QUIC Throughput and Fairness over Dual Connectivity

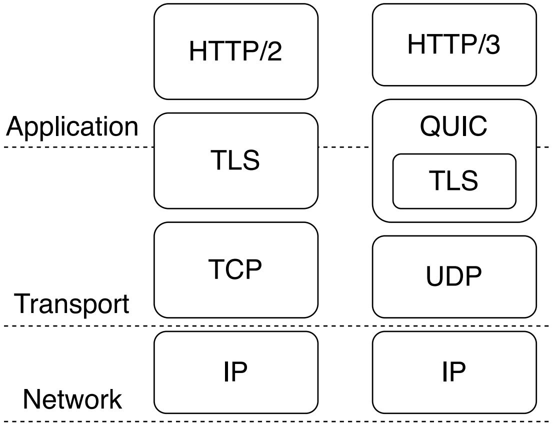
David Hasselquist
Christoffer Lindström
Nikita Korzhitskii
Niklas Carlsson
Andrei Gurtov

LINKÖPING UNIVERSITY

Proc. IEEE MASCOTS workshop, Nov. 2020

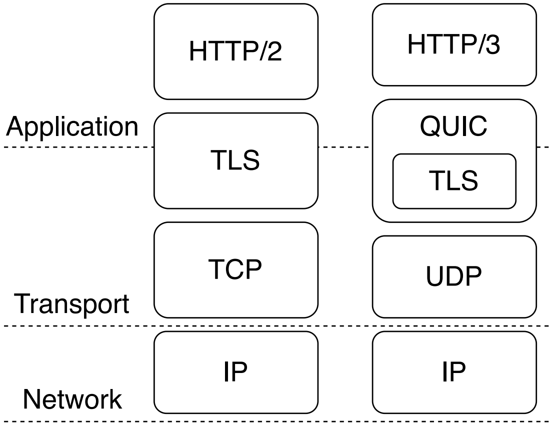
QUIC

- New transport protocol
  - Introduced 2013 by Google
  - Continued by IETF to become HTTP/3
- Multi-layer
  - Reliable in userspace over UDP
  - Rapid deployment
  - 0-RTT
  - Per-stream flow control



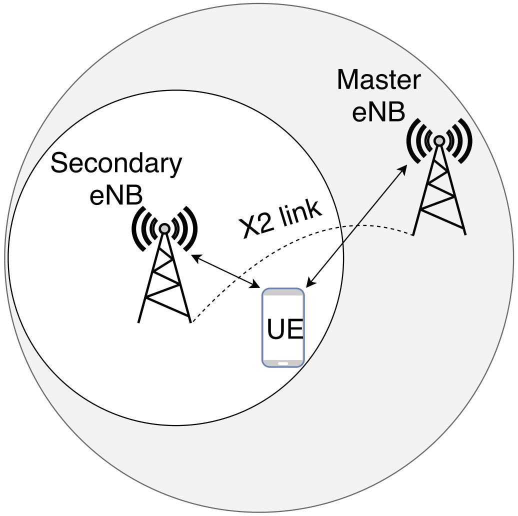
QUIC

- New transport protocol
  - Introduced 2013 by Google
  - Continued by IETF to become HTTP/3
- Multi-layer
  - Reliable in userspace over UDP
  - Rapid deployment
  - 0-RTT
  - Per-stream flow control
- Widespread adoption
  - Used by Google, YouTube, Facebook, and more…



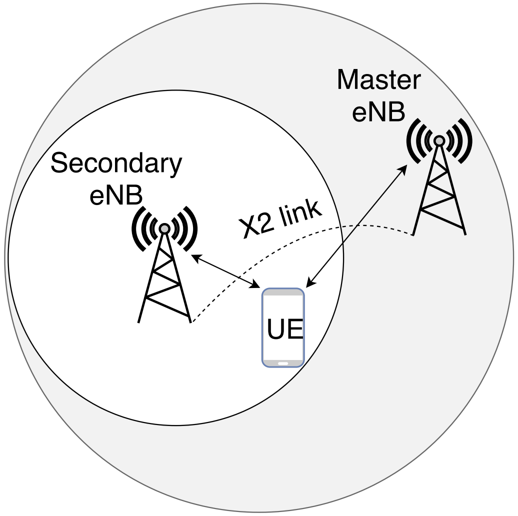
Dual Connectivity

- Multi-connectivity technique
- Accelerating transition to 5G
- Multiple future uses
  - Throughput
  - Reliability



Dual Connectivity

- Multi-connectivity technique
- Accelerating transition to 5G
- Multiple future uses
  - Throughput
  - Reliability
- Splits traffic at link layer in PDCP sublayer



Multipath

Application	
Multipath TCP	
IP	IP
PDCP	PDCP
RLC	RLC
MAC	MAC
Physical	Physical

Dual Connectivity

Application	
TCP	
IP	
PDCP	
RLC	RLC
MAC	MAC
Physical	Physical

Carrier Aggregation

Application	
TCP	
IP	
PDCP	
RLC	
MAC	
Physical	Physical

Motivation

- Dual Connectivity invisible to QUIC and TCP
  - Jitter, reordering can significantly impact performance

- Understanding two "new" technologies together
  - Dual Connectivity parameters
  - Network conditions (bandwidth, delay, loss)

Contributions

- First performance evaluation of QUIC and Dual Connectivity
  - Throughput
  - Fairness

Contributions

- First performance evaluation of QUIC and Dual Connectivity
  - Throughput
  - Fairness

- Comparison with TCP counterparts
  - 2 QUIC implementations (aioquic, ngtcp2)
  - NewReno and CUBIC
  - LTE bandwidth trace

Contributions

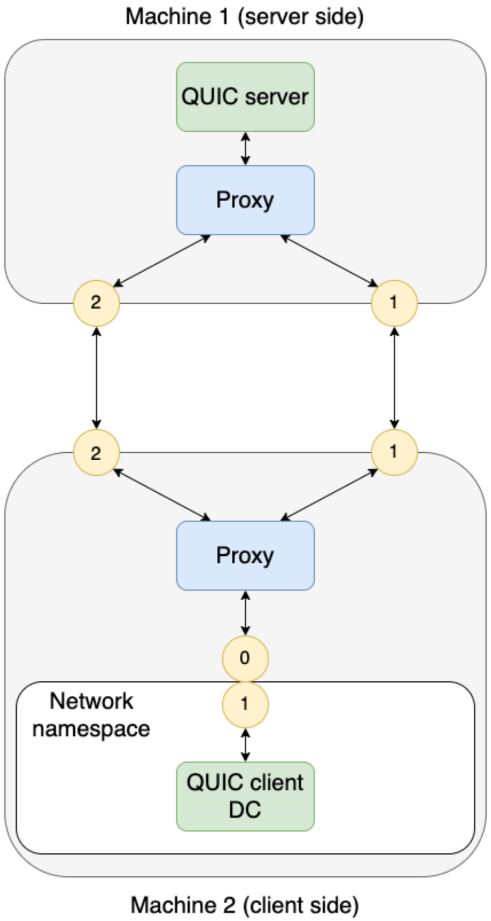
- First performance evaluation of QUIC and Dual Connectivity
  - Throughput
  - Fairness

- Comparison with TCP counterparts
  - 2 QUIC implementations (aioquic, ngtcp2)
  - NewReno and CUBIC
  - LTE bandwidth trace

- Insights for network operators

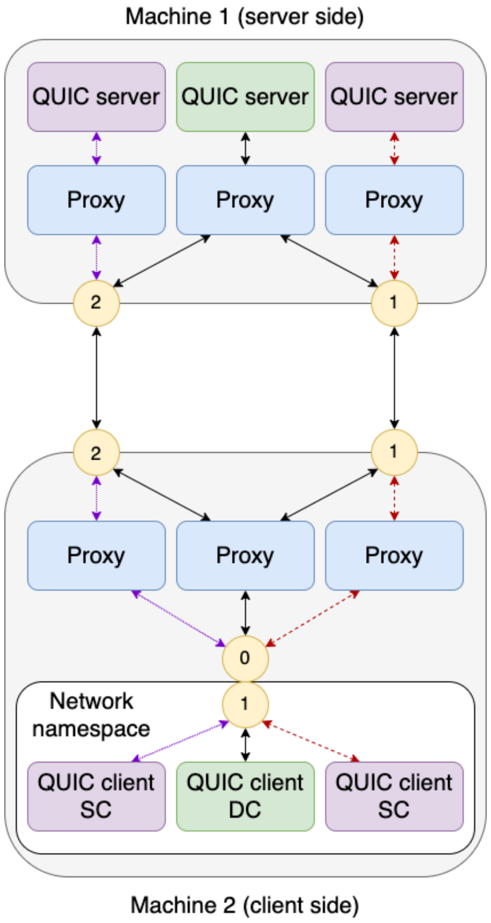
Testbed throughput

- Two machines, 2 interfaces each
- PDCP proxy



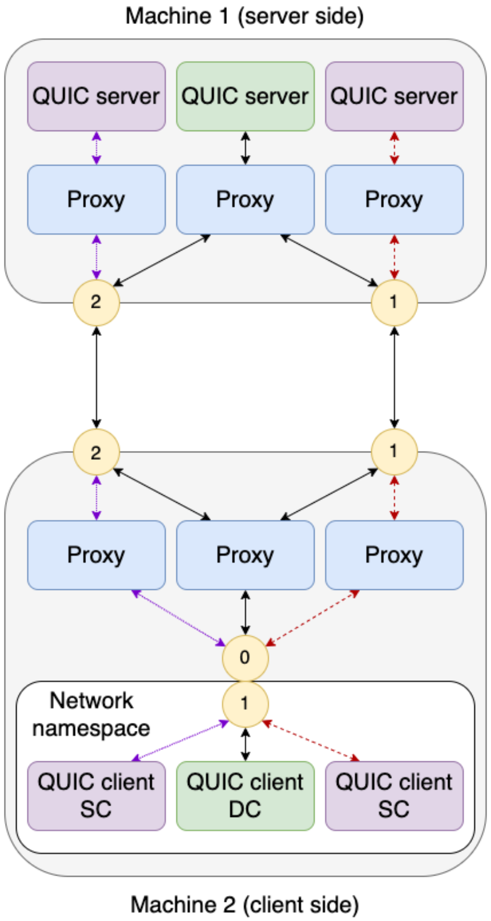
Testbed fairness

- 3 QUIC servers/clients
  - 1 Dual Connectivity
  - 2 Single Connectivity



Testbed fairness

- 3 QUIC servers/clients
  - 1 Dual Connectivity
  - 2 Single Connectivity
- Jain's fairness index (JFI)



Parameters

- DC batch size
- DC ratio (batch split)
- Bandwidth ratio
  - With and without matching DC ratio
- Delay ratio
  - Low and high delay
- Random loss
  - With and without packet duplication

Parameters

- DC batch size
- DC ratio (batch split)
- Bandwidth ratio
  - With and without matching DC ratio
- Delay ratio
  - Low and high delay
- Random loss
  - With and without packet duplication

DC batch size 100
DC ratio 5:1

80 packets -> 20 packets -> …

Parameters

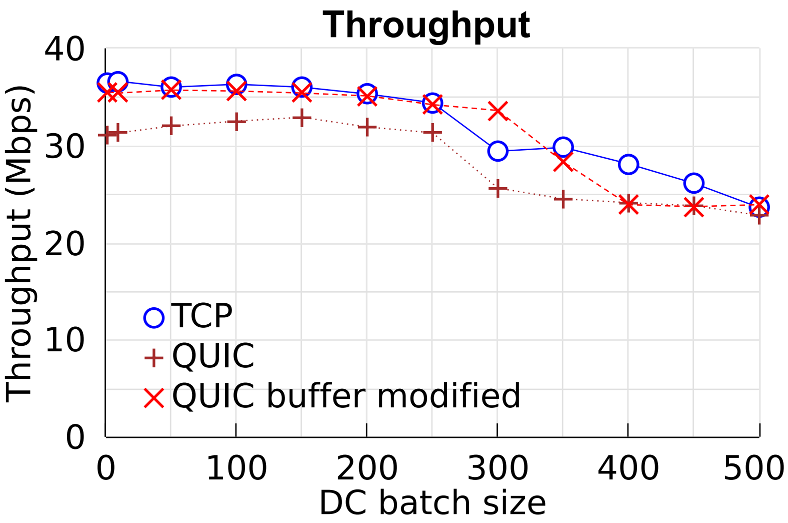
- DC batch size
- DC ratio (batch split)
- Bandwidth ratio
  - With and without matching DC ratio
- Delay ratio
  - Low and high delay
- Random loss
  - With and without packet duplication

DC batch size 100
DC ratio 5:1

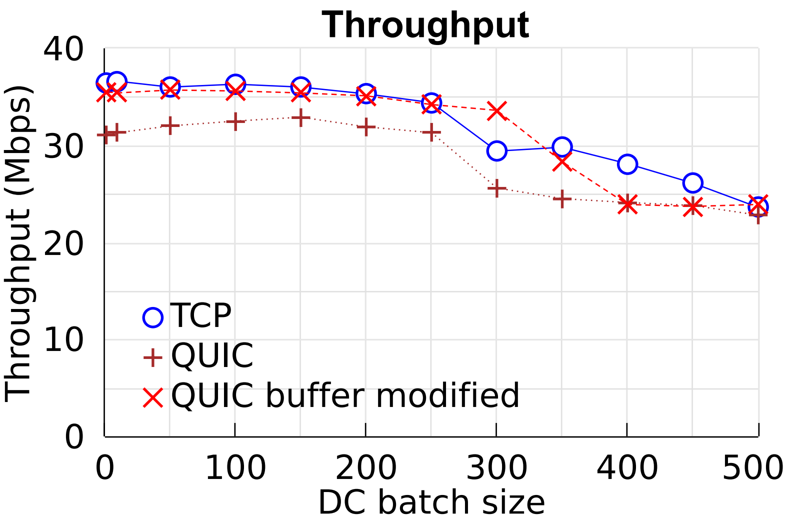
80 packets -> 20 packets -> …

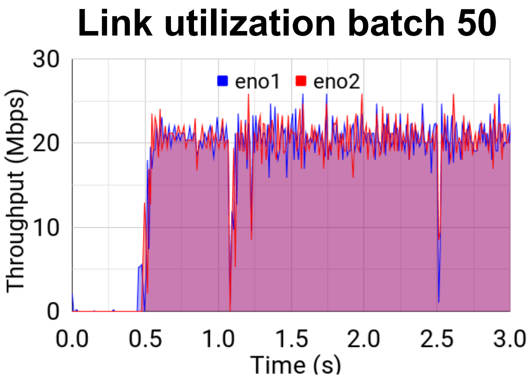
LINKÖPING UNIVERSITY

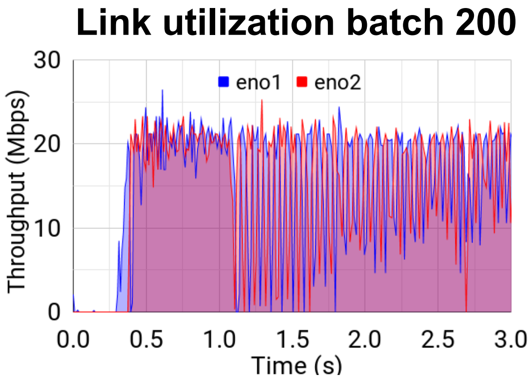
DC batch size

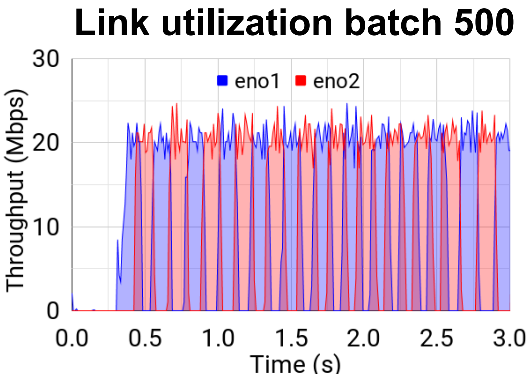


DC batch size

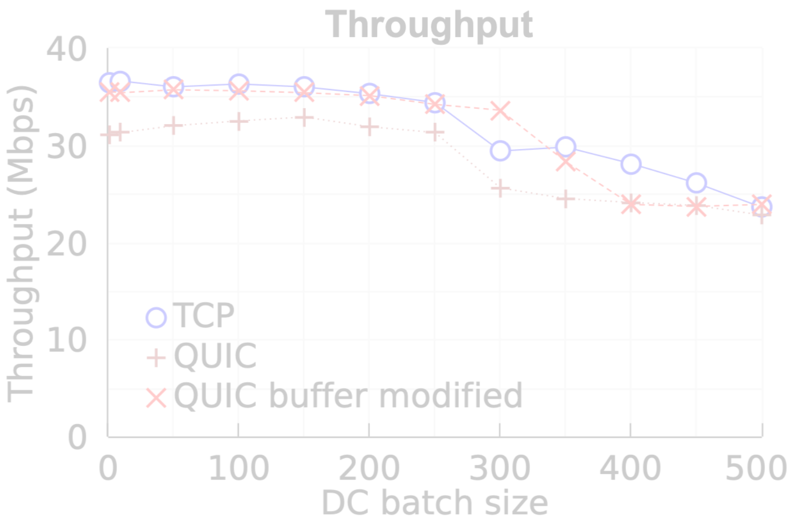


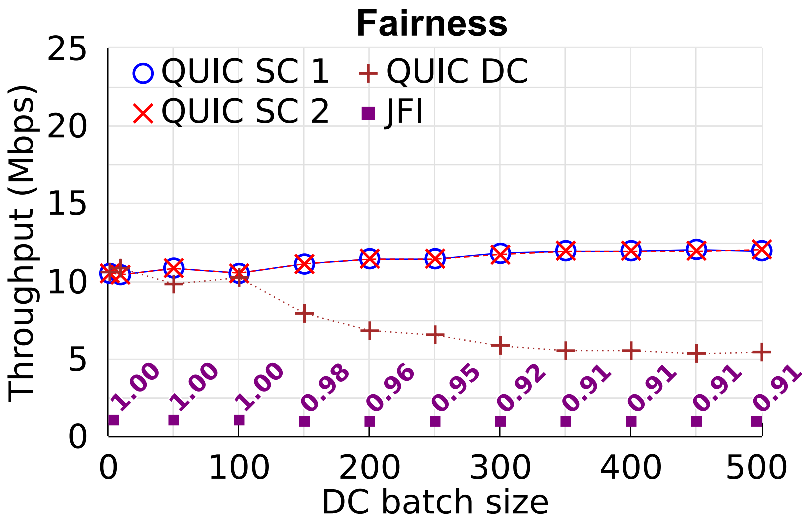


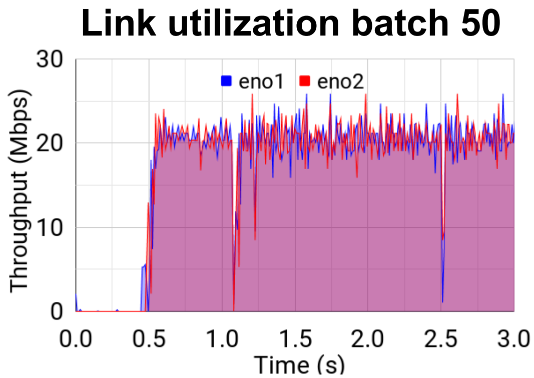


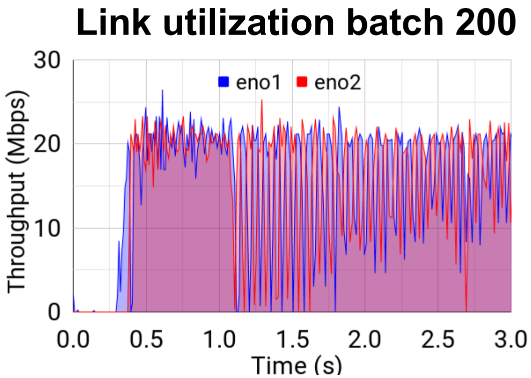


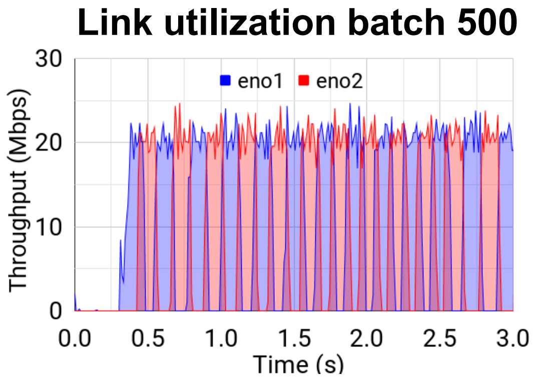
DC batch size



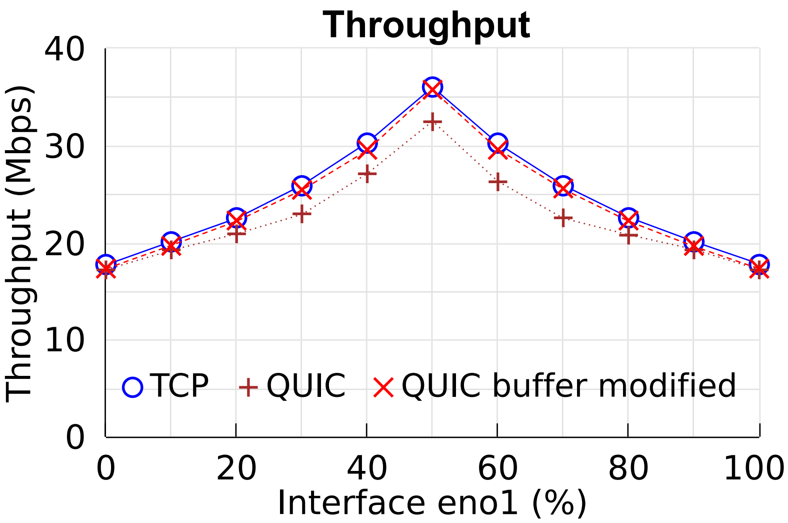




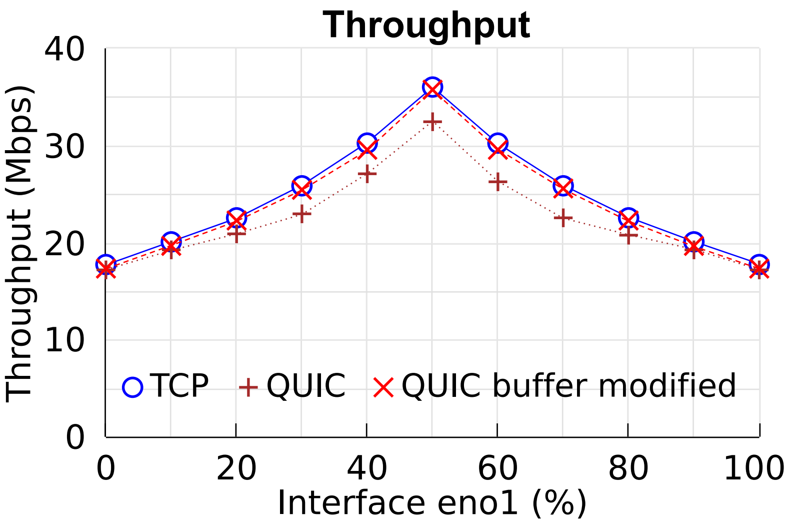


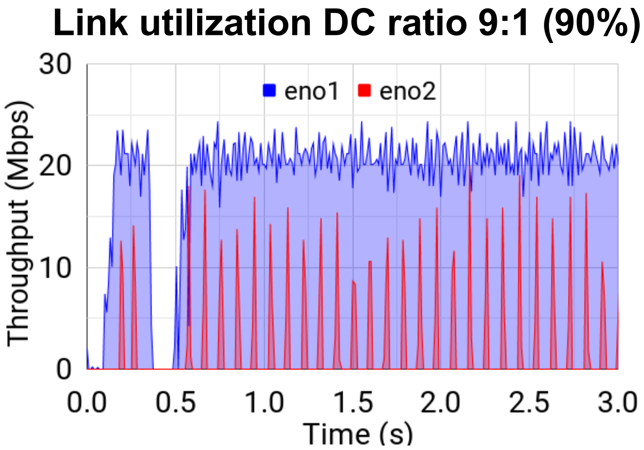


DC ratio (batch split)

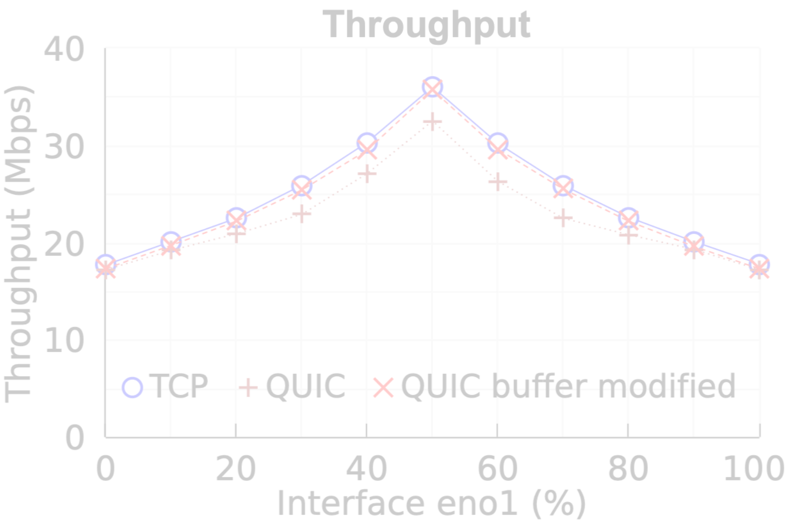


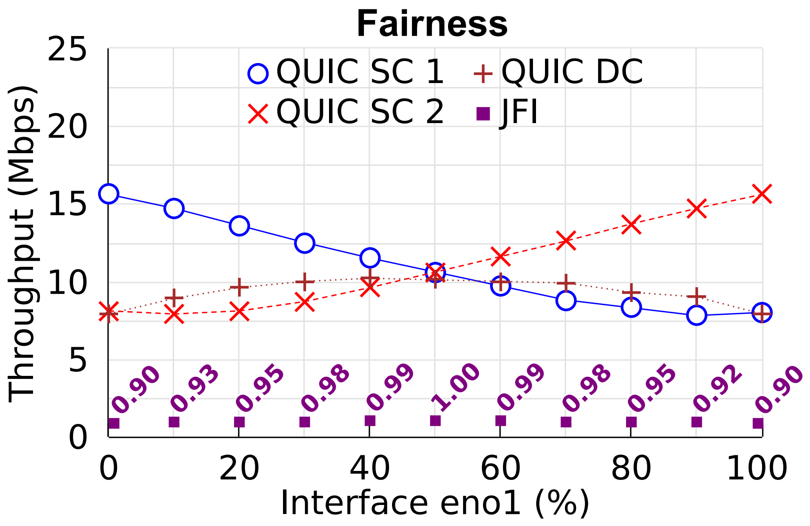
DC ratio (batch split)

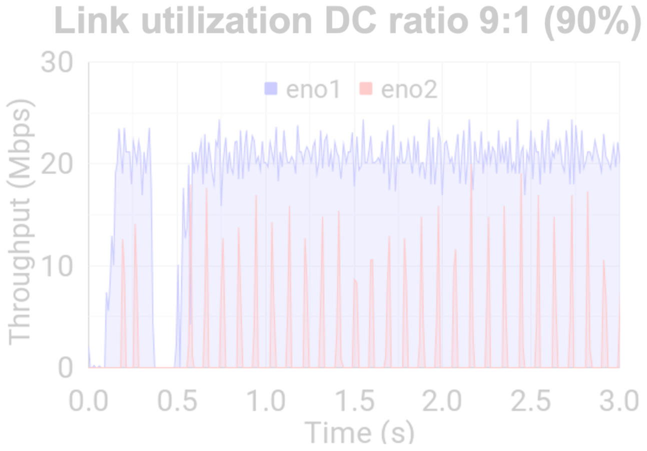




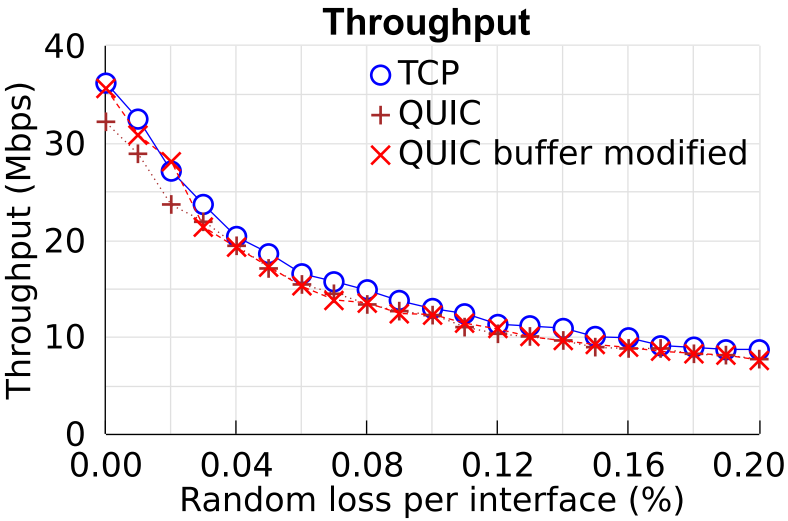
DC ratio (batch split)



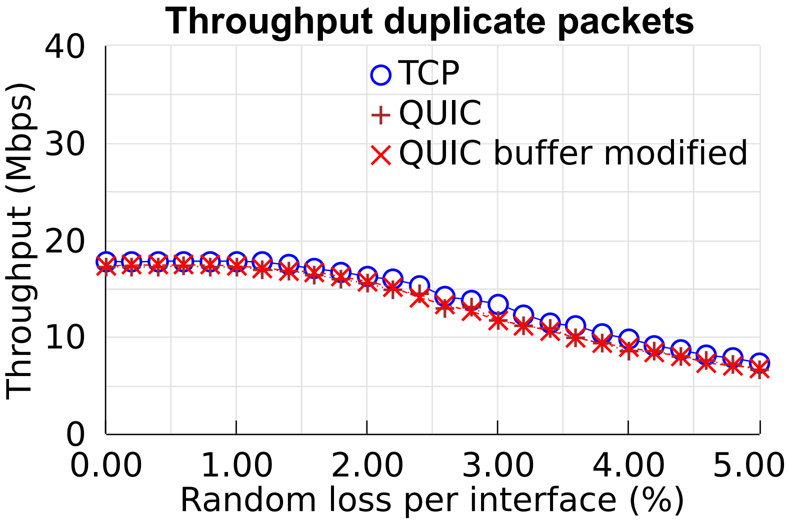




Random loss

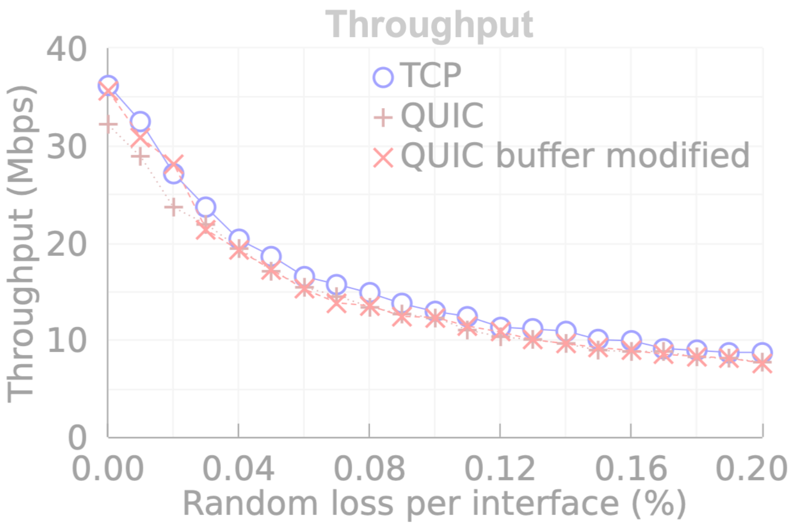


Throughput
Throughput (Mbps)
40
30
20
10
0
0.00
0.04
0.08
0.12
0.16
0.20
Random loss per interface (%)
○ TCP
+ QUIC
× QUIC buffer modified

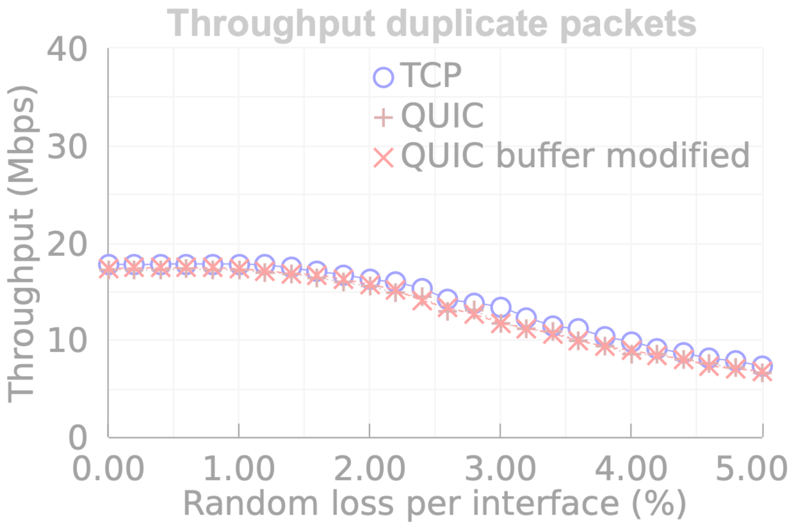


Throughput duplicate packets
Throughput (Mbps)
40
30
20
10
0
0.00
1.00
2.00
3.00
4.00
5.00
Random loss per interface (%)
○ TCP
+ QUIC
× QUIC buffer modified

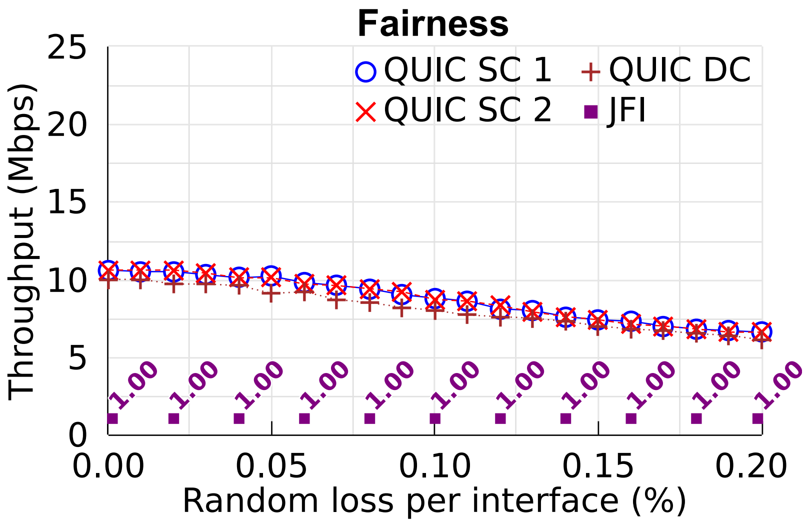
Random loss



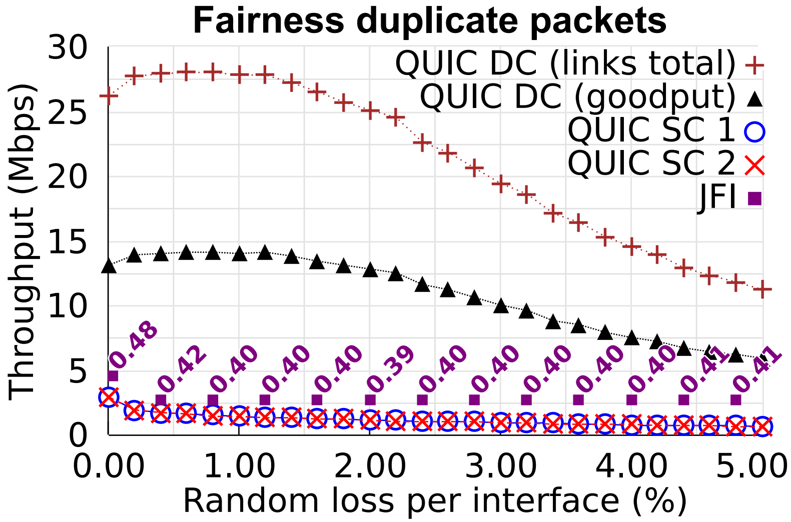
Throughput



Throughput duplicate packets



Fairness



Fairness duplicate packets

QUIC configuration/version

- QUIC implementation
  - Second QUIC implementation more aggressive, achieves higher throughput in general but unfair
  - Differences observed due to execution speed and pacer implementation

QUIC configuration/version

- QUIC implementation
  - Second QUIC implementation more aggressive, achieves higher throughput in general but unfair
  - Differences observed due to execution speed and pacer implementation

- Little to no differences when using:
  - CUBIC instead of NewReno
  - Trace-based bandwidth variation

Conclusions

- First performance evaluation of QUIC over DC

Conclusions

- First performance evaluation of QUIC over DC
- QUIC performs similarly, but not identical, to TCP over DC

Conclusions

- First performance evaluation of QUIC over DC
- QUIC performs similarly, but not identical, to TCP over DC
- QUIC can utilize the increased throughput/reliability of DC
  - If link properties similar and batch size small
  - Otherwise better to turn DC off

Conclusions

- First performance evaluation of QUIC over DC

- QUIC performs similarly, but not identical, to TCP over DC

- QUIC can utilize the increased throughput/reliability of DC
  - If link properties similar and batch size small
  - Otherwise better to turn DC off

- Optimal systemwide fairness
  - If symmetric link conditions and not duplicating packets

Thanks for listening!

QUIC Throughput and Fairness over Dual Connectivity

David Hasselquist, Christoffer Lindström,
Nikita Korzhitskii, Niklas Carlsson, Andrei Gurtov

LINKÖPING
UNIVERSITY