DATA CENTER TO THE HOME

Koen De Schepper, Inton Tsang — Bell Labs

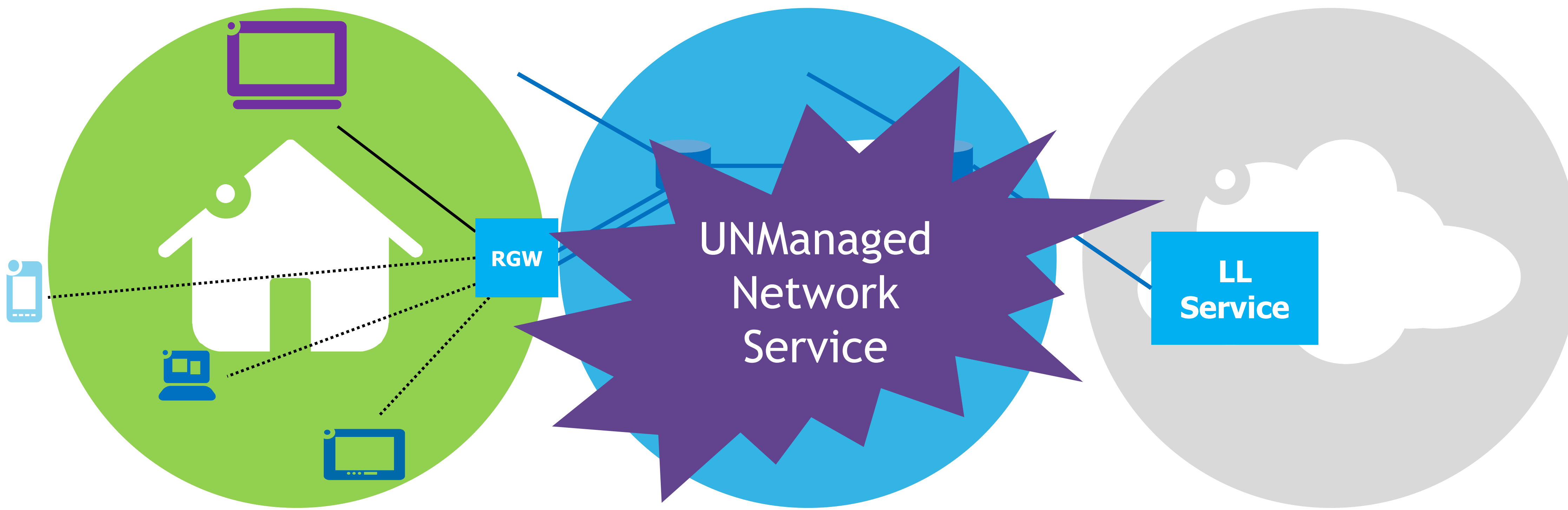
Olga Bondarenko — [simula . research laboratory]

Bob Briscoe — BT

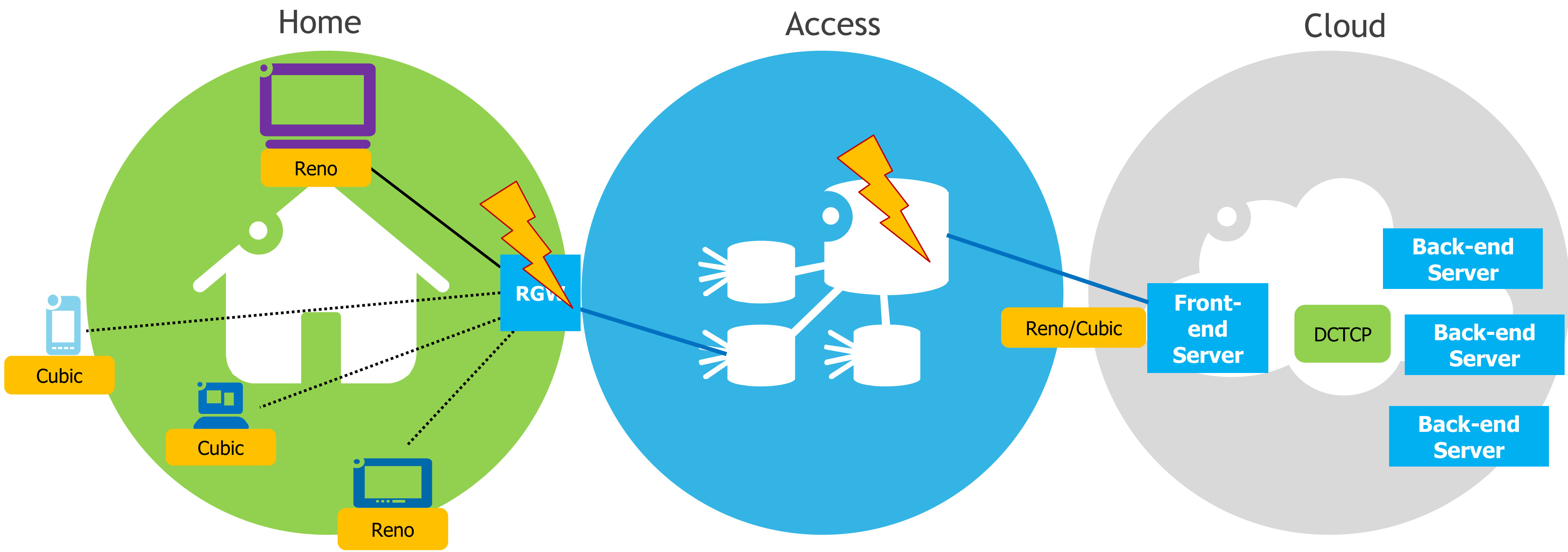
koen.de_schepper@alcatel-lucent.com

March, 2015

DCttH OBJECTIVE: UNIVERSAL SUPPORT FOR LOW LATENCY = SUPPORT FOR ADAPTIVE INTERACTIVE APPLICATIONS



INTERACTIVE APPLICATIONS on the INTERNET ?

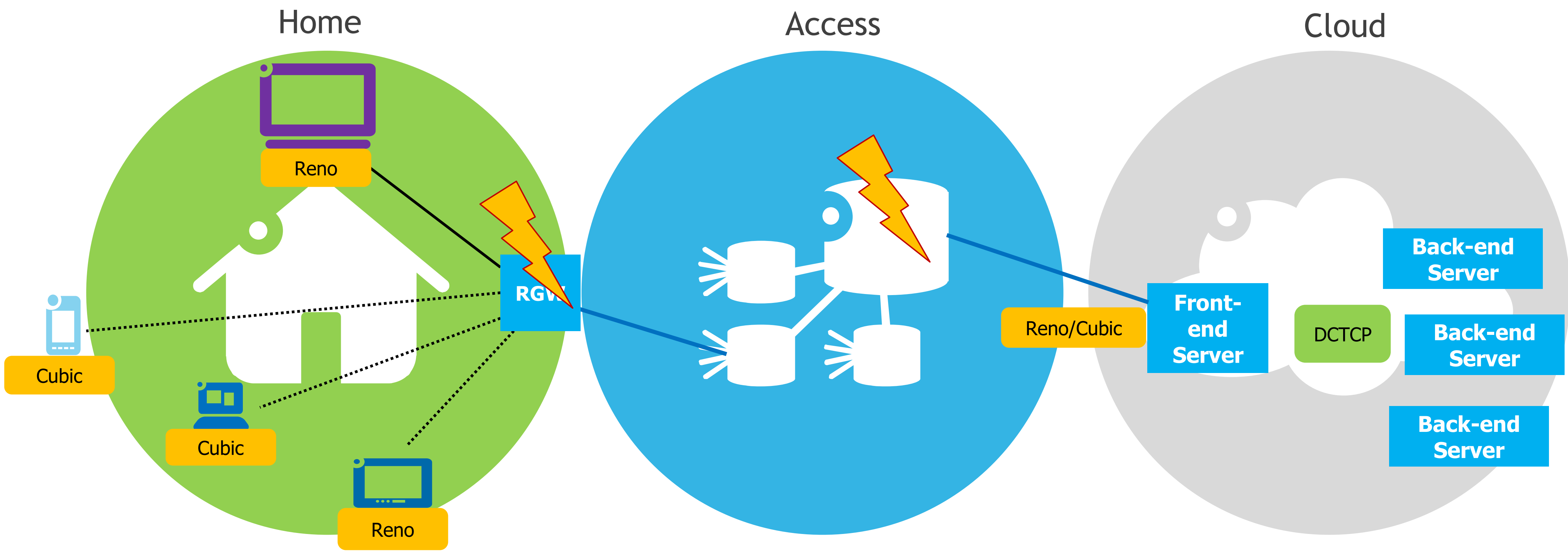


Large queues for high throughput and low drop
= Poor Latency
= Bad for interactive applications

ECN = No drop
ECN++ = Small queues
= Low latency & High throughput

RITE
REDUCING INTERNET TRANSPORT LATENCY

DATACENTER to the HOME ?



Windows and Linux 3.18
have DCTCP implementations ready

Public Internet
does not support DCTCP

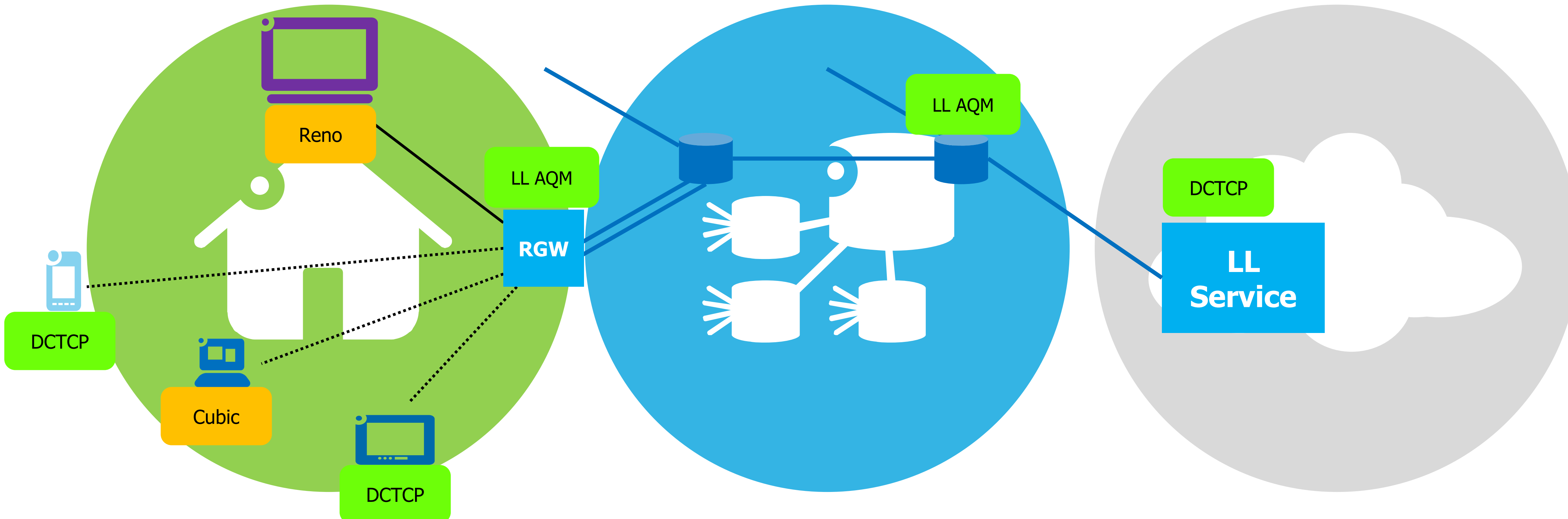
DCTCP available on
Windows Server and Linux 3.18
used internally in the data center

Clients use Reno and Cubic
Can’t use DCTCP without causing trouble

RITE
Reducing Internet Transport Latency

MIGRATION OBJECTIVE: LOW LATENCY ACCESS TO THE CLOUD, EQUAL STEADY STATE THROUGHPUT TO RENO/CUBIC

Can DCTCP be used as Low Latency congestion controller ?



Which AQM to be deployed on queuing bottlenecks ?

Support migration !

LOWER LATENCY BY SMARTER USE OF ECN
DATA CENTER TCP

TCP (Reno)	←→	DCTCP

Response to congestion in sender

TCP (Reno):
- Half the congestion window when drop detected in one RTT

DCTCP:
- Reduce partially per marked packet; half if all marked in one RTT

  ➔ React according to level of congestion

ECN feedback in receiver

TCP (Reno):
- Echo Congestion Experienced (CE) until sender acknowledges Congestion Window Reduced (CWR)

DCTCP:
- Echo marking state of received packets without acknowledgement ➔ accurate ECN feedback

ECN marking in network

TCP (Reno):
- Smooth and delay a drop or mark to allow bursts

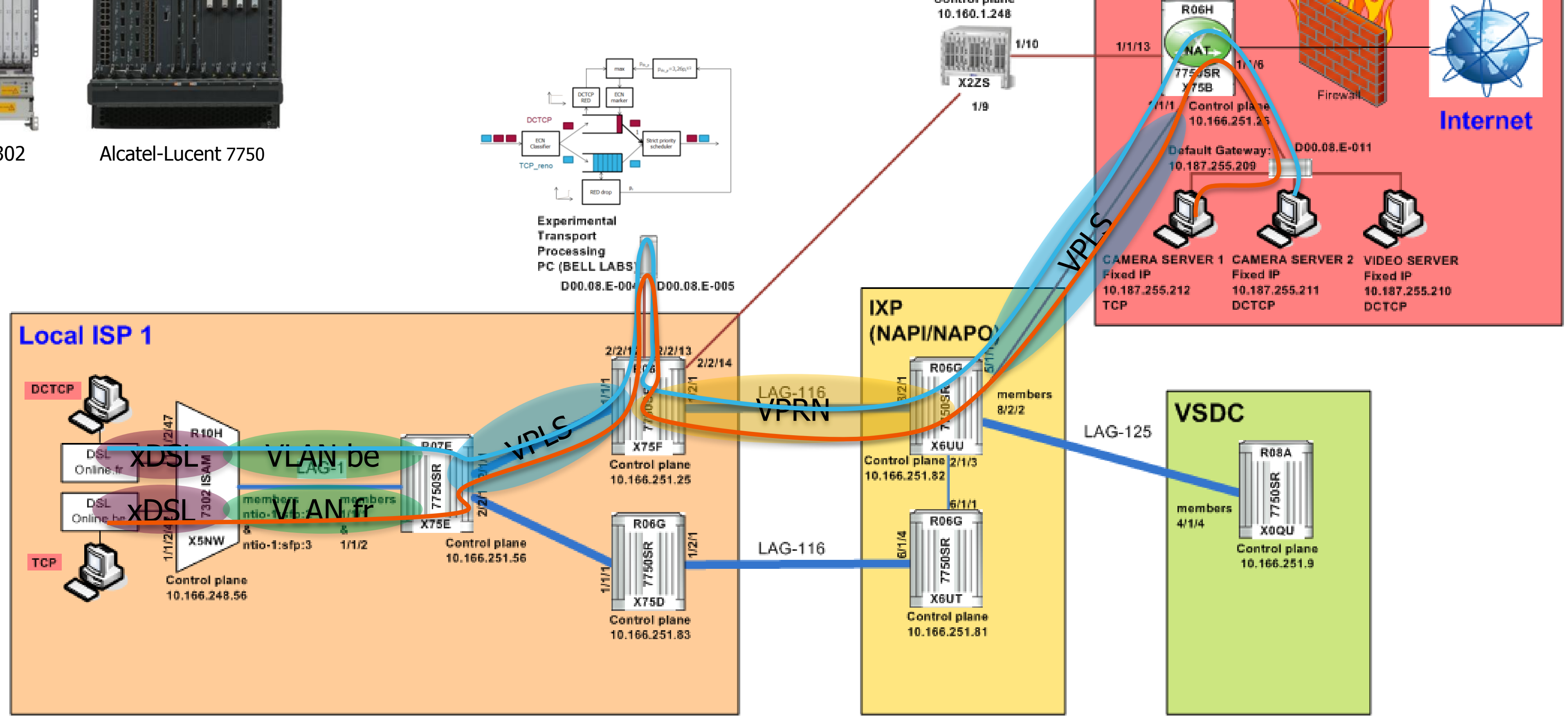
DCTCP:
- Don’t smooth or delay queue size
- Shallower marking threshold

  ➔ immediate ECN marking

DEMONSTRATED ON A REAL BB RESIDENTIAL TESTBED

Alcatel-Lucent 7302

Alcatel-Lucent 7750



RTT =

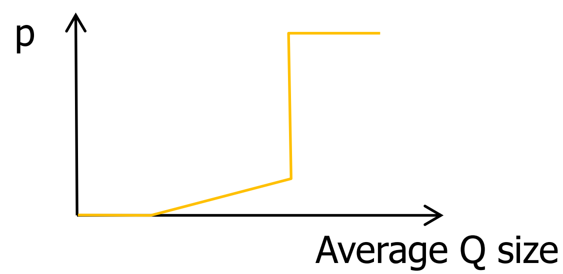
8ms

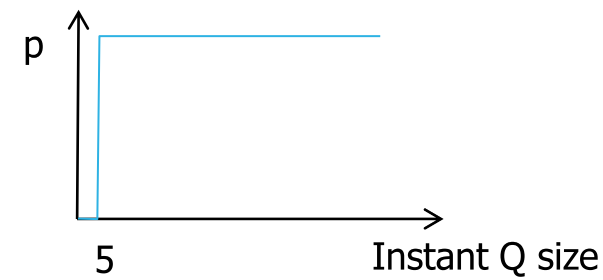
40ms

LOWER LATENCY BY SMARTER USE OF ECN
DATA CENTER TCP

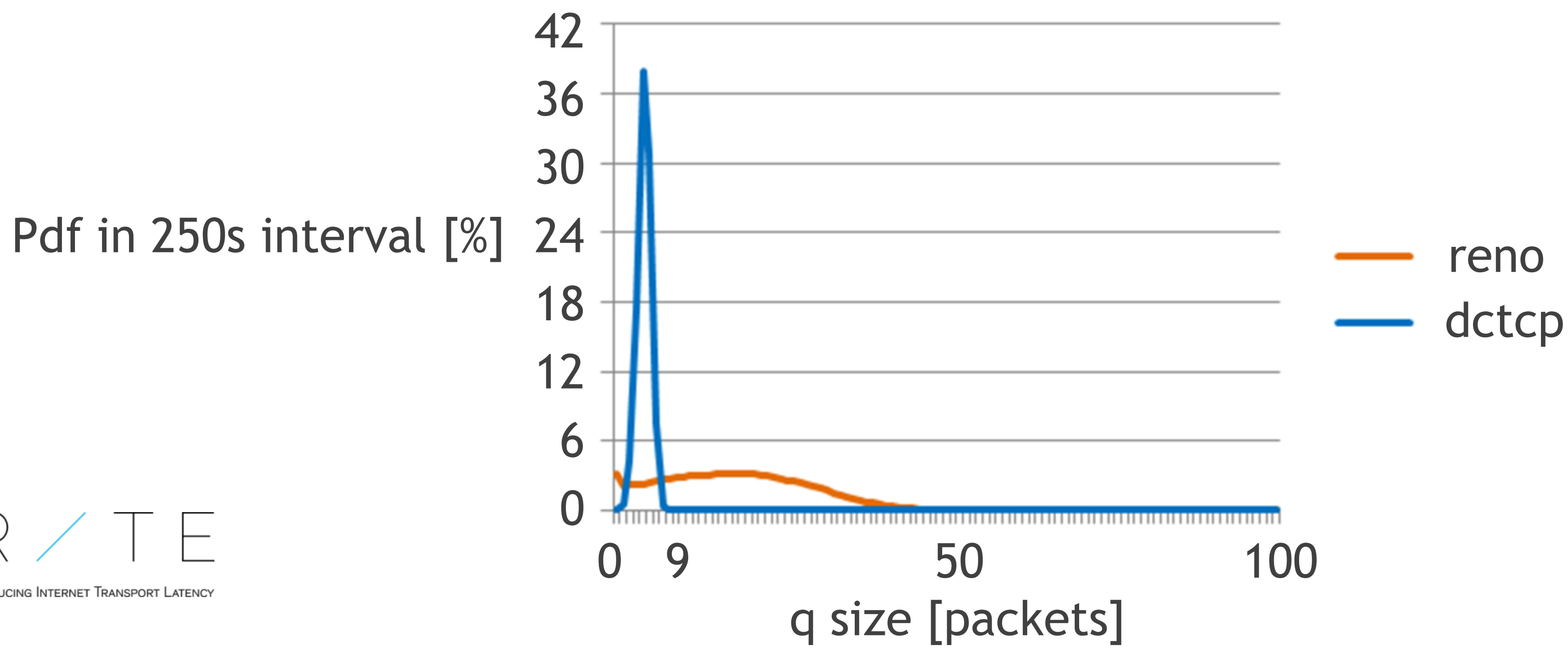
TCP (Reno) ↔ **DCTCP**

AQM configuration





Q size variation



Measured in a BB DSL testbed

RTT = 8 ms (unloaded)

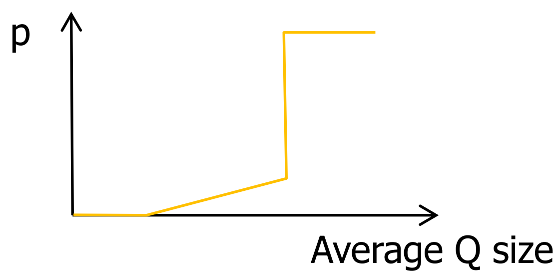
BW = 40 Mbps (downstream)

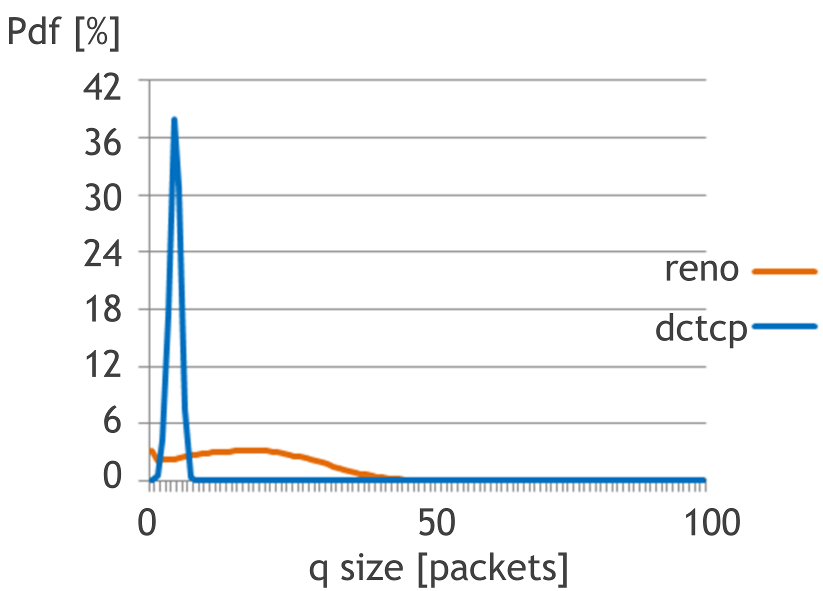
1 steady state flow running alone

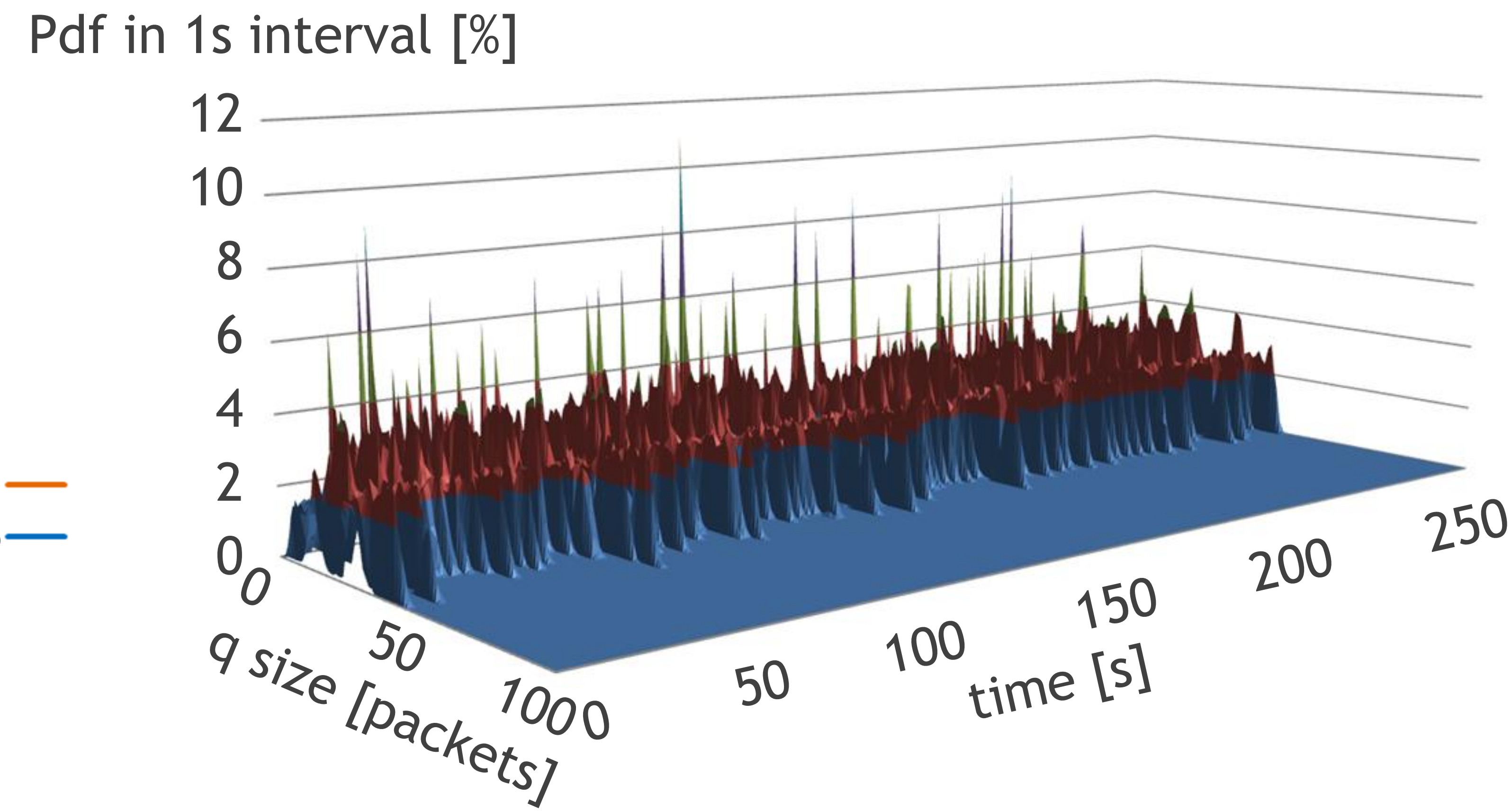
Reno/Cubic/DCTCP = Linux kernel 3.18

R⁄TE
Reducing Internet Transport Latency

QUEUE SIZE AT DEQUEUE
1 TCP RENO FLOW (STEADY STATE)

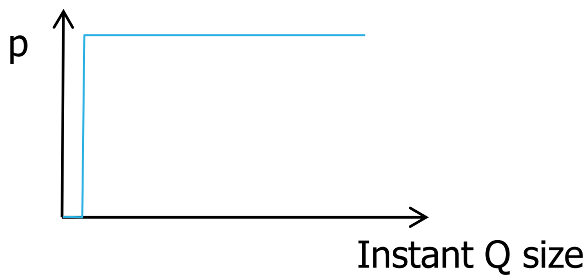


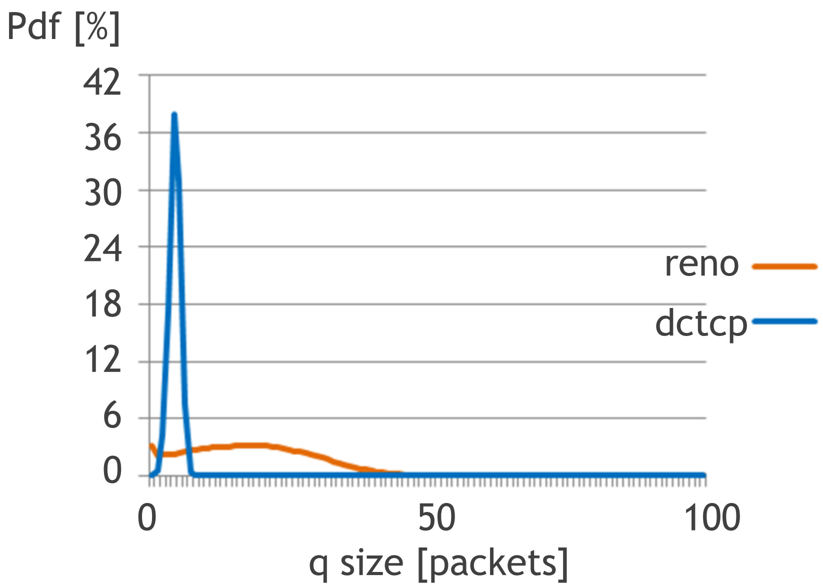


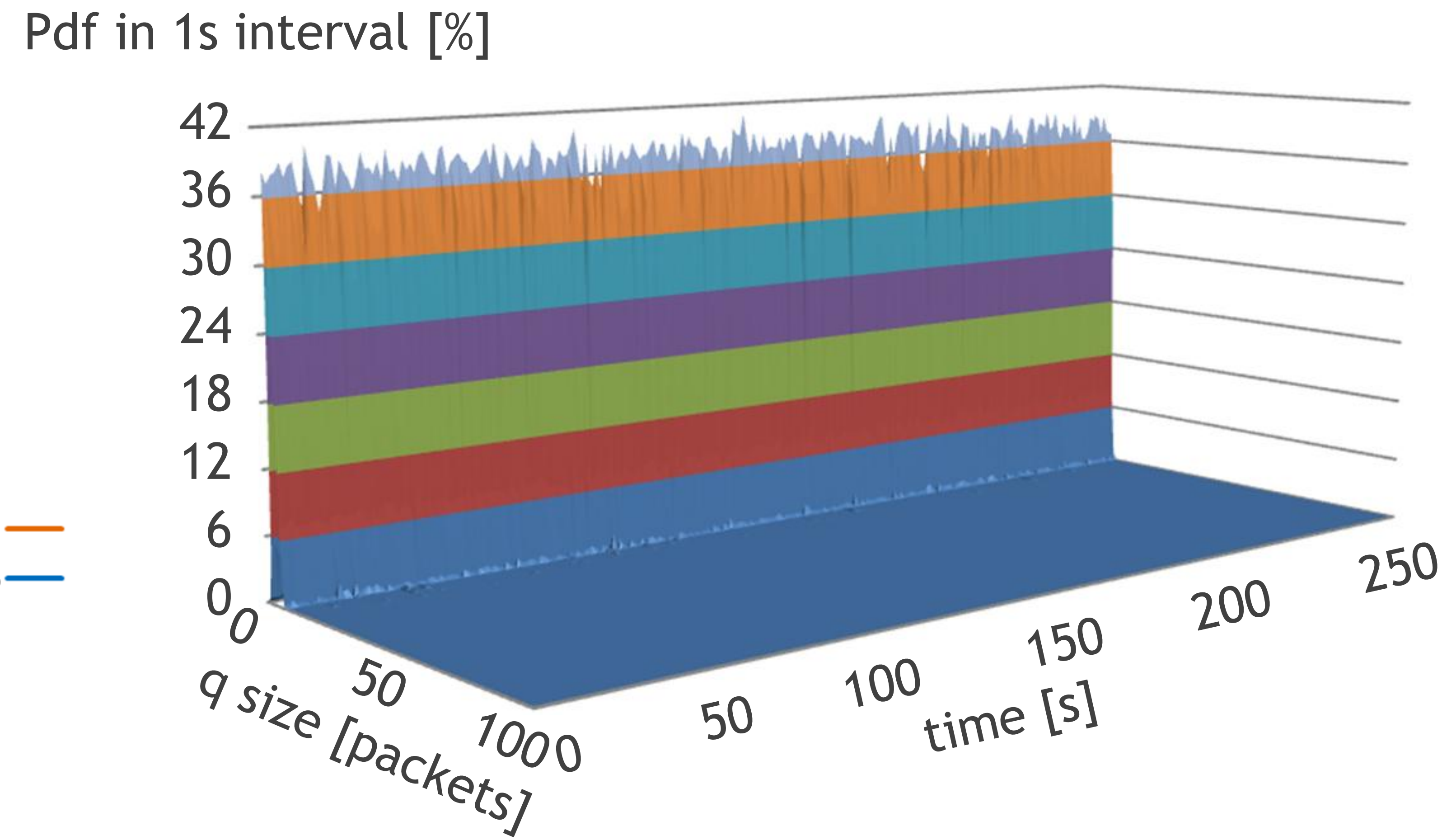


REDUCING INTERNET TRANSPORT LATENCY

QUEUE SIZE AT DEQUEUE
1 DCTCP FLOW (STEADY STATE)

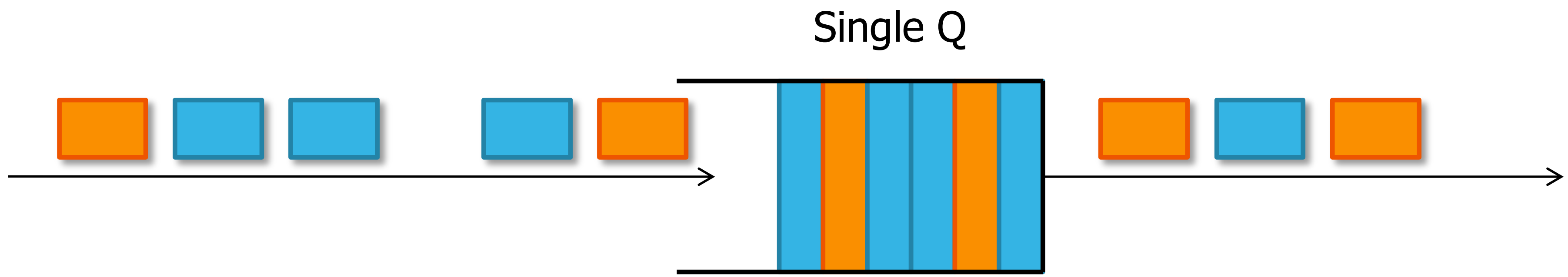




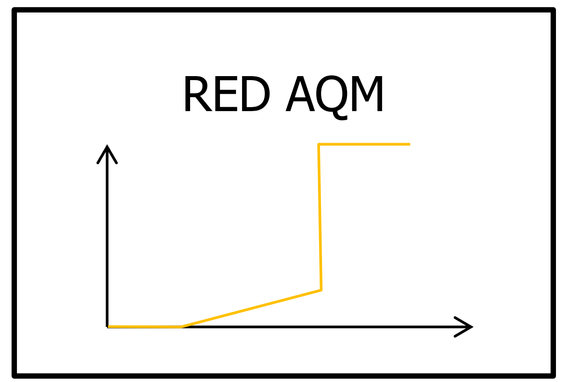


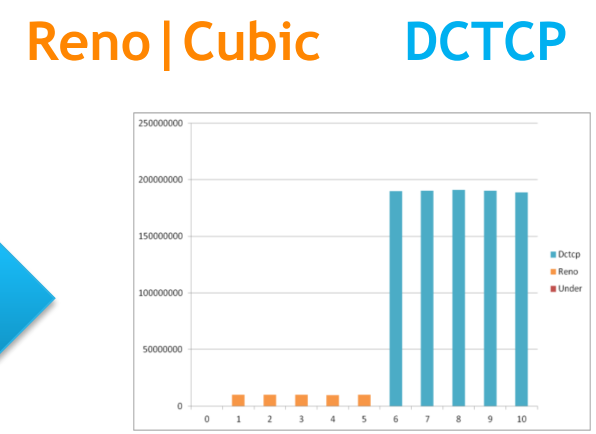
R/TE
Reducing Internet Transport Latency

DCTCP DOES NOT WORK ON TRADITIONAL RED-ECN

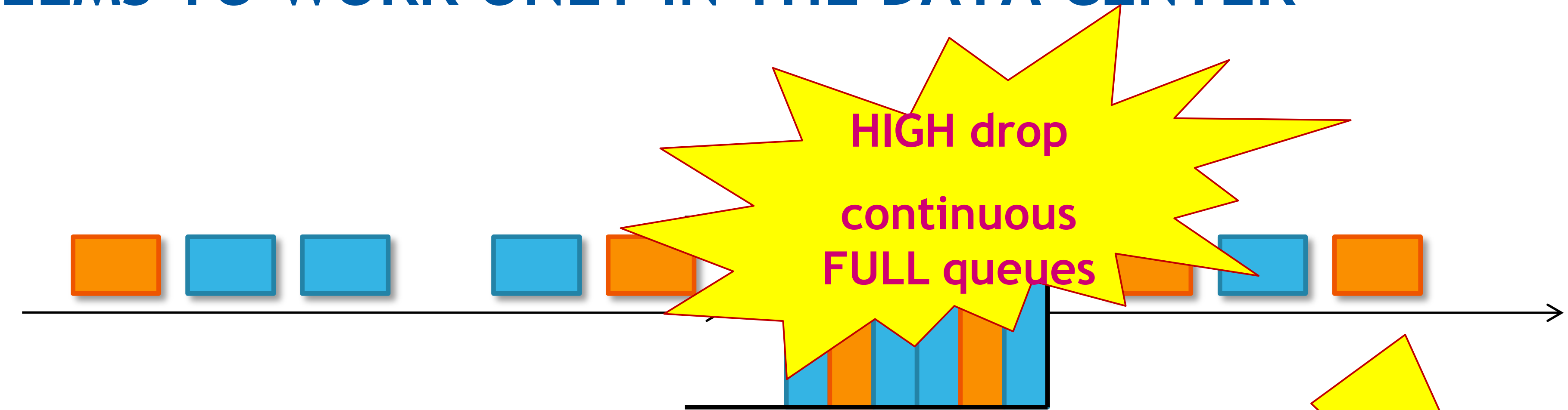


$$p_{dc} = p_{reno}$$

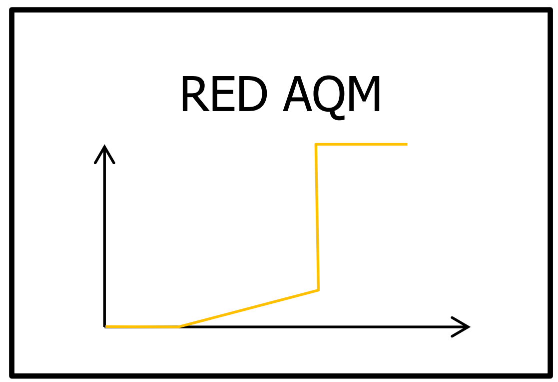


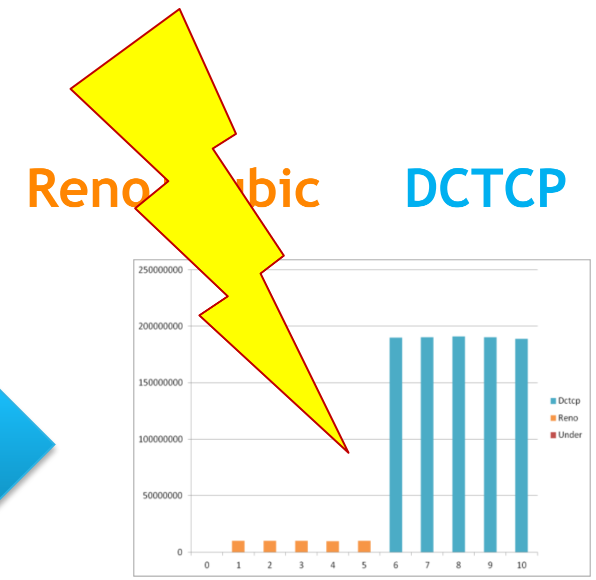


DCTCP SEEMS TO WORK ONLY IN THE DATA CENTER



$$p_{dc} = p_{reno}$$





REDUCING INTERNET TRANSPORT LATENCY

THROUGHPUT:

RTT = 8 ms (unloaded)

BW = 40 Mbps (downstream)

BDP = 27 full sized packets

AQM = RED with recommended configuration*

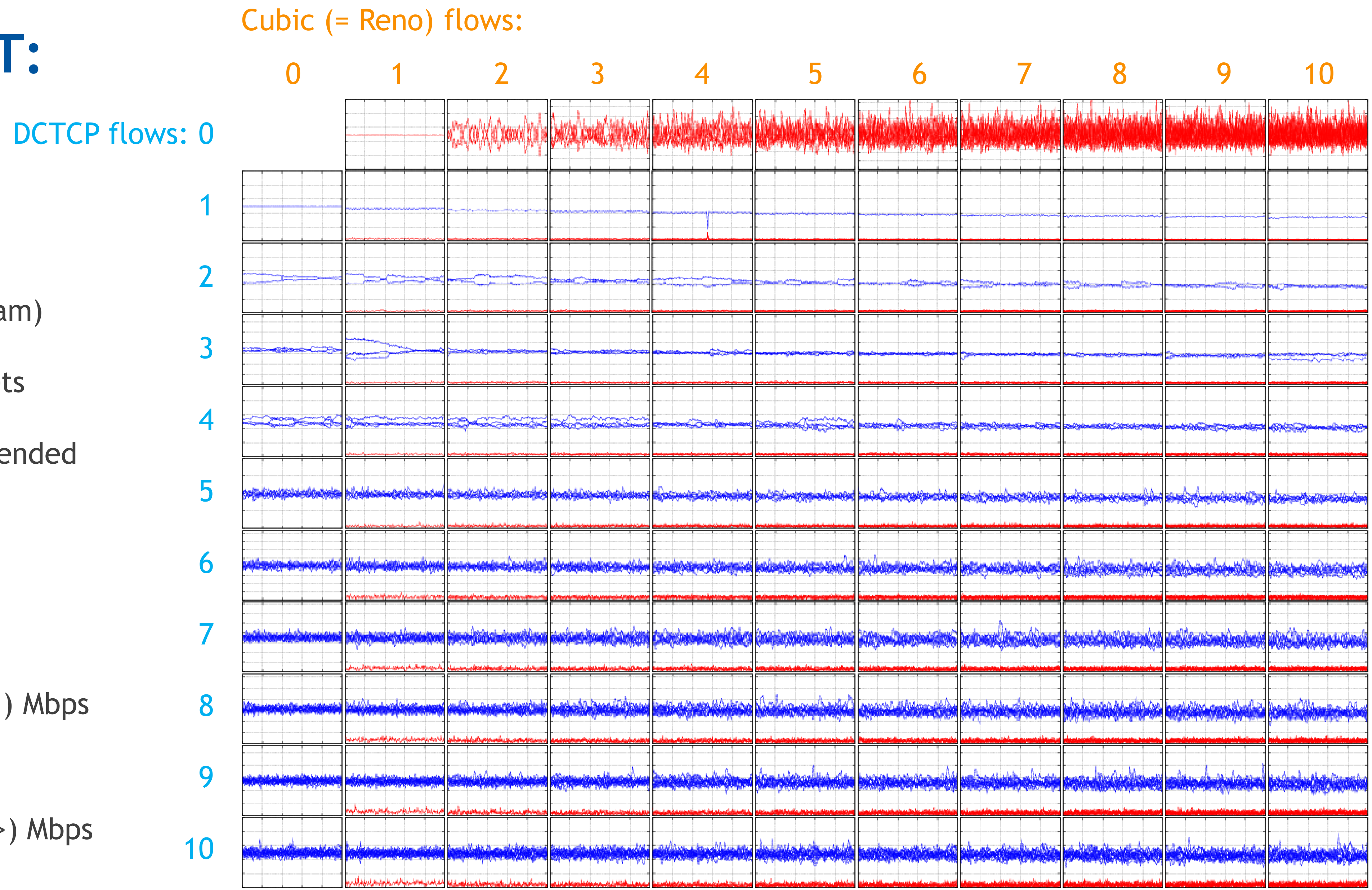
X-axis: 0 – 250 sec

Y-axis: first row:

0 – (80 / <nbr_flows>) Mbps

Y-axis: other rows

0 – (80 / <nbr_dctcp>) Mbps



R / T E
Reducing Internet Transport Latency

* tc qdisc add dev eth2 root red limit 1600000 min 120000 max 360000 avpkt 1000 burst 220 ecn bandwidth 40Mbit

Q SIZE PDF:

RTT = 8 ms (unloaded)

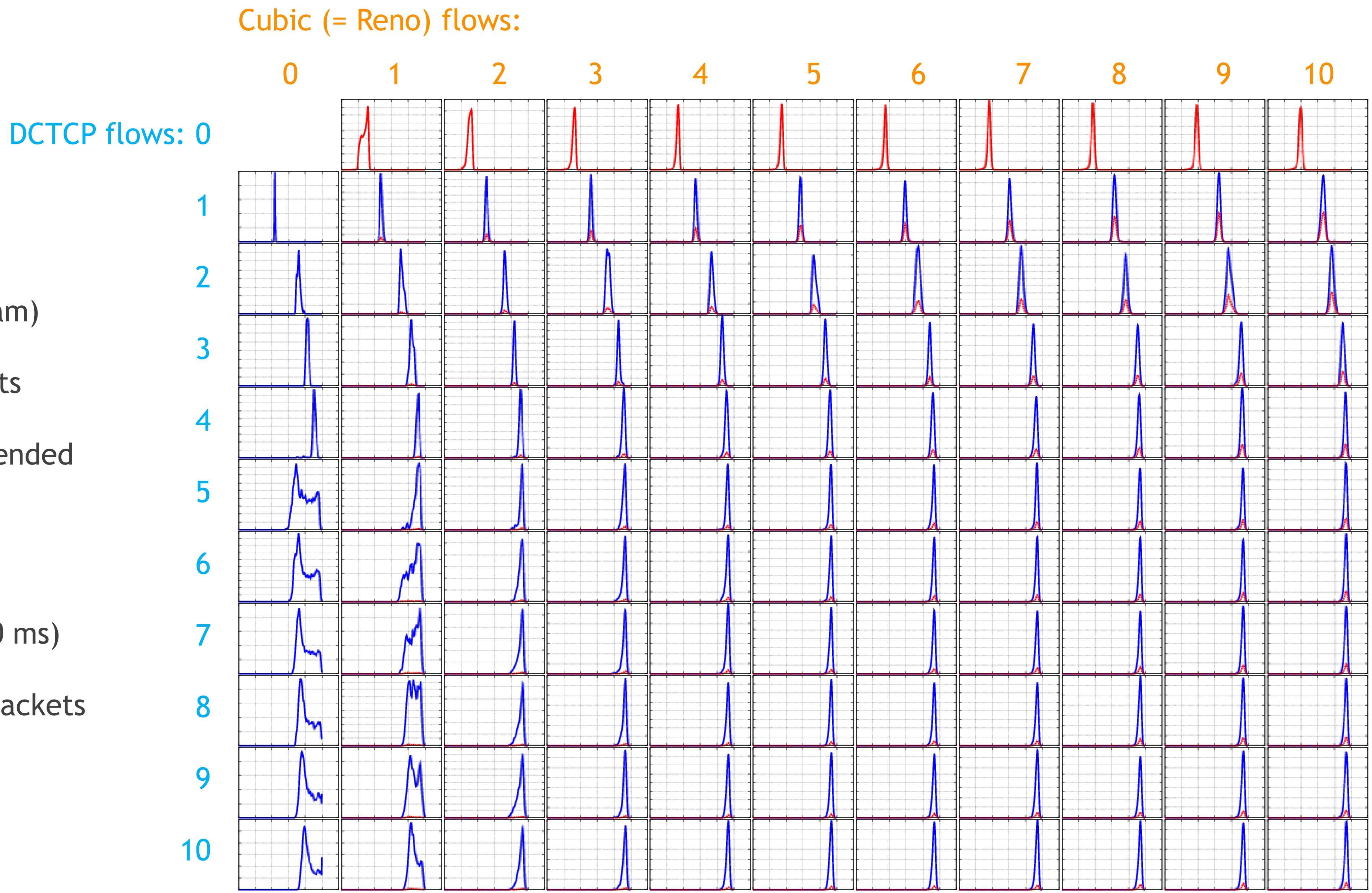
BW = 40 Mbps (downstream)

BDP = 27 full sized packets

AQM = RED with recommended configuration*

X-axis: 0 – 300 packets (450 Kbytes, 90 ms)

Y-axis: autoscale count packets

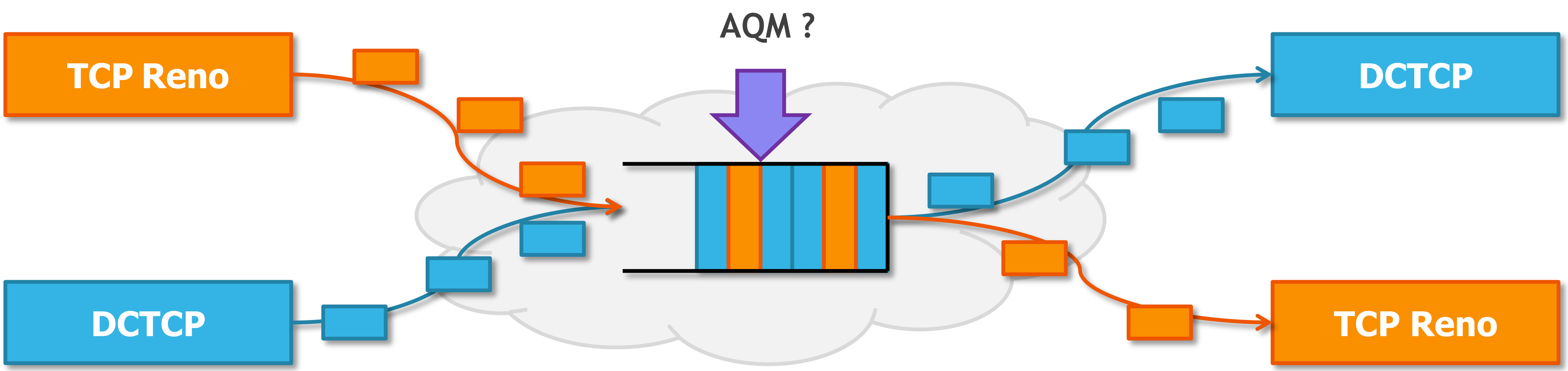


* tc qdisc add dev eth2 root red limit 1600000 min 120000 max 360000 avpkt 1000 burst 220 ecn bandwidth 40Mbit

RITE
Reducing Internet Transport Latency

AQMS FOR EQUAL STEADY STATE RATE
MIGRATION PATH FOR NEW CC SCHEMES

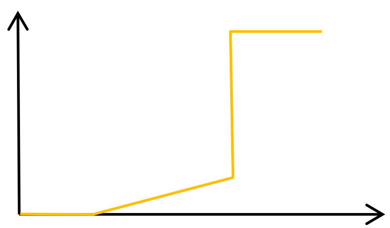
- How should an AQM guarantee an equal steady state rate for flows with different congestion control schemes
  - classify packets according to CC schemes
  - align the drop/mark probabilities



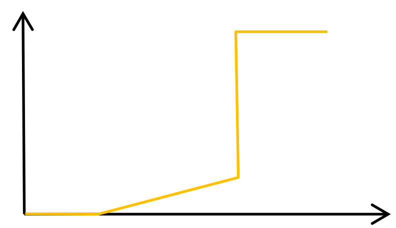
TCP CONGESTION CONTROL SCHEMES
STEADY STATE RATE

- Steady state rate has been calculated for existing CC schemes:

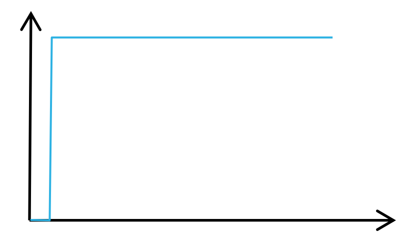
$$r_{reno} = \frac{1.22}{p^{1/2}\mathrm{RTT}}$$



$$r_{cubic} = \frac{1.17}{p^{3/4}\mathrm{RTT}^{1/4}}$$



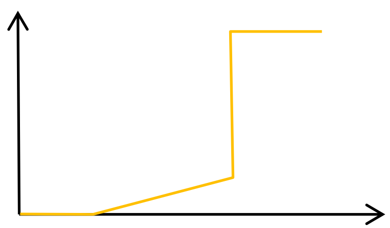
$$r_{dc} = \frac{2}{p^{2} \cdot \mathrm{RTT}}$$



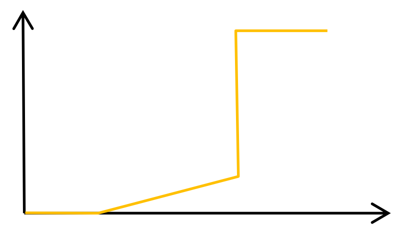
TCP CONGESTION CONTROL SCHEMES
STEADY STATE RATE

- Steady state rate has been calculated for existing CC schemes:

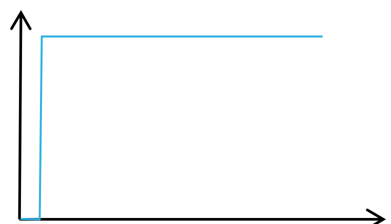
$$r_{reno} = \frac{1.22}{p^{1/2}\mathrm{RTT}}$$



$$r_{cubic} = \frac{1.17}{p^{3/4}\mathrm{RTT}^{1/4}}$$

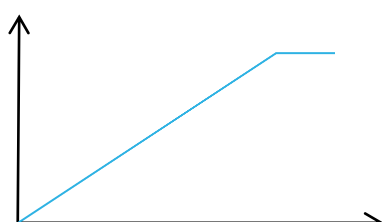


$$r_{dc} = \frac{2}{p^{2} \cdot \mathrm{RTT}}$$



- But we calculated that DCTCP running in non-on/off mode behaves as:

$$r_{dc\_p} = \frac{2}{p \cdot \mathrm{RTT}}$$



TCP CONGESTION CONTROL SCHEMES
FAIRNESS BETWEEN DCTCP AND RENO

- Mark/drop probability relation for equal rate and RTT:

$$r_{reno} = r_{dc}$$
$$\mathrm{RTT}_{reno} = \mathrm{RTT}_{dc}$$

$$\frac{1.22}{p_{reno}^{1/2}\mathrm{RTT}_{reno}} = \frac{2}{p_{dc} \cdot \mathrm{RTT}_{dc}}$$

$$p_{reno} = \left(\frac{p_{dc}}{1.63}\right)^2$$

TCP CONGESTION CONTROL SCHEMES
FAIRNESS BETWEEN DCTCP AND RENO

- Mark/drop probability relation for equal rate and RTT:

$$r_{reno} = r_{dc}$$
$$\mathrm{RTT}_{reno} = \mathrm{RTT}_{dc}$$

$$\frac{1.22}{p_{reno}^{1/2}\mathrm{RTT}_{reno}} = \frac{2}{p_{dc} \cdot \mathrm{RTT}_{dc}}$$

$$p_{reno} = \left(\frac{p_{dc}}{1.63}\right)^2$$

Square is easy!

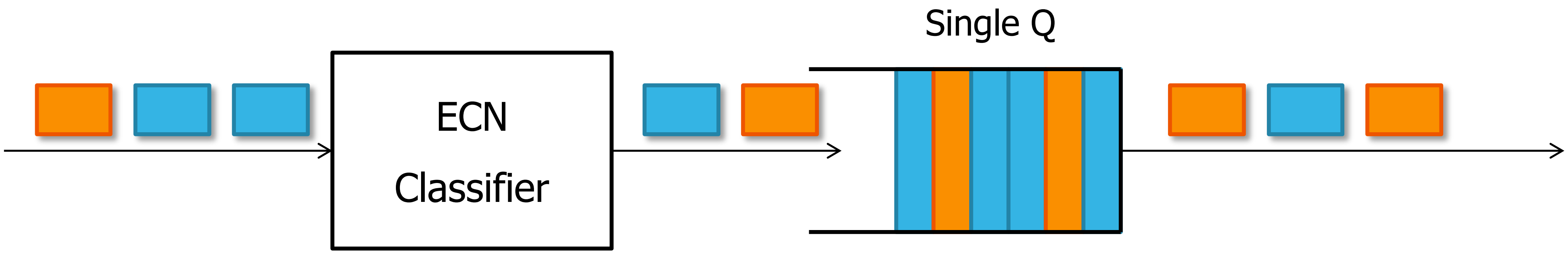
Compare Q size with 2 random variables

$$P = f(Q) \qquad p \Rightarrow \mathrm{Random}() < P$$

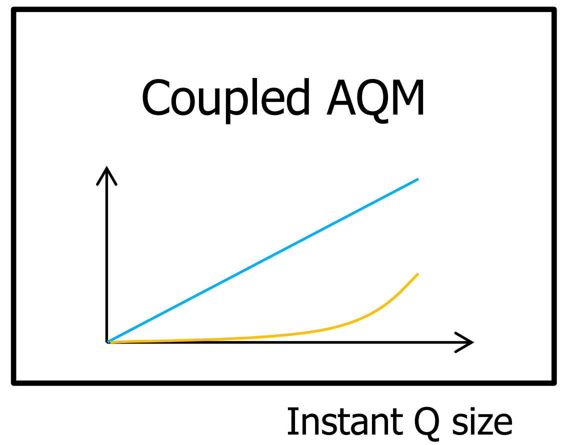
$$p^2 \Rightarrow (\mathrm{Random}() < P) \,\&\&\, (\mathrm{Random}() < P)$$

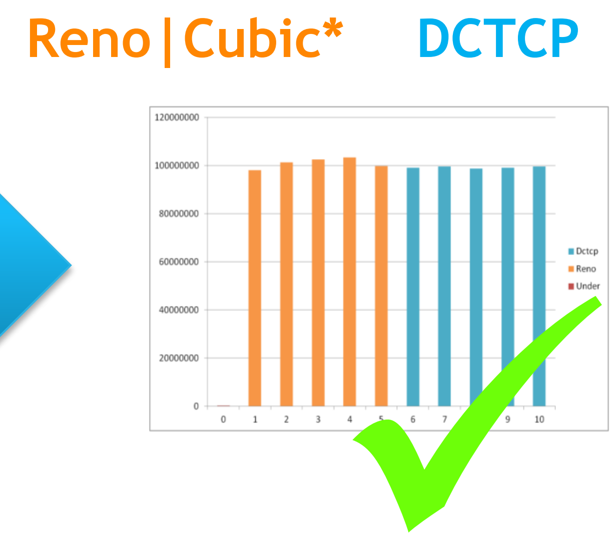
$$p^2 \Rightarrow \max(\mathrm{Random}(), \mathrm{Random}()) < P$$

DCTCP BEHAVES EXACTLY AS RENO IF WE CORRECTLY CORRELATE MARKING AND DROPPING



$$p_{reno} = \left(\frac{p_{dc}}{1.63}\right)^2$$

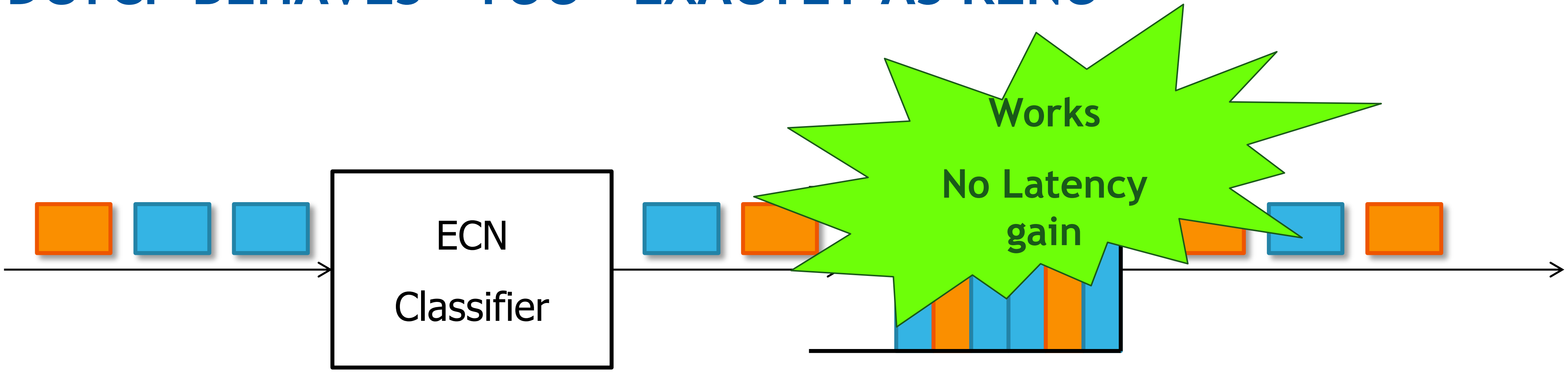




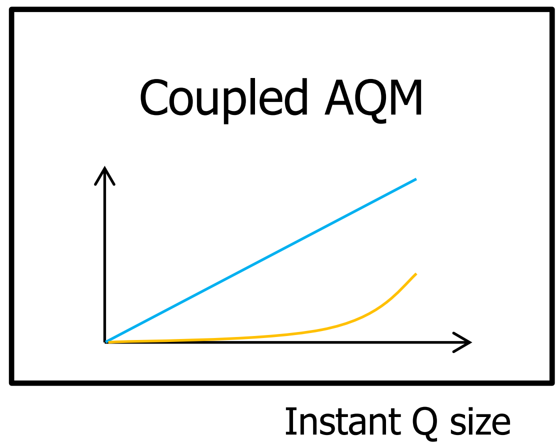
* Under local DC-access conditions (small BDP) Cubic behaves as Reno

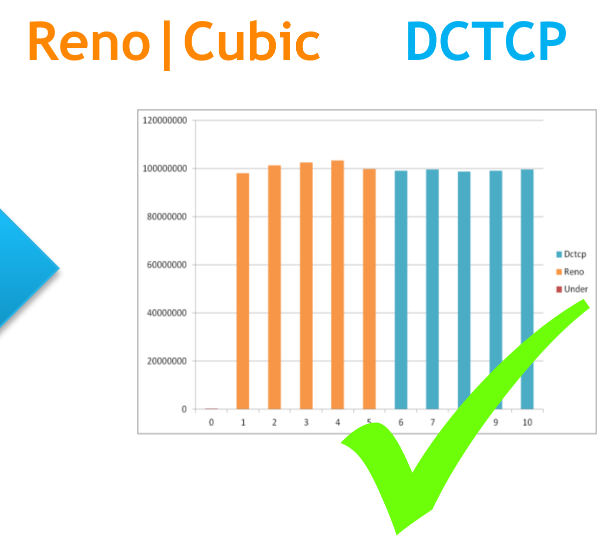
Slope starts from the origin to avoid ON/OFF behavior in steady state

DCTCP BEHAVES "TOO" EXACTLY AS RENO

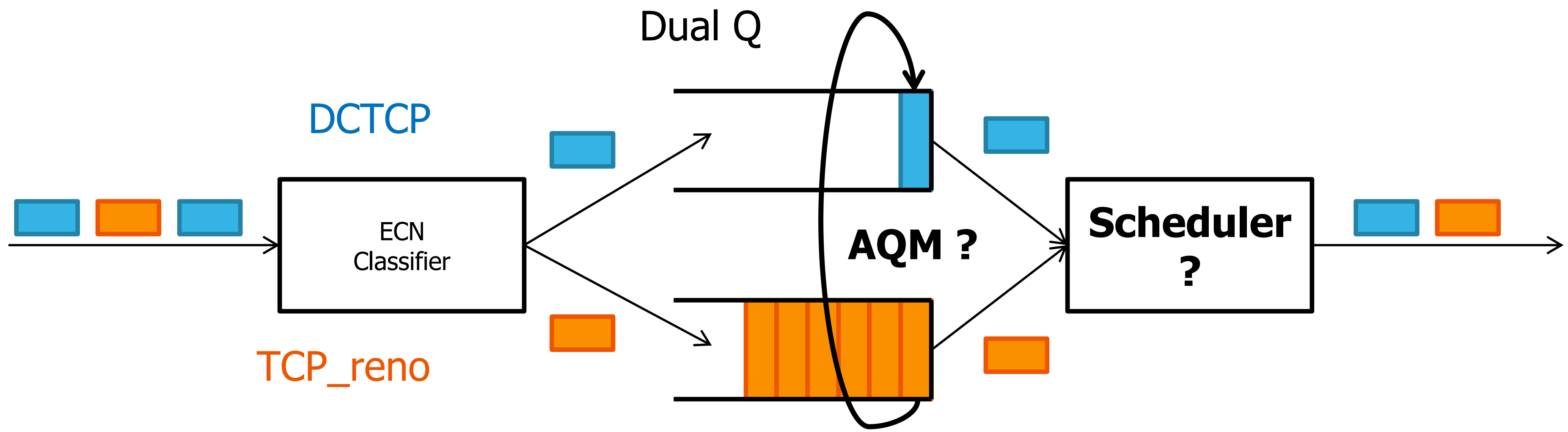


$$p_{reno} = \left(\frac{p_{dc}}{1.63}\right)^2$$



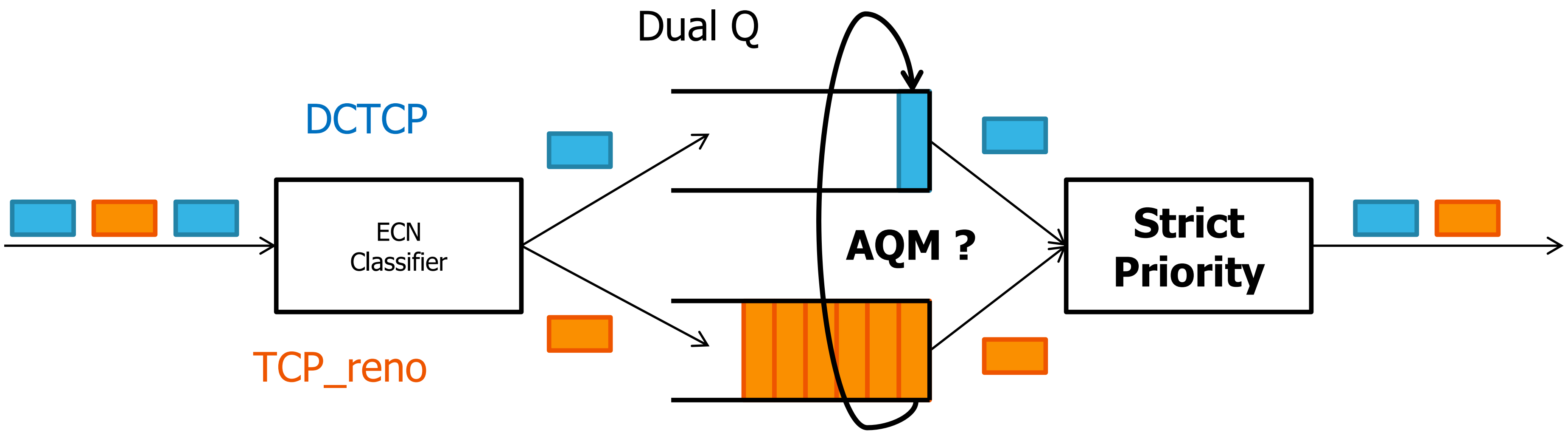


DUAL QUEUE - LOW LATENCY



$$\frac{r_{reno}}{r_{dc}} = \frac{1.22}{2} \frac{p_{dc}}{\sqrt{p_{reno}}} \frac{\mathrm{RTT}_{dc}}{\mathrm{RTT}_{reno}} \quad \Leftarrow \quad p_{reno} = \left(\frac{p_{dc}}{8} \right)^2$$

DUAL QUEUE - LOW LATENCY



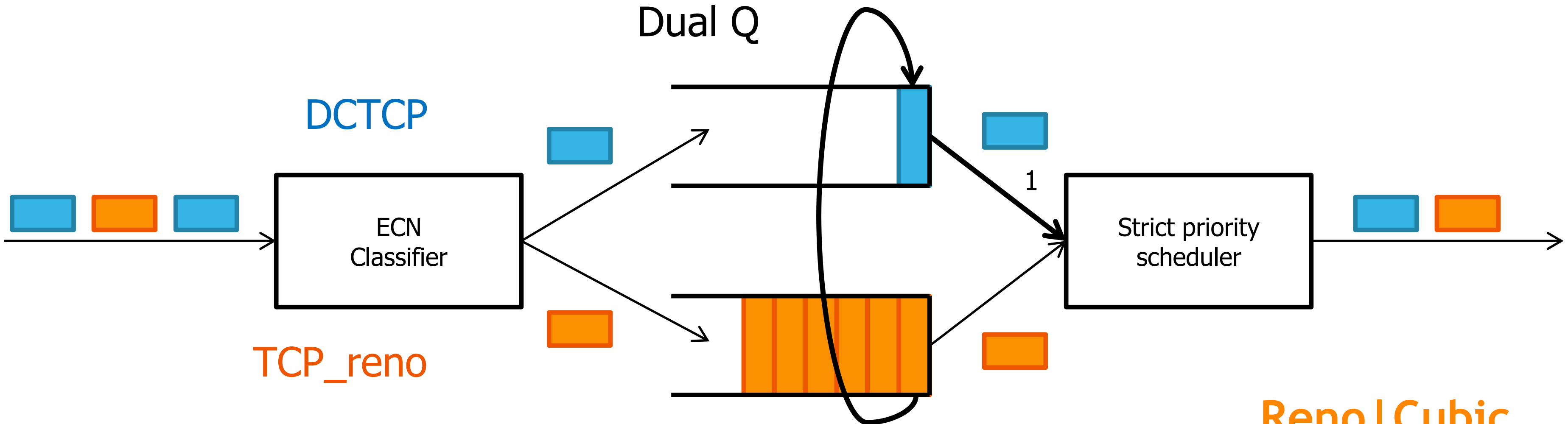
1/5 = 8 ms /(8 + 32) ms

$$\frac{r_{reno}}{r_{dc}} = \frac{1.22}{2} \frac{p_{dc}}{\sqrt{p_{reno}}} \frac{RTT_{dc}}{RTT_{reno}}$$

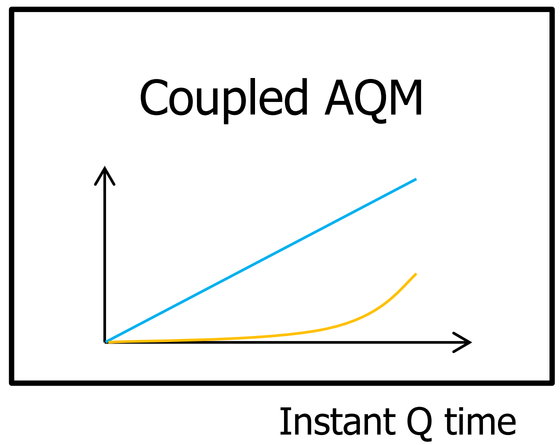
= 5

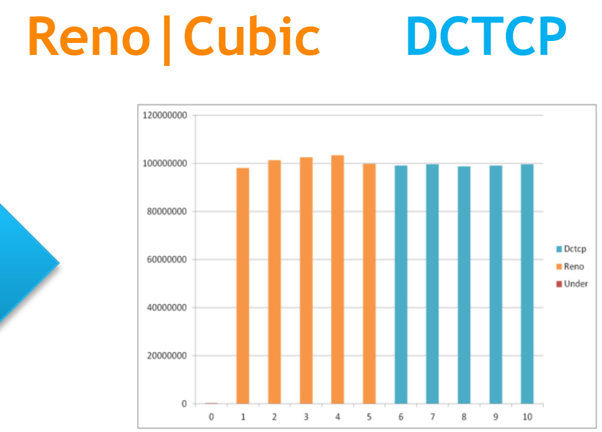
$$p_{reno} = \left(\frac{p_{dc}}{8} \right)^2$$

DUAL QUEUE - LOW LATENCY

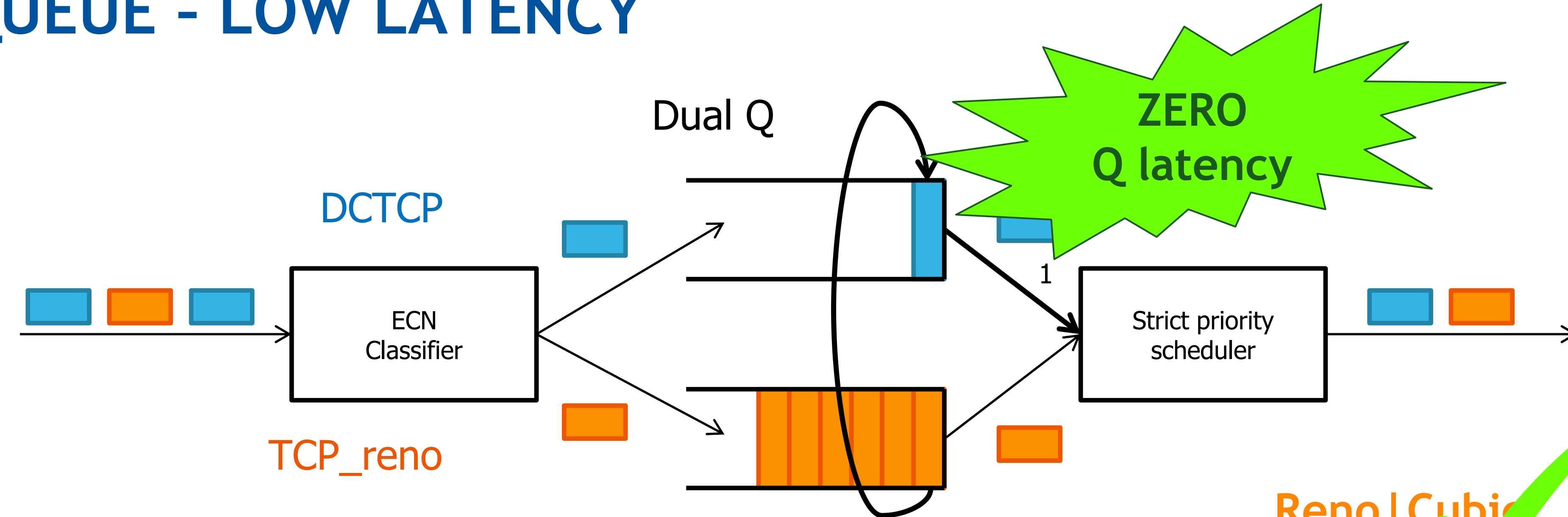


$$p_{reno} = \left(\frac{p_{dc}}{8}\right)^2$$

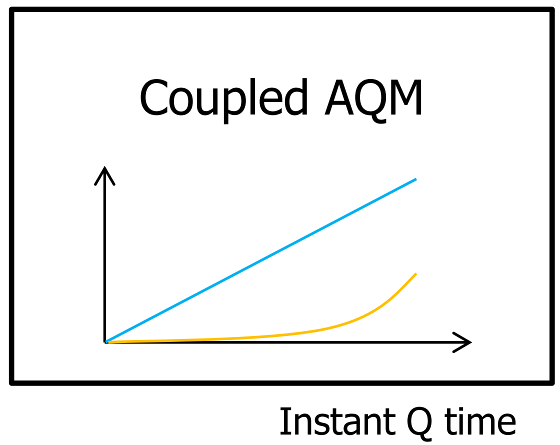


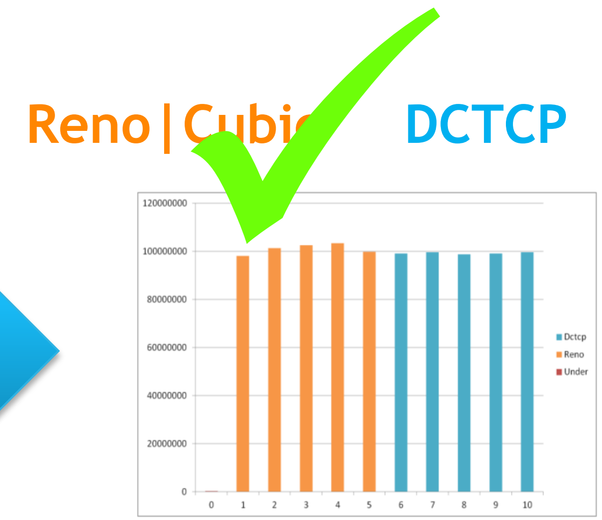


DUAL QUEUE - LOW LATENCY



$$p_{reno} = \left(\frac{p_{dc}}{8}\right)^2$$





Measure Q in time is important for optimal fairness !

R / T E
Reducing Internet Transport Latency

THROUGHPUT:

RTT = 8 ms (unloaded)

BW = 40 Mbps (downstream)

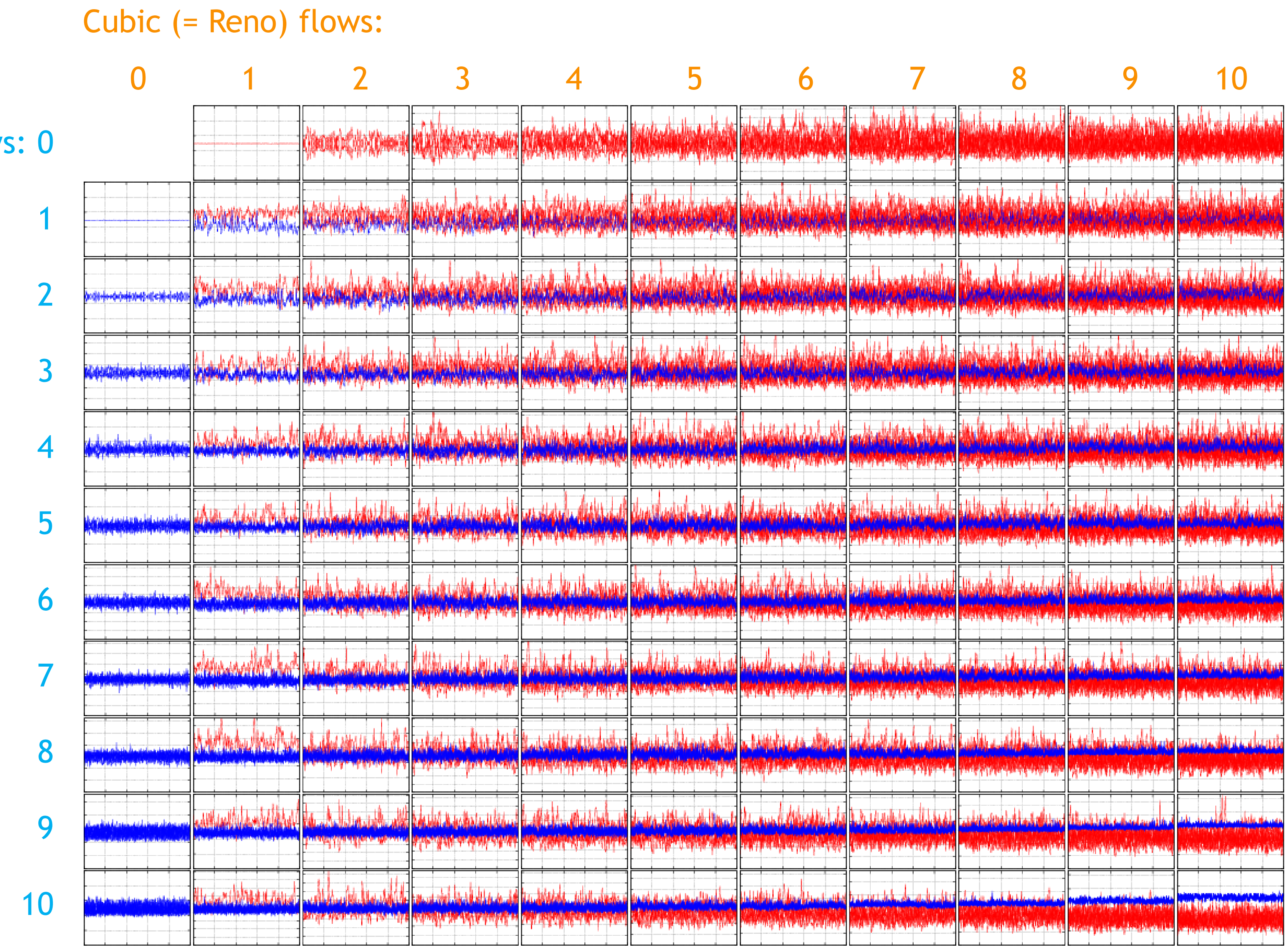
BDP = 27 full sized packets

AQM = DualQ Coupled

X-axis: 0 – 250 sec

Y-axis: all rows:

0 – (80 / <nbr_flows>) Mbps



REDUCING INTERNET TRANSPORT LATENCY

Q SIZE PDF:

RTT = 8 ms (unloaded)

BW = 40 Mbps (downstream)

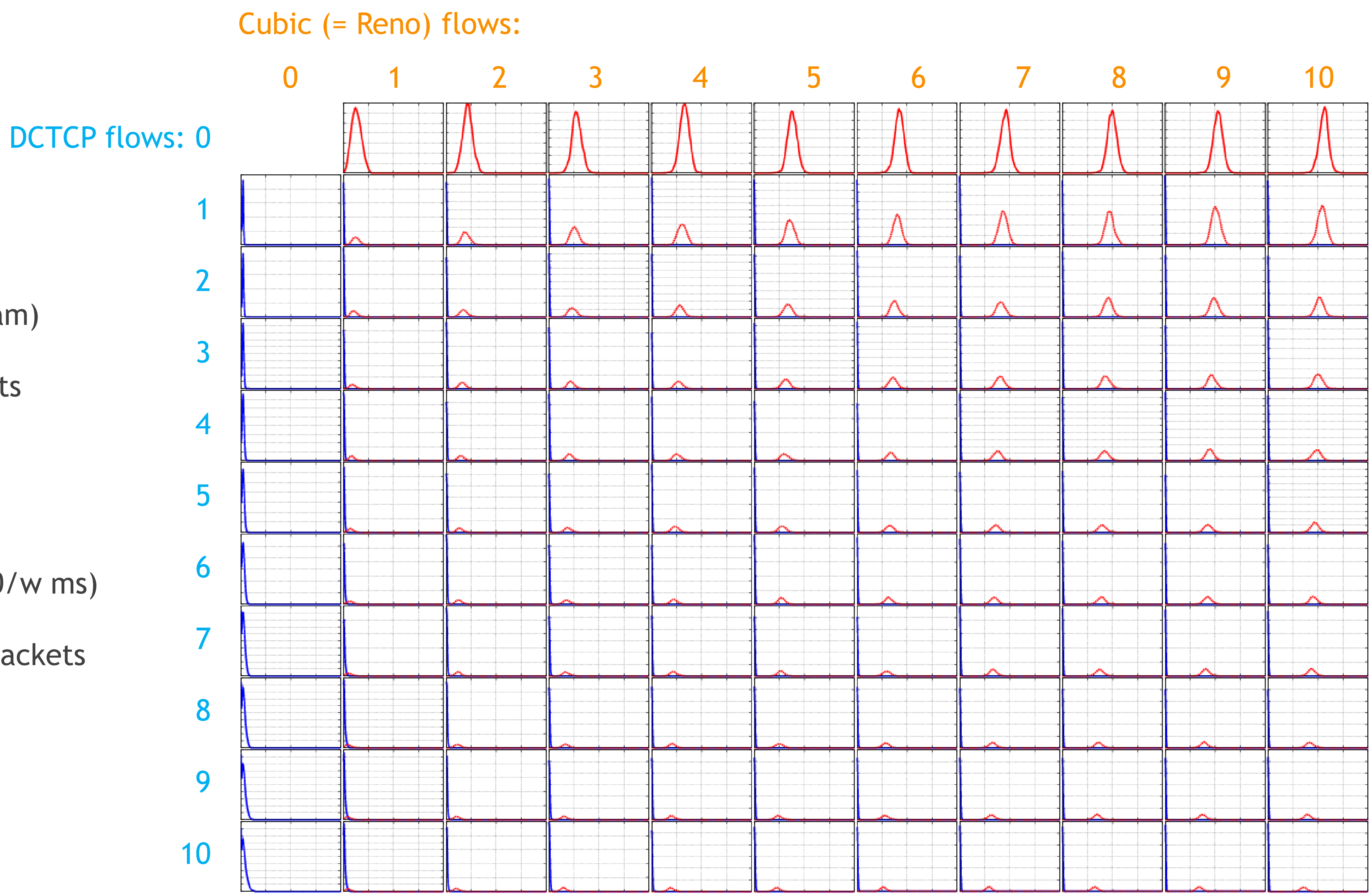
BDP = 27 full sized packets

AQM = DualQ Coupled

X-axis: 0 – 300 packets

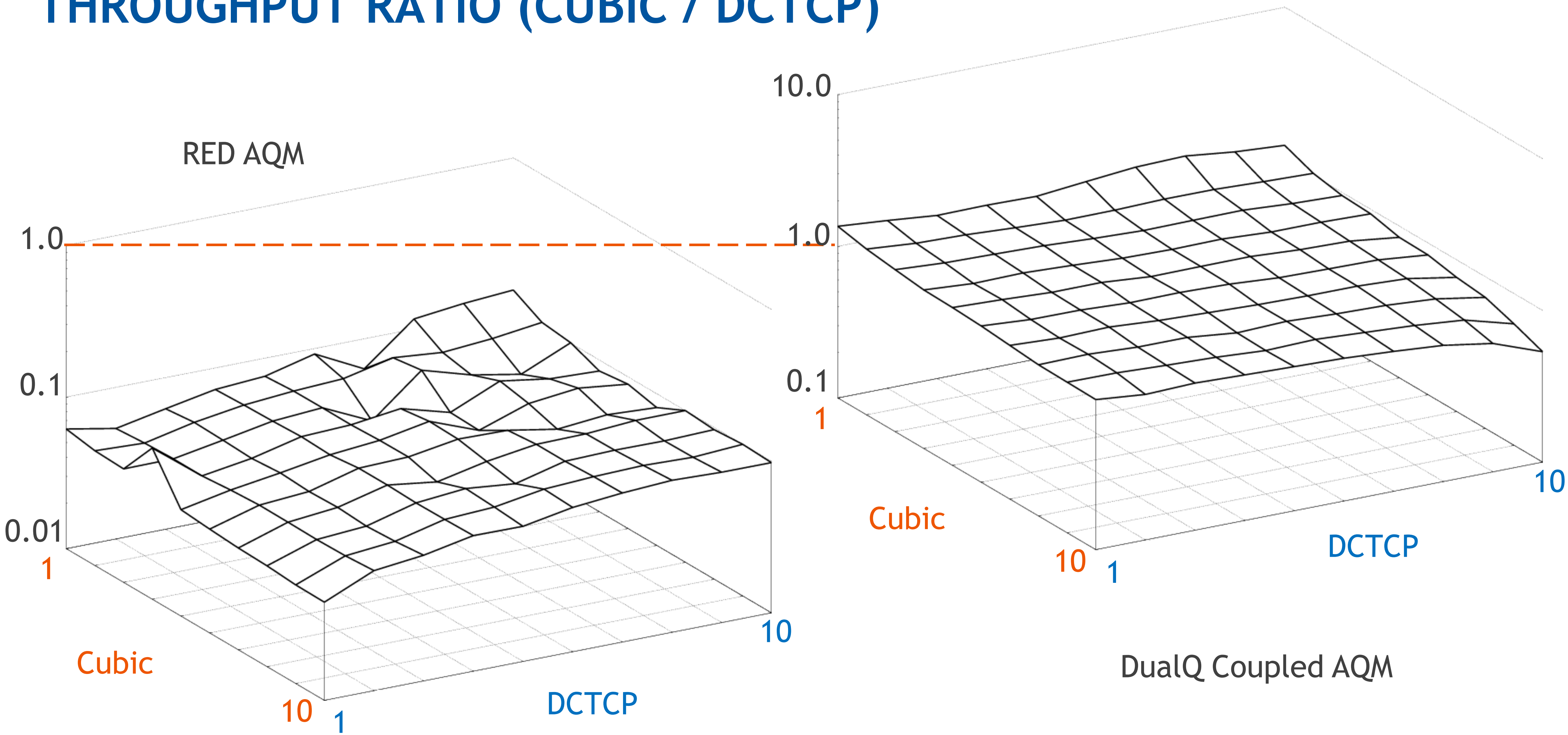
(450 Kbytes, 90/w ms)

Y-axis: autoscale count packets



R ⁄ T E
Reducing Internet Transport Latency

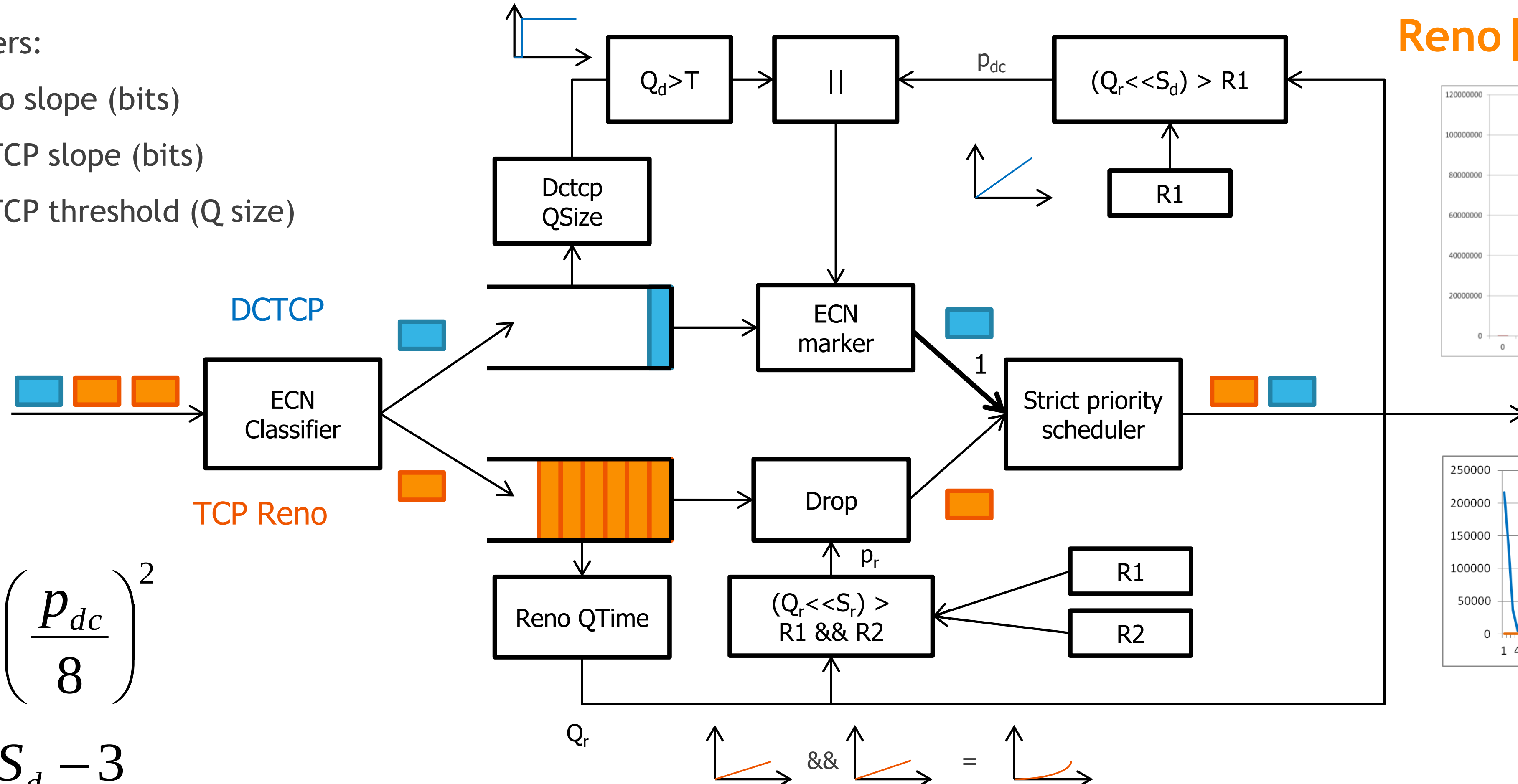
THROUGHPUT RATIO (CUBIC / DCTCP)



DETAILED IMPLEMENTATION

3 parameters:

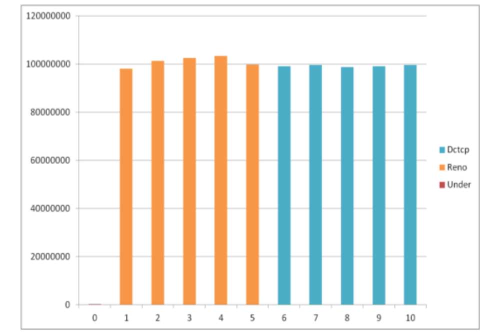
- Reno slope (bits)
- DCTCP slope (bits)
- DCTCP threshold (Q size)

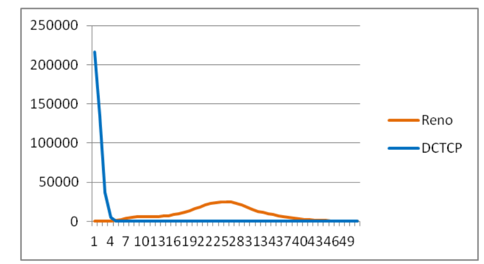


$$p_{reno} = \left(\frac{p_{dc}}{8}\right)^2$$

$$S_r = S_d - 3$$

Reno | Cubic DCTCP



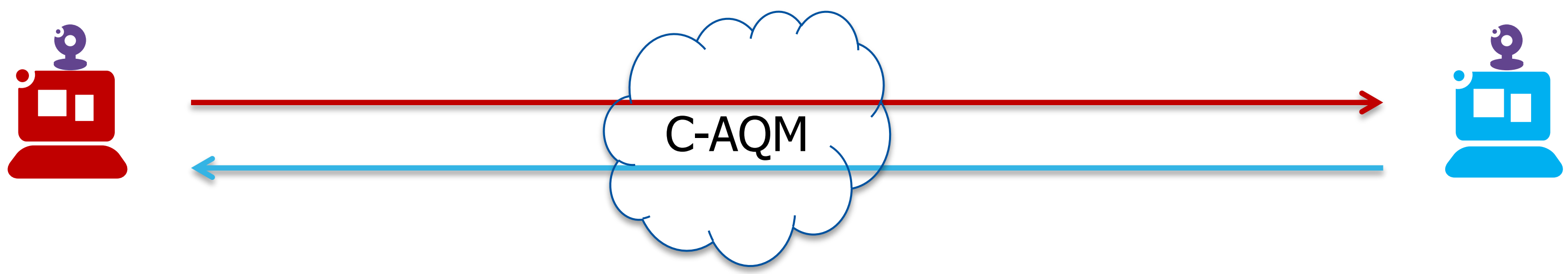


Reducing Internet Transport Latency

ADAPTIVE INTERACTIVE APPLICATIONS

- Panoramic interactive video

- Video/Voice conferencing



- Remote control, ….

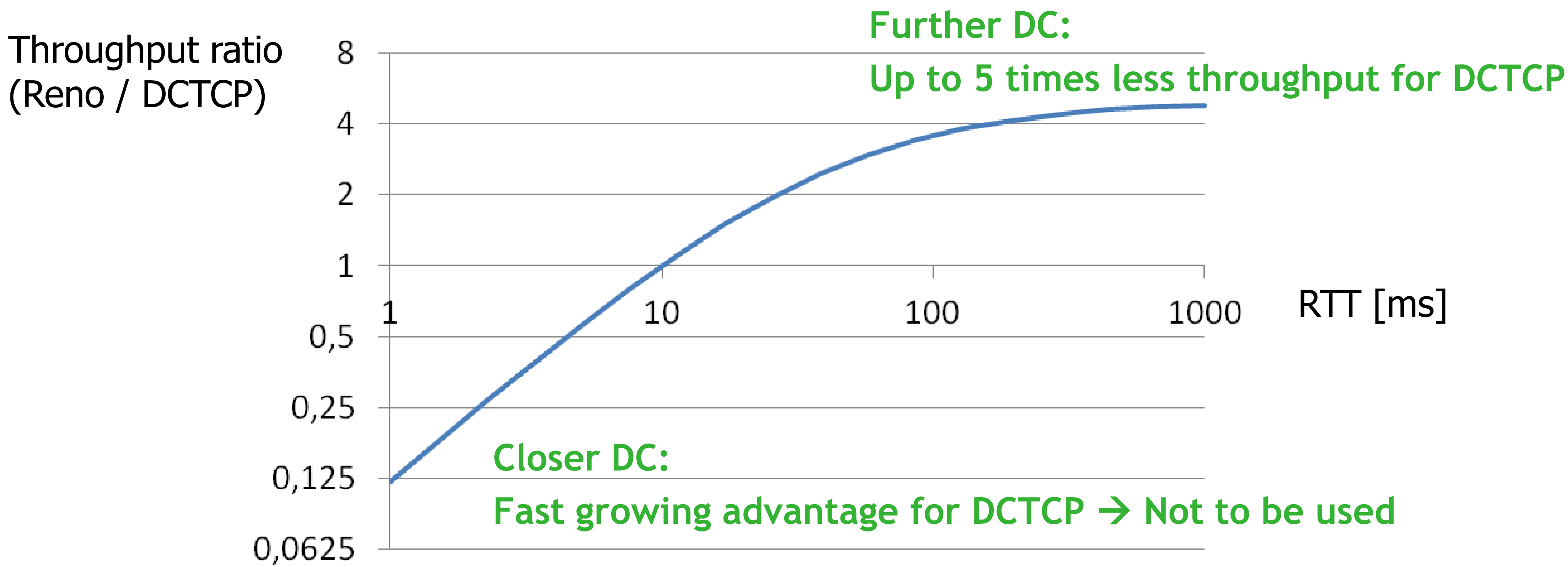
FUTURE WORK & CONCLUSIONS ?

- Dynamic behaviour to be investigated (expected to be 5x better due to 5x latency reduction)
- Unmanaged Low Level network Service ➔ Native support for Adaptive Interactive Applications
- Better usage for ECN (marking can be more often than dropping)
  ➔ x/p relation for ECN based congestion controller (x determining the marking rate)
  ➔ p^2 relation between mark and drop in AQM
- Backwards compatible: DCTCP should respond to drop as Reno
  - Currently 3.18 Linux implementation fails on this aspect
- Steady state throughput fairness between DCTCP and Reno|Cubic
  - Only if DCTCP flows are terminated to nearby (local) datacenter
  - If longer RTT, DCTCP flows are getting lower throughput than Reno|Cubic
  - ➢ Reno|Cubic fallback if throughput is too low and base RTT is too long ?
  - ➢ Define a TCP congestion controller which is less (/not) dependent on the RTT ?

Questions

koen.de_schepper@alcatel-lucent.com

BASE RTT FAIRNESS
WHAT IF THE DATACENTER IS FURTHER OR CLOSER



Coupled AQM configured for 10ms base RTT and 40ms Reno queue time (1/5 RTT ratio)

RITE
Reducing Internet Transport Latency

DCTCP STEADY STATE THROUGHPUT WITH SLOPE-RED

Per "long" RTT: $W \leftarrow W + 1$ (1)

And also per RTT: $W \leftarrow W\left(1 - \frac{\alpha}{2}\right)$ (2)

In steady state if (1) is compensated $W \leftarrow W - 1 = W\left(1 - \frac{1}{W}\right)$ so from (2) if $\frac{\alpha}{2} = \frac{1}{W}$ (3)

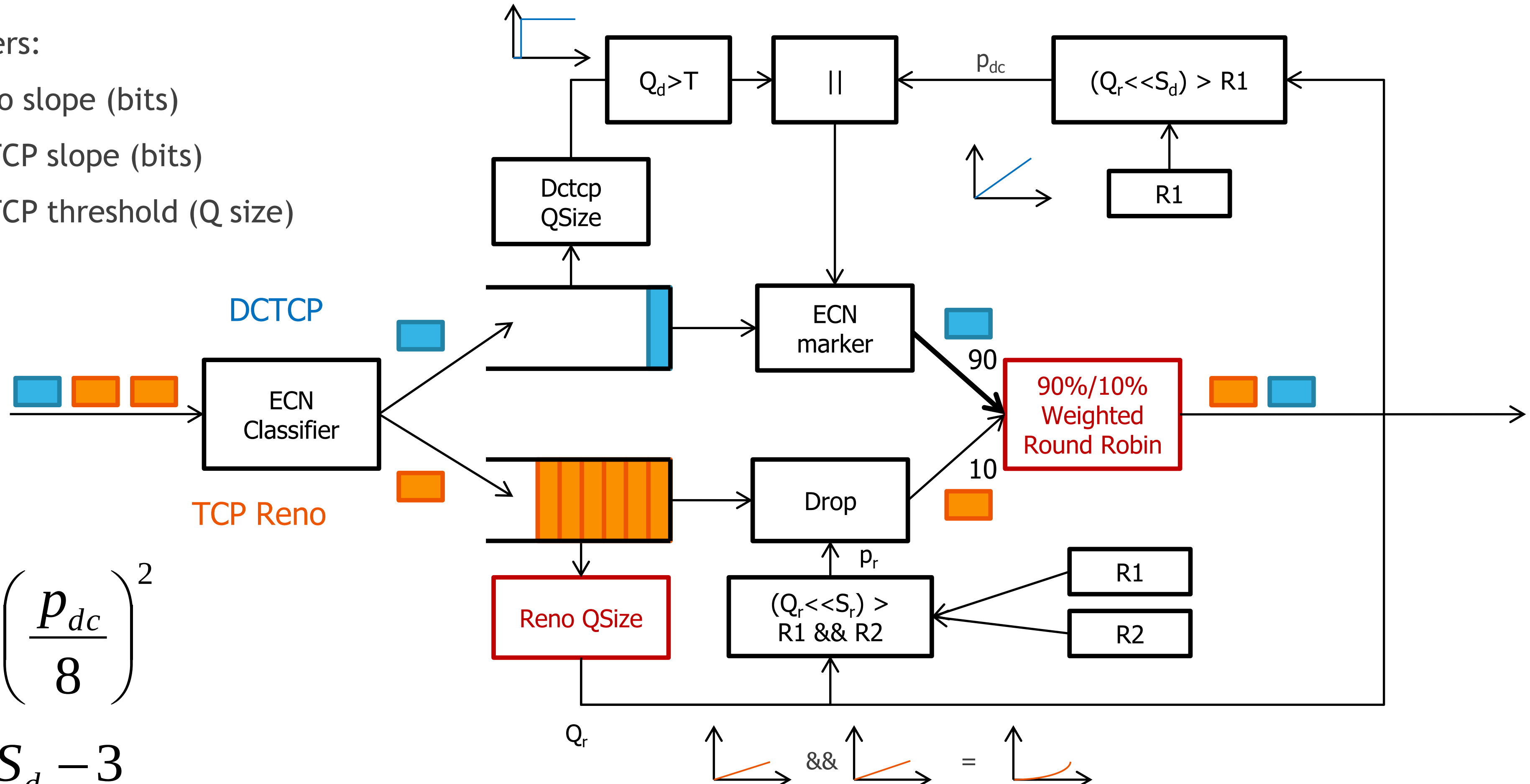
As $\alpha \leftarrow (1-g)\alpha + gp$, if p is stable in steady state $\alpha = p$ (4)

The instantaneous rate $r = \frac{W}{rtt}$ (5) thus (3,4,5) $r = \frac{2}{p \cdot rtt}$

QUEUE SIZE BASED COUPLED AQM

3 parameters:

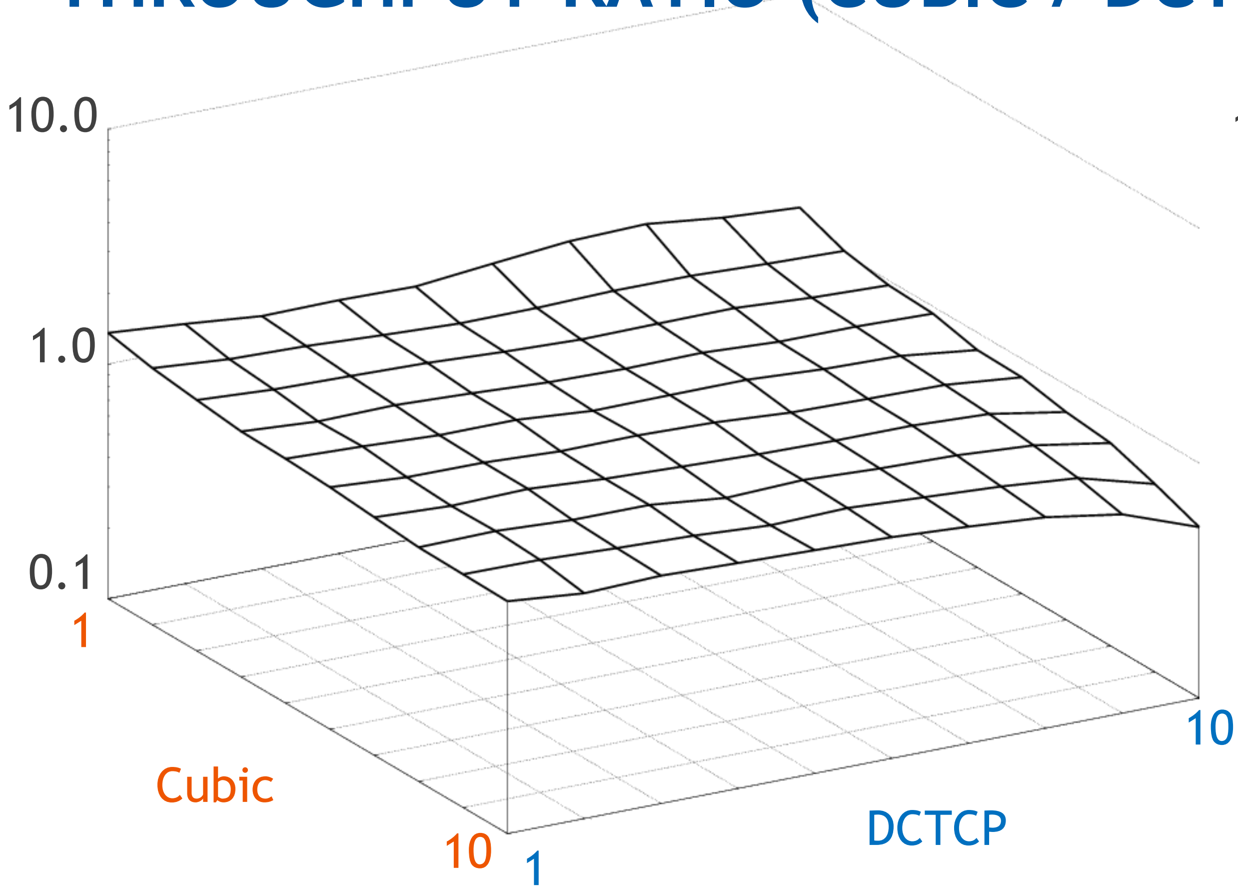
- Reno slope (bits)
- DCTCP slope (bits)
- DCTCP threshold (Q size)



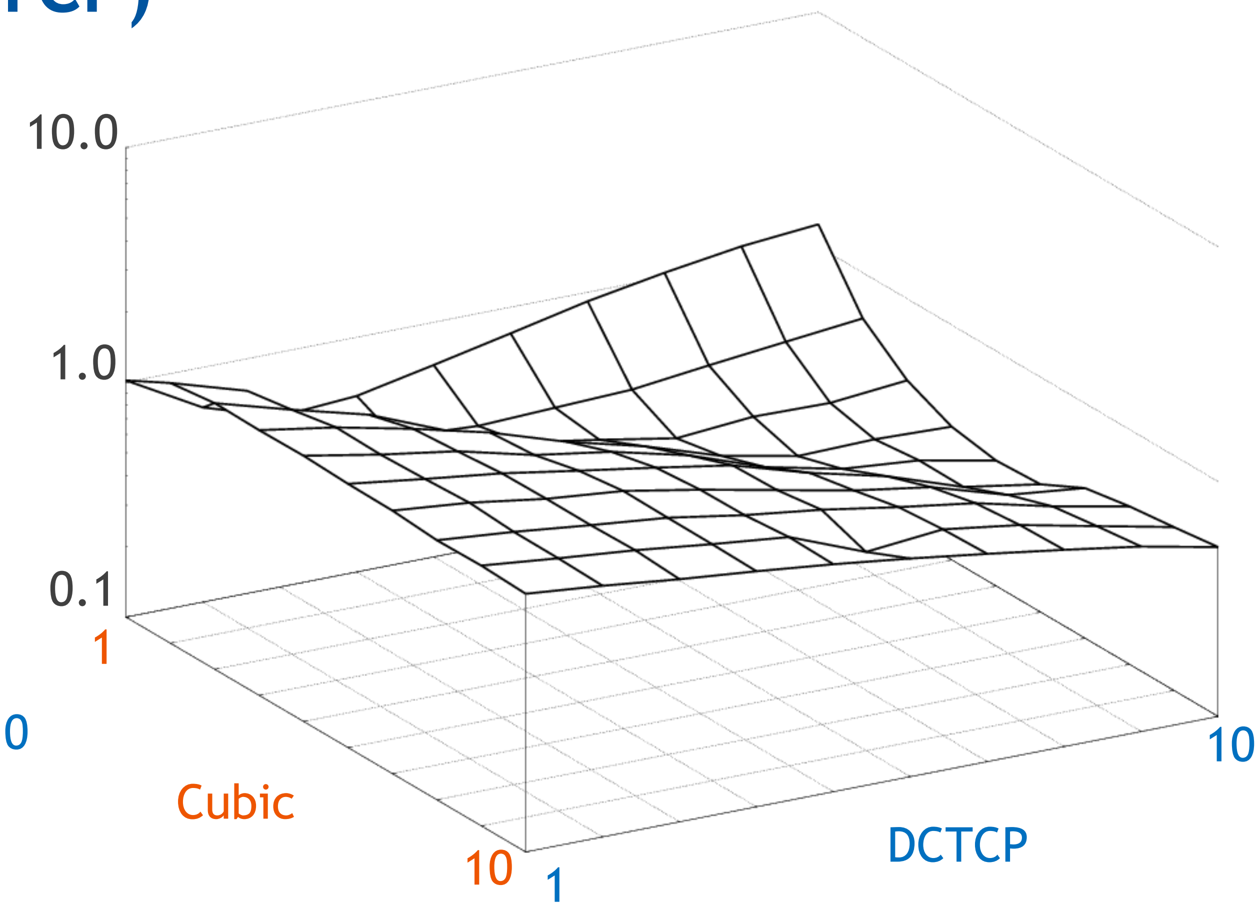
$$p_{reno} = \left(\frac{p_{dc}}{8}\right)^2$$

$$S_r = S_d - 3$$

QUEUE SIZE BASED COUPLED AQM THROUGHPUT RATIO (CUBIC / DCTCP)



Qtime - DualQ Coupled AQM



Qsize - DualQ Coupled AQM