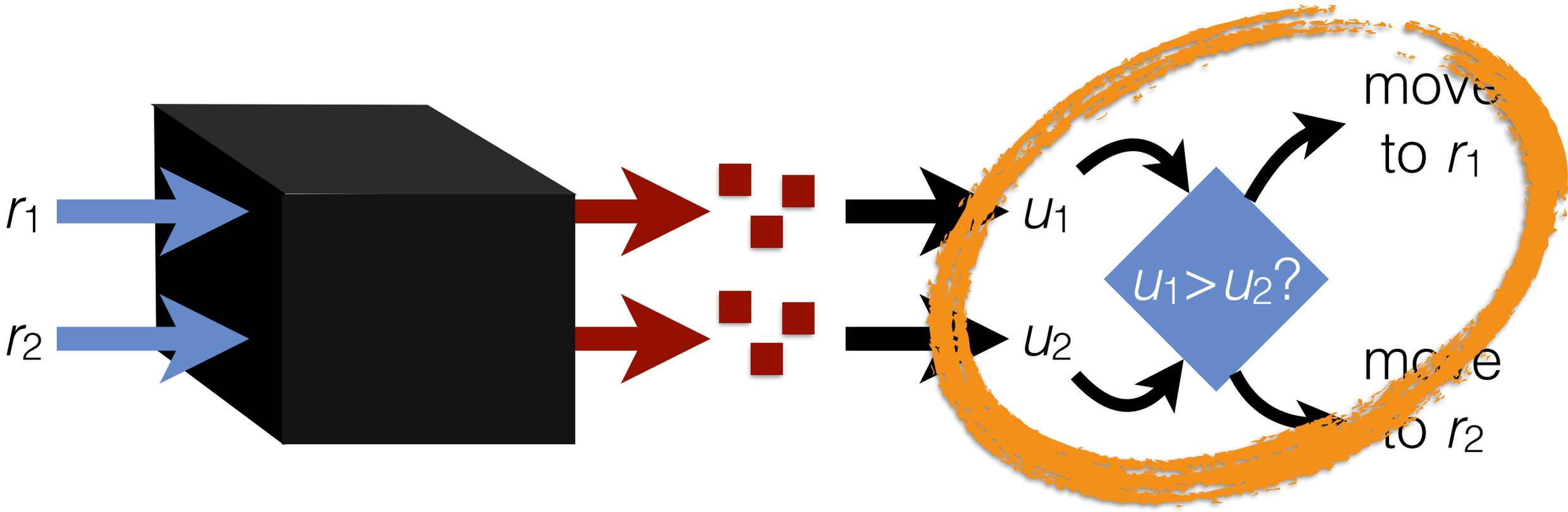
Similar to ICTCP
IntraDC
Incast



Close to Optimal
Rapidly Changing Networks

5X in median
Global Commercial
Internet

Where is Congestion Control?

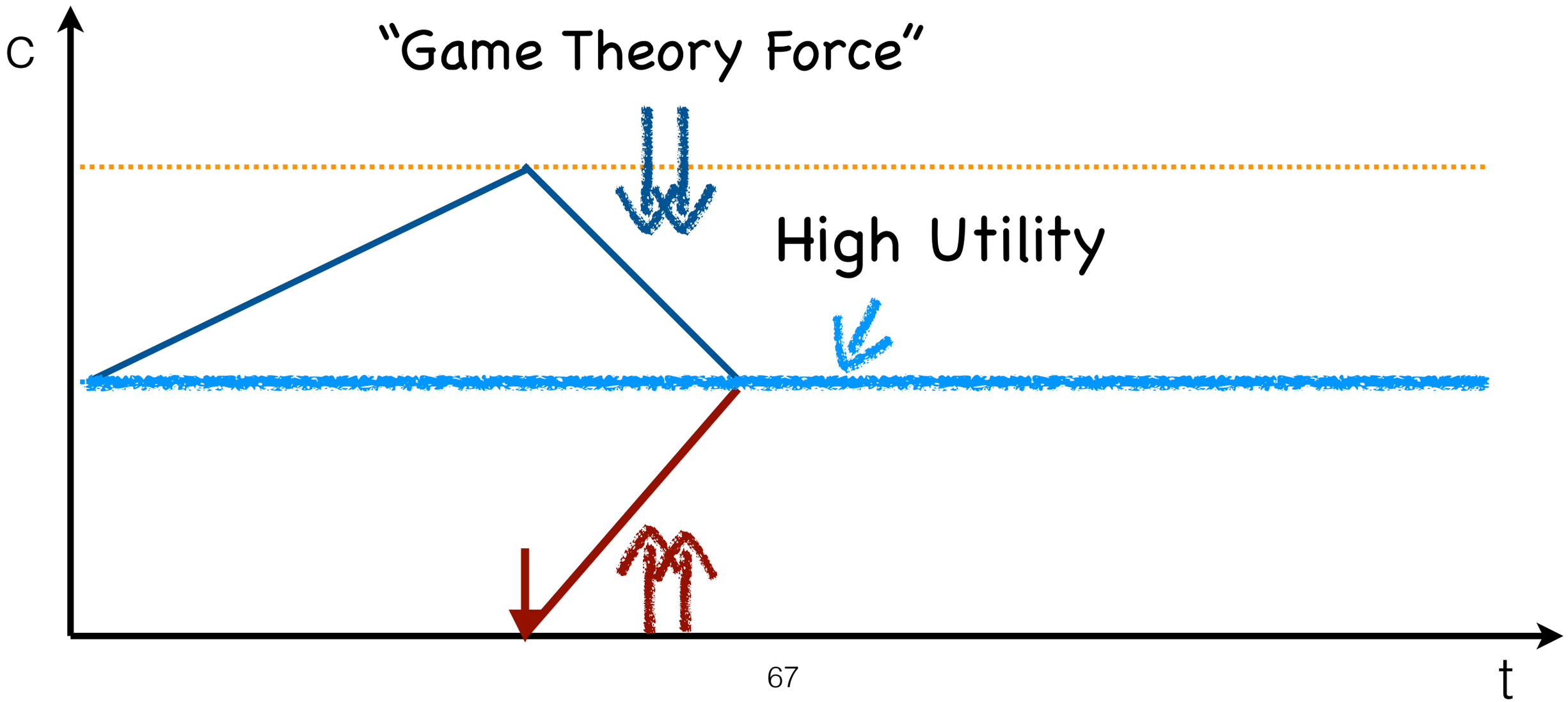


Selfishly maximizing utility
=> non-cooperative game

Do we converge to a fair Nash equilibrium?

PCC Dynamics

PCC does not need AIMD because it looks at real performance



Discussion:

- What's the catch?
- Specialized vs general-purpose
- Different utility function competing in the network?
- Who cares about TCP friendliness and why?

Announcements

Announcements

Wed Feb 15: 50-Gb/s IP Router

Project Proposal: Wed Feb 15 due

Project proposals

Project proposals due 11am Wednesday Feb 15

- Submit via email to Brighten
- 1/2 page, plaintext

Describe:

- the problem you plan to address
- what will be your first steps
- what is the most closely related work, and why it has not addressed your problem
  - at least 3 full academic paper citations (title, authors, publication venue, year) plus paper URLs
- if there are multiple people on your project team, who they are and how you plan to partition the work

Project proposals

Talk to us if…

- You need a project idea
- You'd like advice on a project idea
- You need partners
- You're just a nice person and want to say hi

After submission

- Course staff will give feedback and approve or request changes
- Proposal is 5% of course grade

See also course syllabus