Low Latency MultiPath TCP

Morteza Kheirkhah

University of Edinburgh, UK

Data Centre Network (DCN)

- **Diverse applications with diverse communication patterns and requirements**
  - Some apps are bandwidth hungry (online file storage)
  - Other apps are latency sensitive (online search)

- **Short flow dominance**
  - Majority of flows are short-lived with deadline in flow completion time (FCT).
  - Majority of data volumes come from a few long flows.

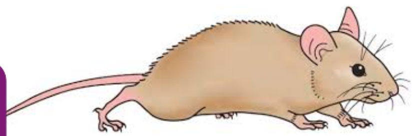
Data centers exhibit a highly dynamic network

Data Centre Basic Problem

- **Persistent Congestion:** Two or more long flows collide on a link (due to poor load-splitting of ECMP routing)
  - Low overall network throughput
- **Transient Congestion:** Many short flows collide on a link
  - High queuing delays and packet drop probability
  - Latency sensitive short flows miss their deadlines

Existing Solutions

Transient Congestion

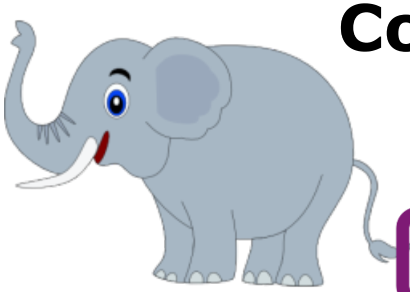


DCTCP
(SIGCOMM ’10)
D²TCP
(SIGCOMM ’12)

Low latency
for short flows

XMP
(coNEXT ’13)

Persistent Congestion



MPTCP
(SIGCOMM ’11)
Hedera
(NSDI ’10)

High throughput
for long flows

These solutions do not coexist with each other

Existing Solutions Goals/Problems

- **DCTCP** tries to keep a low buffer occupancy of links, achieving a low FCT for short flows
  - **Single-path transport protocol**

- **MPTCP** tends to fully occupy the network buffers, achieving high goodput for long flows
  - **High queueing delays and packet drop probability**

- **XMP** tries to deal with the latency-throughput trade-off by exploiting MPTCP and ECN
  - **Does not coexist with any flows other than itself**
  - **Several interdependent parameters to adjust**

Existing Solutions
Congestion Control Algorithm

- **DCTCP algorithm is, in short:**
  - Each ACK, increases **w** by **1/w**
  - Each loss, decreases **w** by **1/2**
  - Marked ACK, adjusts **w** to **w × (1 - α/2)** [once per rtt]

$$\alpha = (1 - g) \times \alpha + (g) \times F$$

- **MPTCP algorithm is, in short:**
  - Each ACK on subflow(s),increases $\mathbf{w_s}$ by min($\mathbf{a/w_{total}}$ **,** $\mathbf{1/w_s}$)
  - Each loss, decreases $\mathbf{w_s}$ by **1/2**

$$a = w_{total} \frac{Max_r(w_r/rtt_r^2)}{(\sum_r(w_r/rtt_r))^2}$$

- **XMP algorithm is, in short:**
  - Every round on subflow(s), increases $\mathbf{w_s}$ by $\mathbf{a_s}$ *** mss**
  - Every loss, decrease $\mathbf{w_s}$ by **1/2**
  - Marked ACK, decrease $\mathbf{w_s}$ by **1/ß** [once per rtt]

$$a_s = \frac{(w_s/rtt_s)}{\sum_s(w_s/rtt_s)}$$

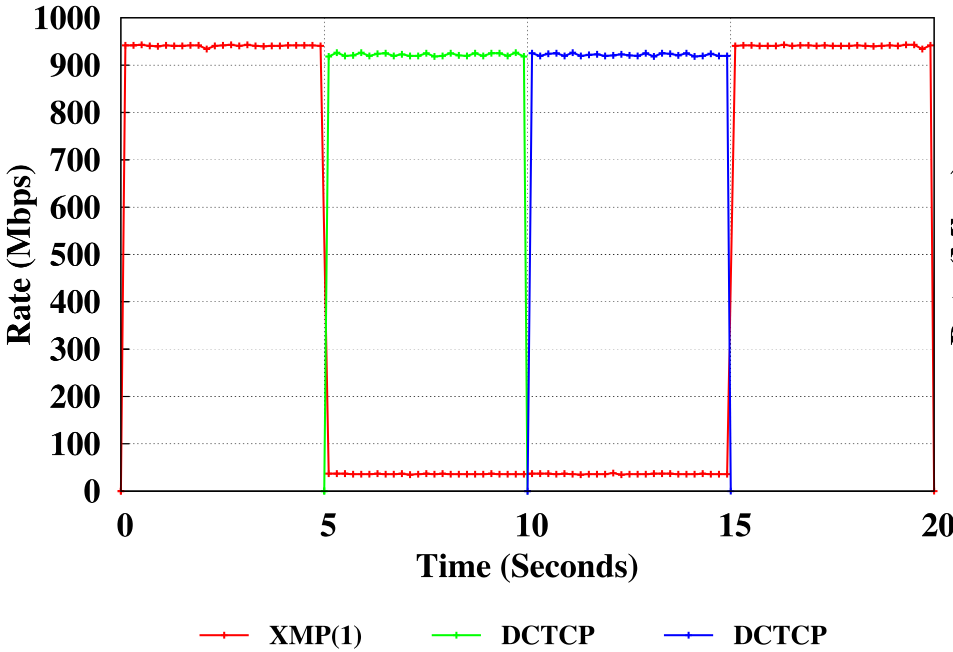
Does XMP Coexist with DCTCP?

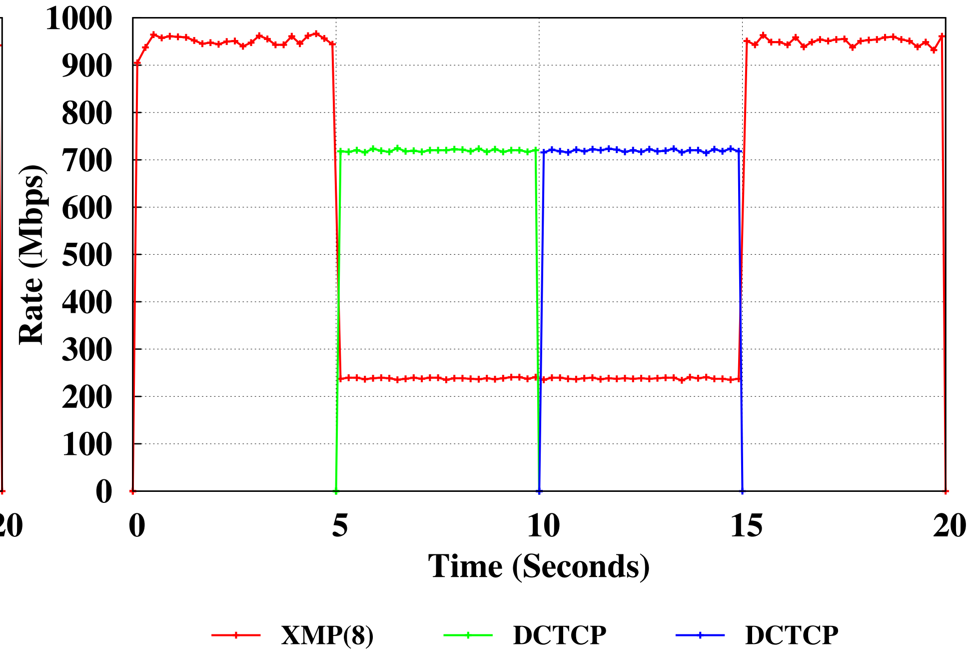
Network Topology

```
S1----|---|     |---|
S2----| R |----| D |
S3----|---|     |---|
```

Simulation Parameters

```
Rate = 1Gbps
RTT  = 240us
K    = 10pkt
```





XMP reduces its CWND largely and rapidly

Our approach

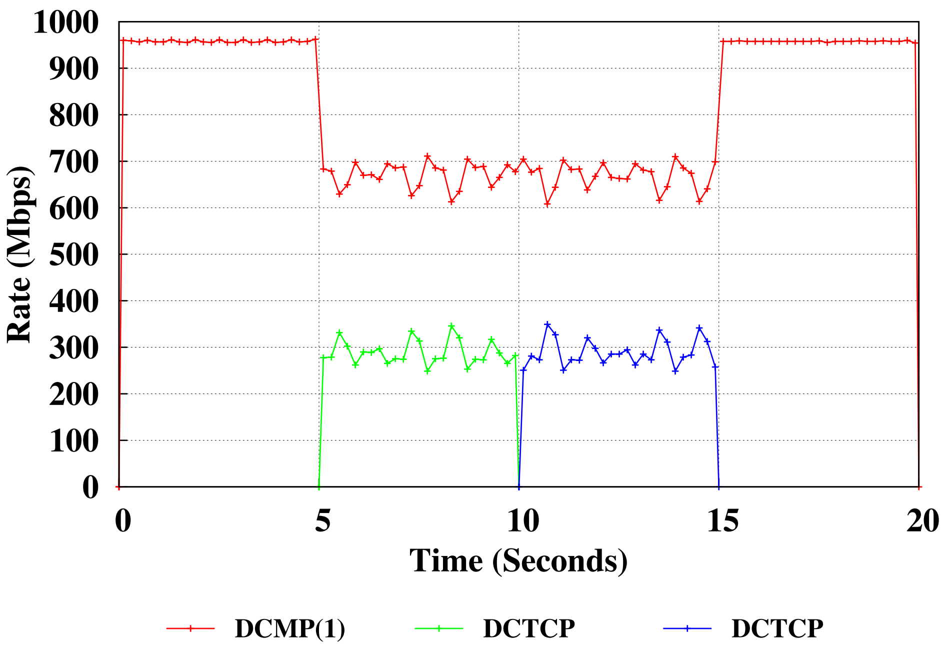
- **Why not running DCTCP with MPTCP?**
  – We call it **D**ata **C**enter **M**ulti**P**ath **T**CP (**DCMPTCP**)

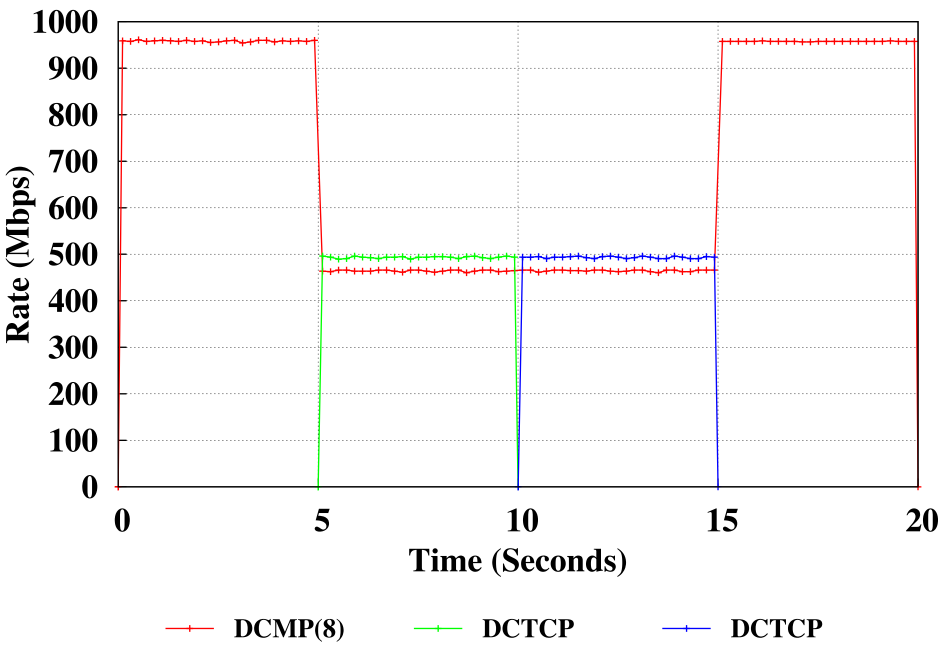
- **Two questions need to be examined:**
  1. Does DCMPTCP (with Linked Increases) balance traffic without packet losses?
  2. Does DCMPTCP coexist with DCTCP?

Does DCMPTCP Coexist with DCTCP?

Network Topology

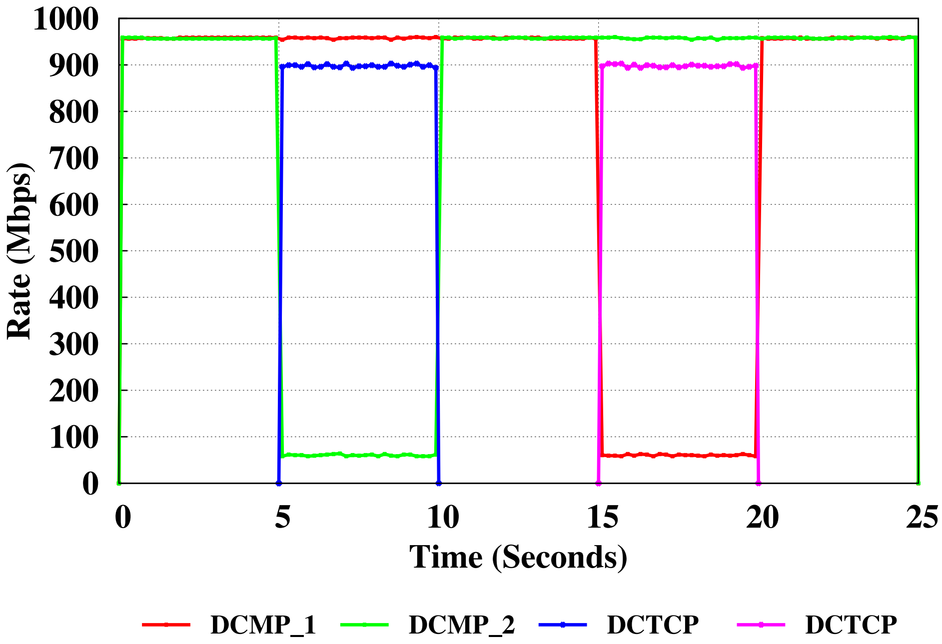
```
S1----|---|    |---|
S2----| R |----| D |
S3----|---|    |---|
```

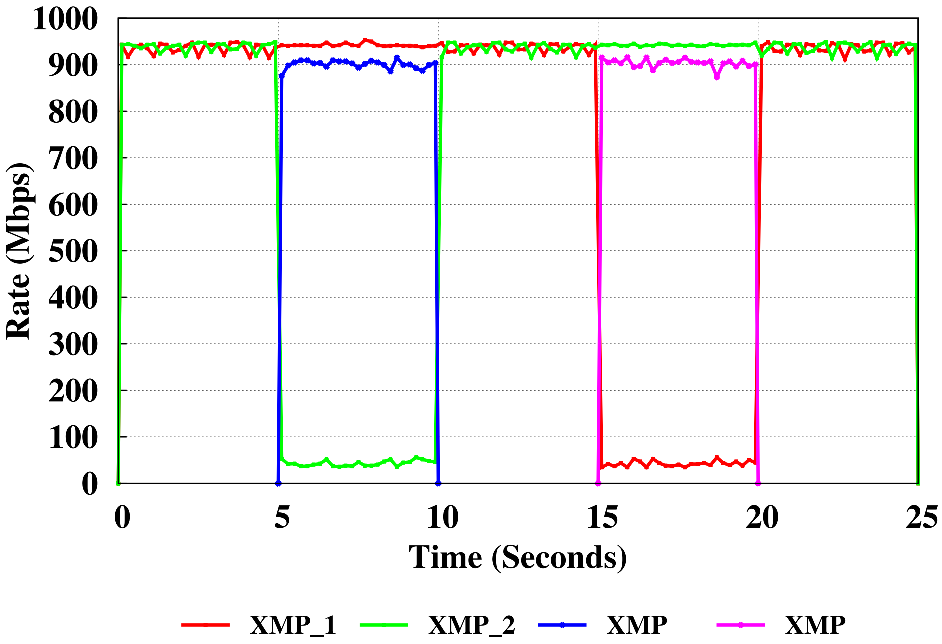




Does DCMPTCP move traffic without packet losses?

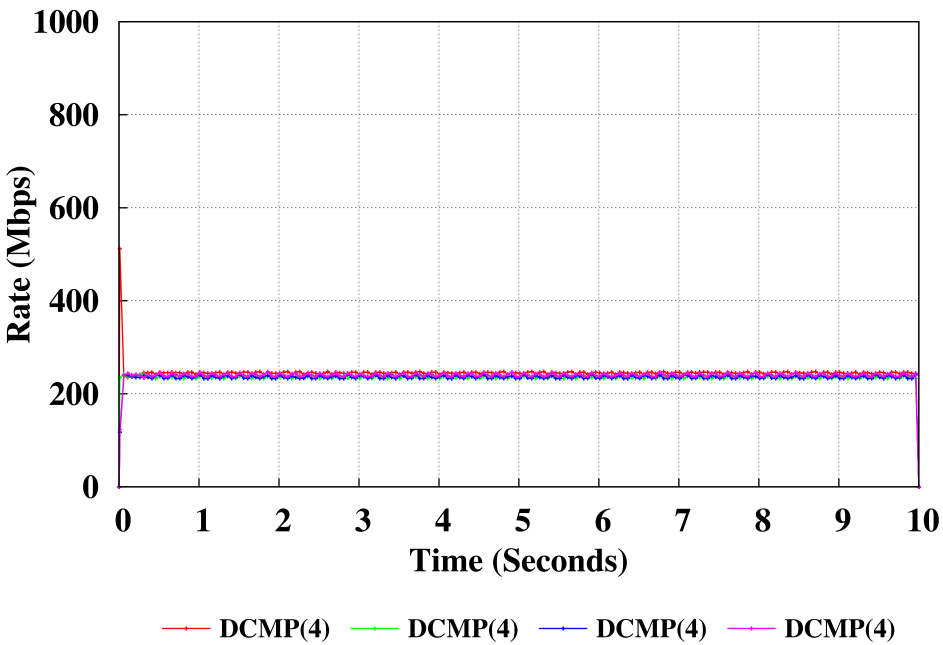
```
       S2
       |
  /----R1----\
S1            D
  \----R2----/
       |
       S3
```

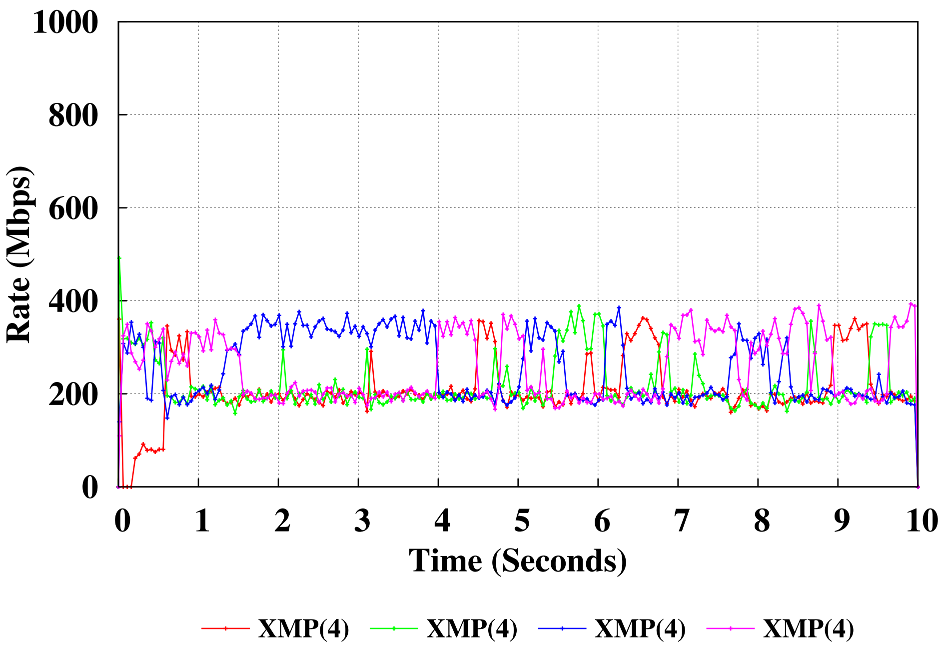




Does DCMPTCP preserve network fairness between flows?

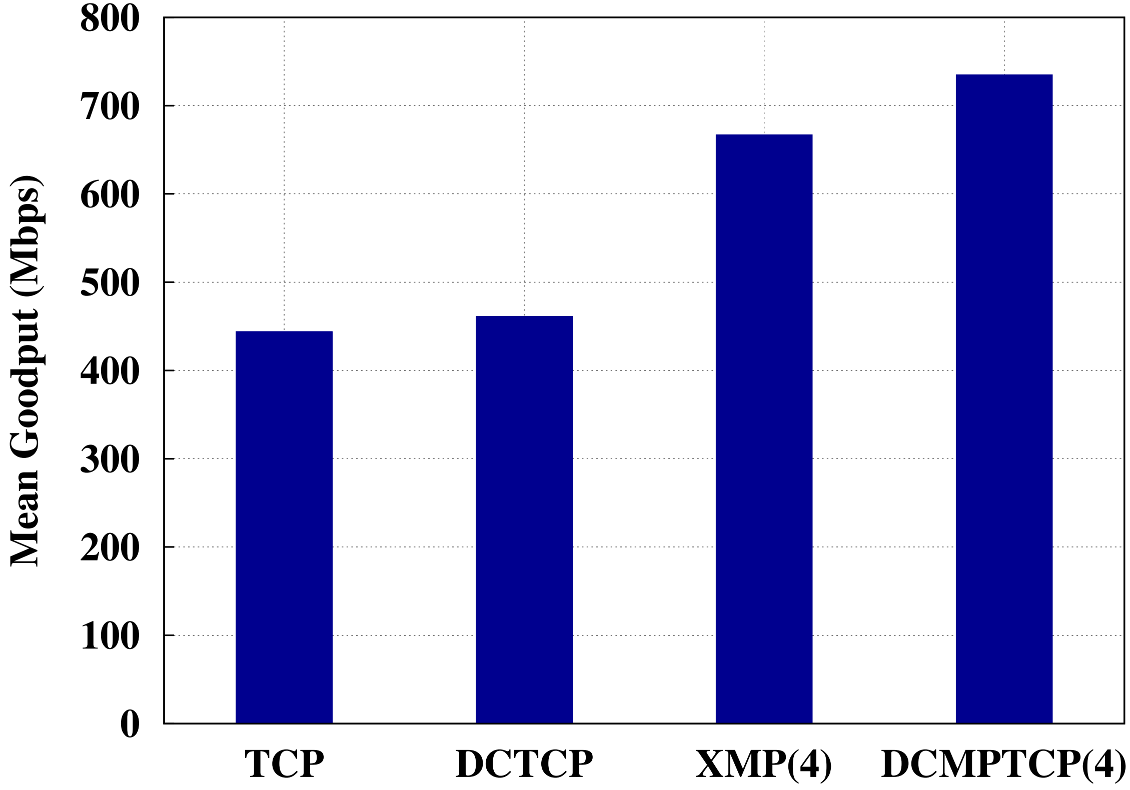
```
S1----|---|     |---|
S2----|   |     |   |
S3----| R |-----| D |
S4----|   |     |   |
      |---|     |---|
```





DCN Experiment 1
Overall Goodput (of long flows)

DCMPTCP achieves better utilization than XMP



FatTree, 128 nodes , full bisection bandwidth, random permutation
all flows are long-lived

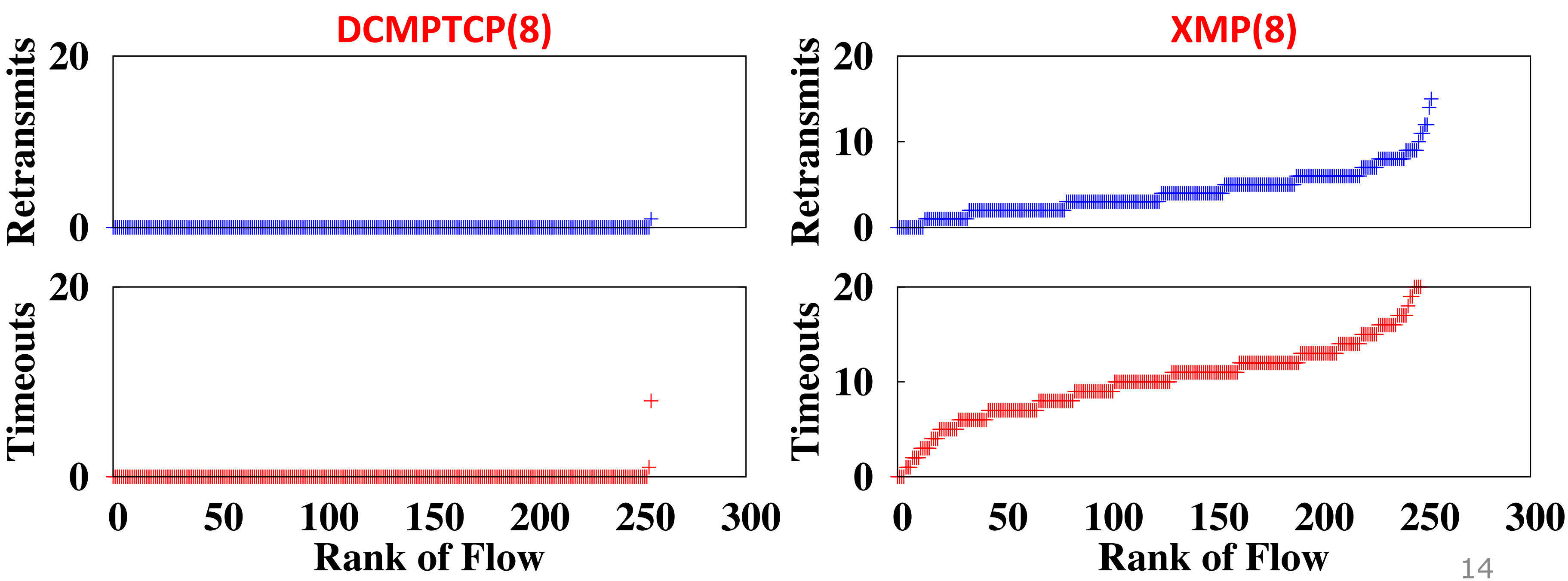
DCN Experiment 2
Fairness (XMP vs DCMPTCP)

XMP with 2 subflows achieves less than DCTCP

Simulation Setup (MPTCP flows with 2 subflows)	Mean Goodput (Mbps)
XMP : DCTCP	356 : 431
DCMPTCP : DCTCP	430 : 391

DCN Experiment 3
Incast (mix of long and short flows)

Subflows	Scheme	Long Flow – Avg. GP	JCT - 50th	JCT - 99th
2	DCMPTCP	644 (Mbps)	16 (ms)	208 (ms)
	XMP	589 (Mbps)	14 (ms)	1507 (ms)
8	DCMPTCP	772 (Mbps)	18 (ms)	612 (ms)
	XMP	650 (Mbps)	15 (ms)	1510 (ms)



Incast summary

More subflows -> More background traffic -> Less short flows can be accomodated

More long flows & more subflows -> buffer occupancy can’t be controlled

We think DCMPTCP with two subflows works well in a wide range of network scenarios, including Incast

Thank You!