Adaptive FEC for Congestion Control

Varun Singh, Marcin Nagy, Jörg Ott, Lars Eggert

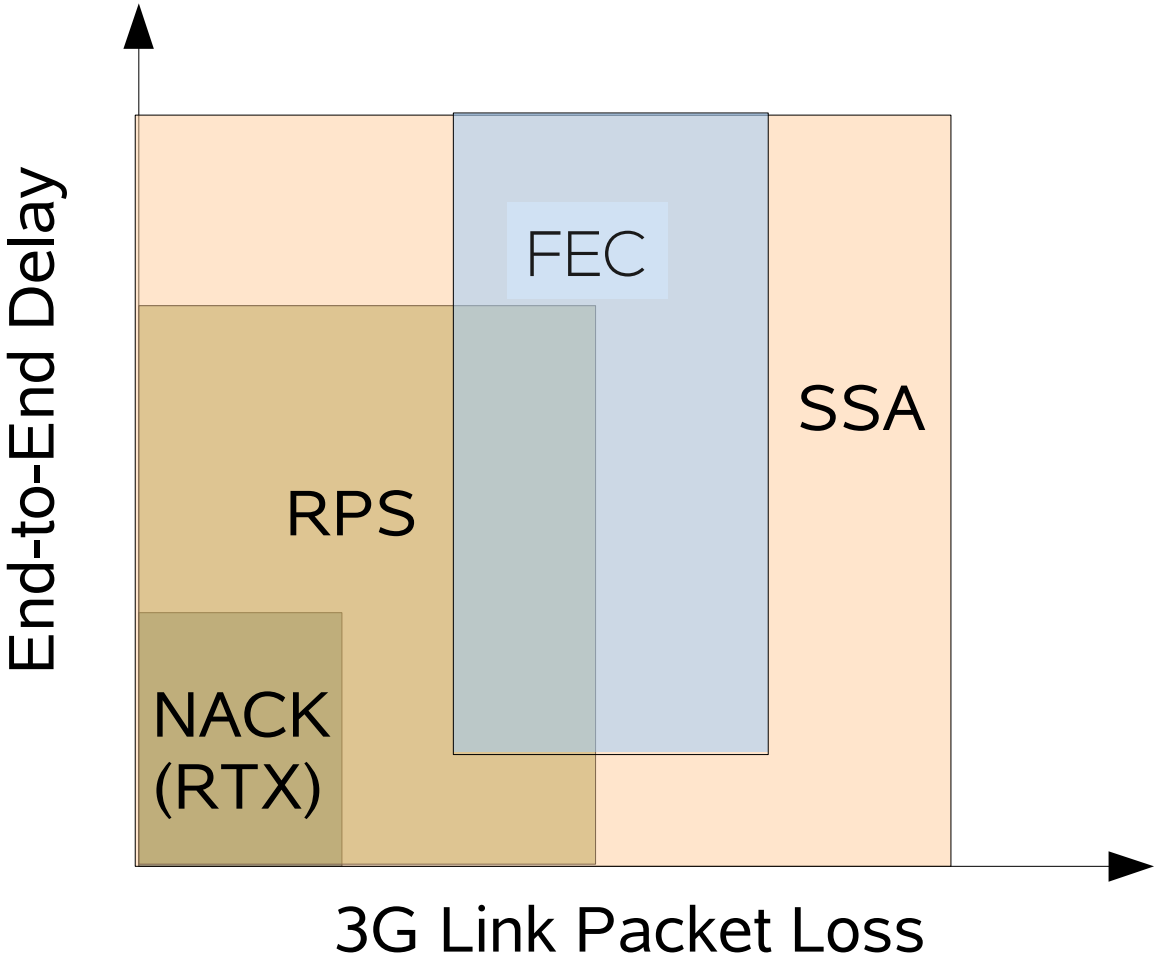
IETF 91, Honolulu,
09. November 2014

https://tools.ietf.org/html/draft-singh-rmcat-adaptive-fec-01

IPR

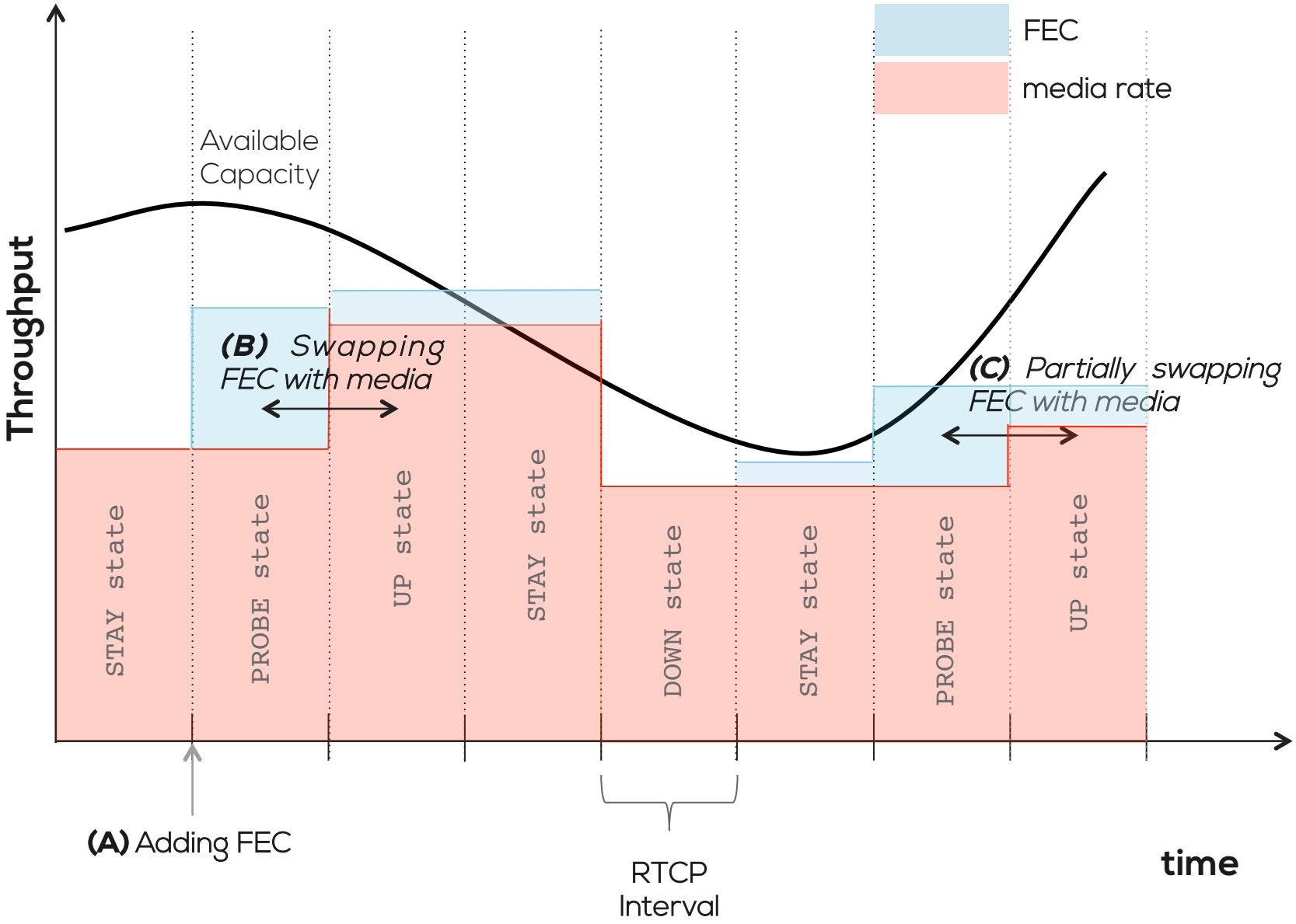
- 2 IPR disclosures
  - Nokia
  - Polycom

Error Resilience

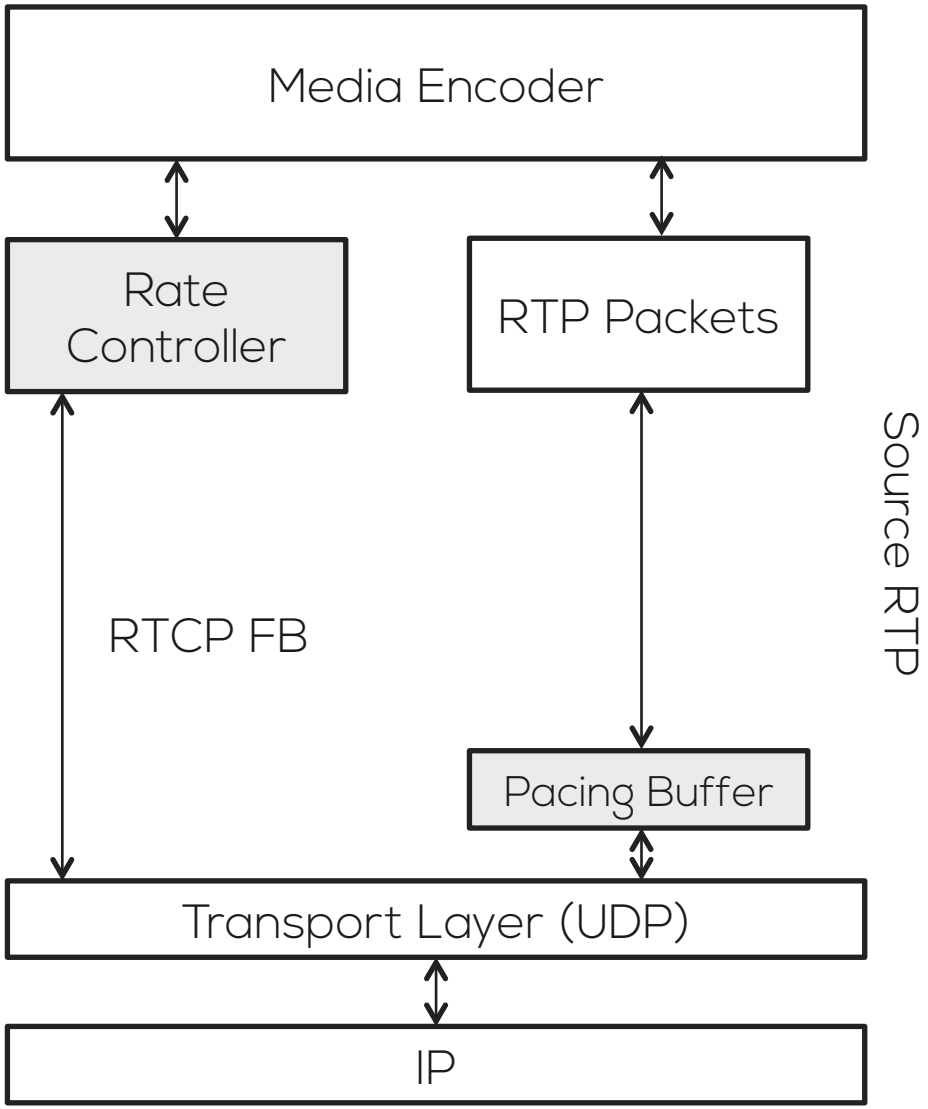


SSA: Adaptive packet sizes
RPS: reference picture selection

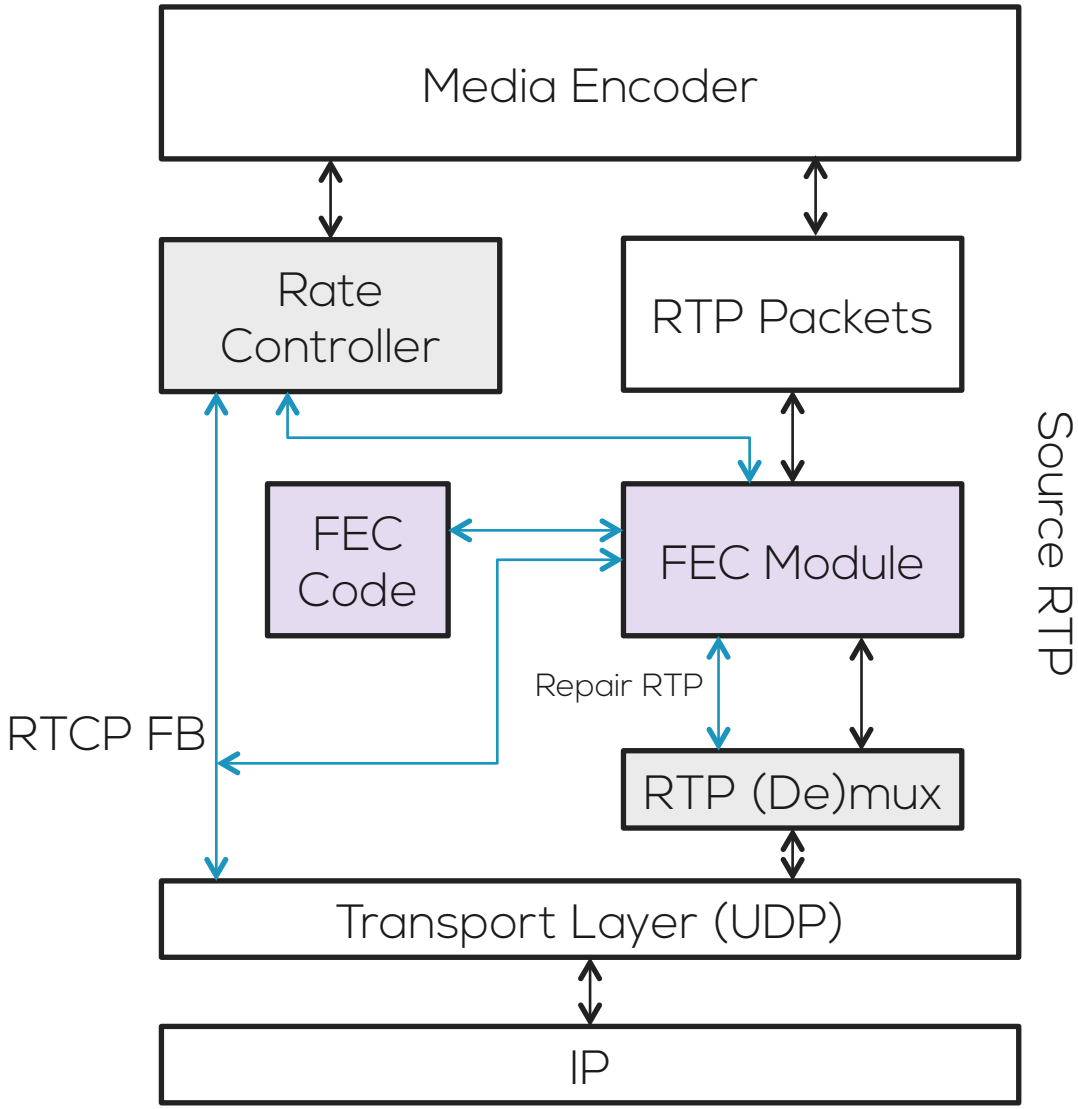
CONCEPT


FEC
media rate
Available Capacity
Throughput
(B) Swapping FEC with media
(C) Partially swapping FEC with media
STAY state
PROBE state
UP state
STAY state
DOWN state
STAY state
PROBE state
UP state
(A) Adding FEC
RTCP Interval
time

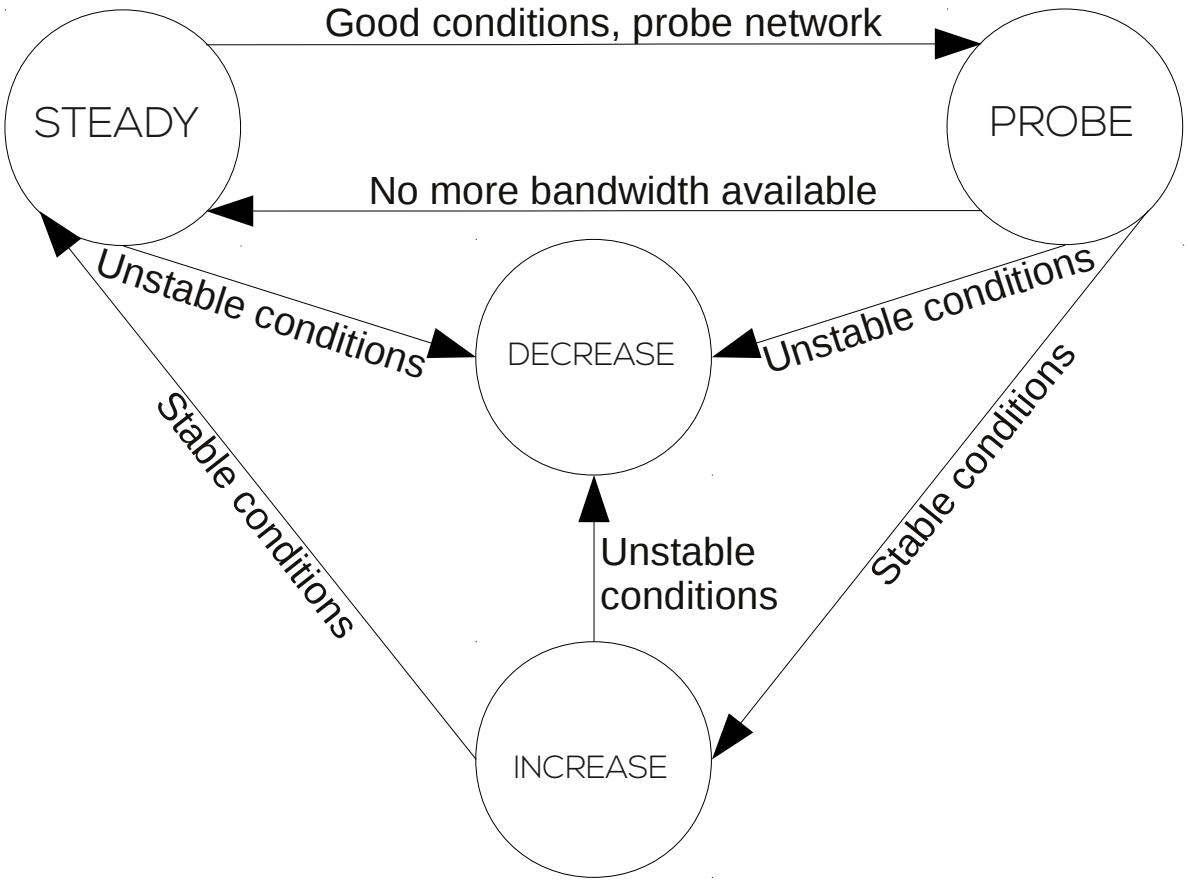
CC Framework



RFC 6363: FEC Framework



State Machine



FEC Scheme

- Currently, using
  - parity, 1- or 2-d interleaved XOR (for burst loss)
  - There is discussion in RTCWEB on FEC schemes.

- But,
  - It could be applied to other schemes

S, Holmer, et al., "**Handling Packet Loss in WebRTC**", Proc. of IEEE International Conference on Image Processing (ICIP 2013) , 9 2013.

RTCP Feedback

- RLE of Post-repair (RFC5725)
- RLE of loss packets (RFC3611)
- RLE of discarded packets (RFC7097)

- Packet count of lost and repaired packets
  - (draft-ietf-xrblock-post-loss-repair)

Calculating goodput

- Receiver reports goodput:
  - Goodbytes in reporting interval =
    bytes received – bytes discarded

OR

- Sender calculates goodput:
  - Goodbytes in reporting interval =
    bytes sent – bytes lost – bytes discarded

Undershoot

- Congestion reported
  - Sender calculates duration of congestion =
    HSN when RTCP scheduled – HSN when congestion detected
  - undershoot bytes = sending rate * duration/8
  - New rate = Goodbytes – undershoot bytes [1]

OR

  - Delta = Sending rate – goodput
  - New rate = sending rate – 2 x delta [2]

[1] http://www.netlab.tkk.fi/~jo/papers/2012-videv-rate-control.pdf
[2] http://www.netlab.tkk.fi/~varun/nagy2014mmsys.pdf

Typical results

- ~90% recovery in time for decoding when FEC interval was short
  - Fewer packets protected
- ~80% recovery in time for decoding when FEC interval was long
  - More packets protected

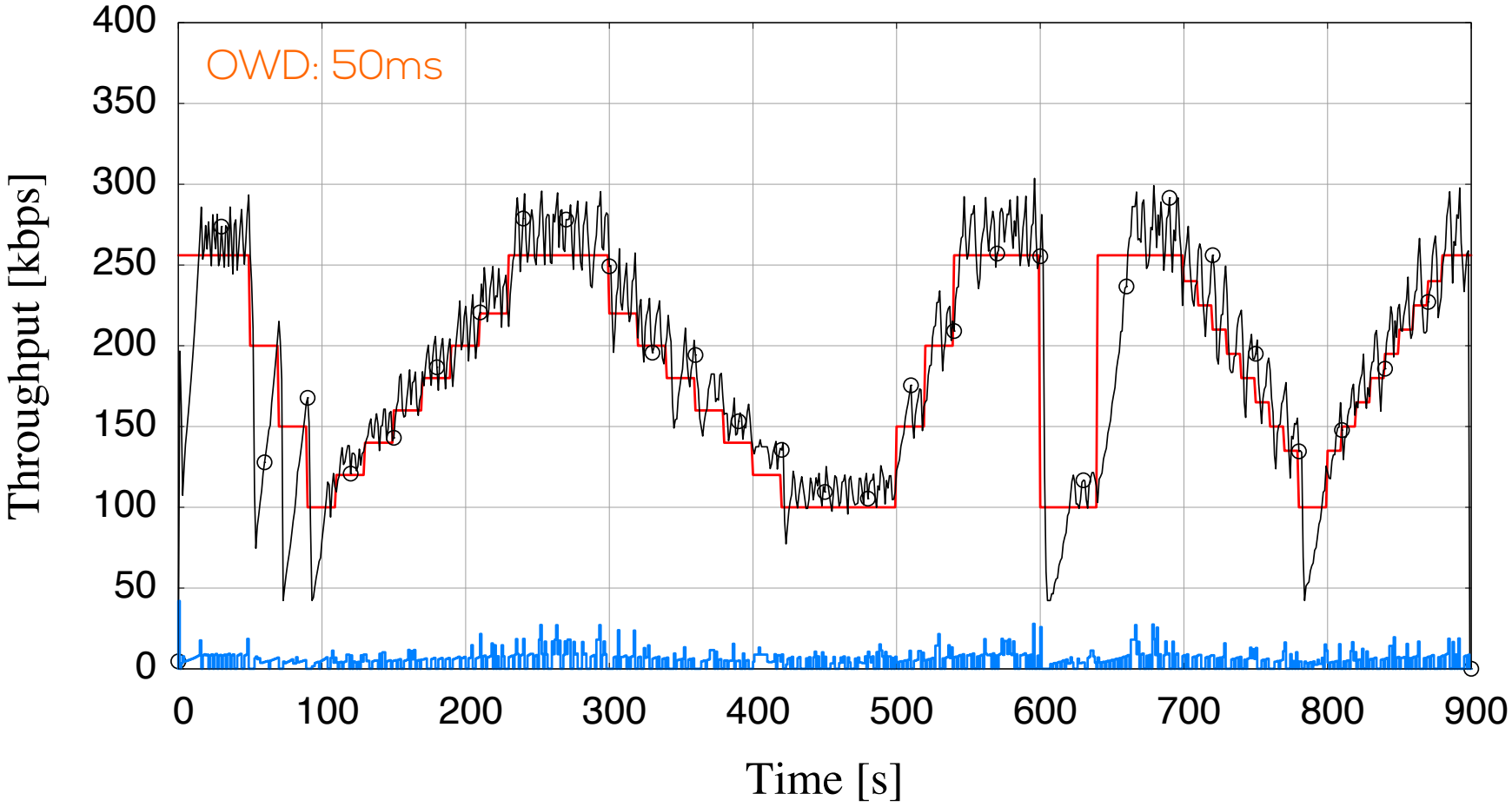
Applicability

- Implemented over a delay-based congestion control
  - See paper for details

- However would like to generalize it.
  - Apply to SCReAM, GCC, …

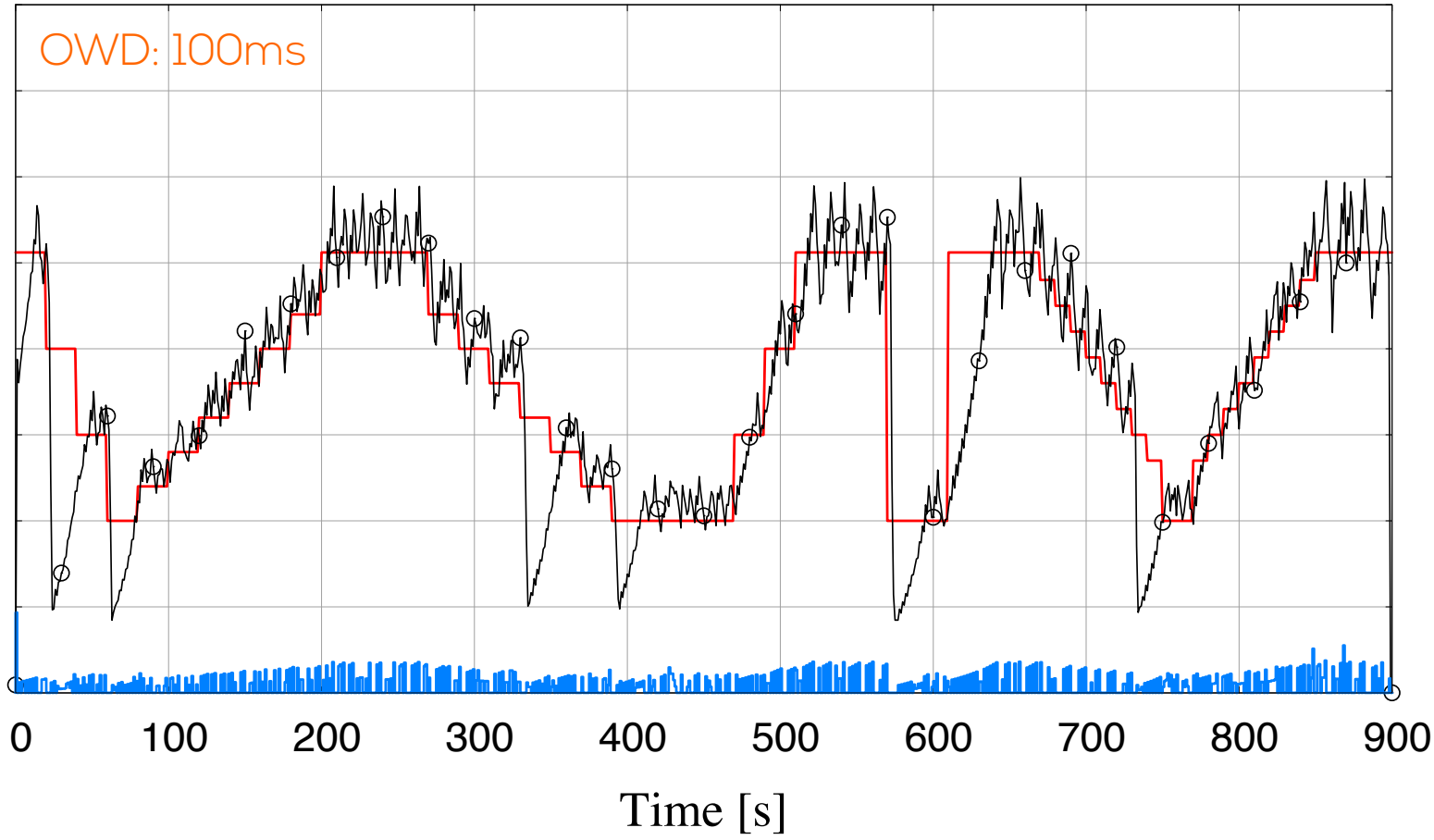
Nagy M., Singh V., Ott J., Eggert L., **Congestion Control using FEC for Conversational Multimedia Communication**, Proc. of ACM Multimedia Systems, Singapore, SG, Mar, 2014,

Evaluation (1/3)

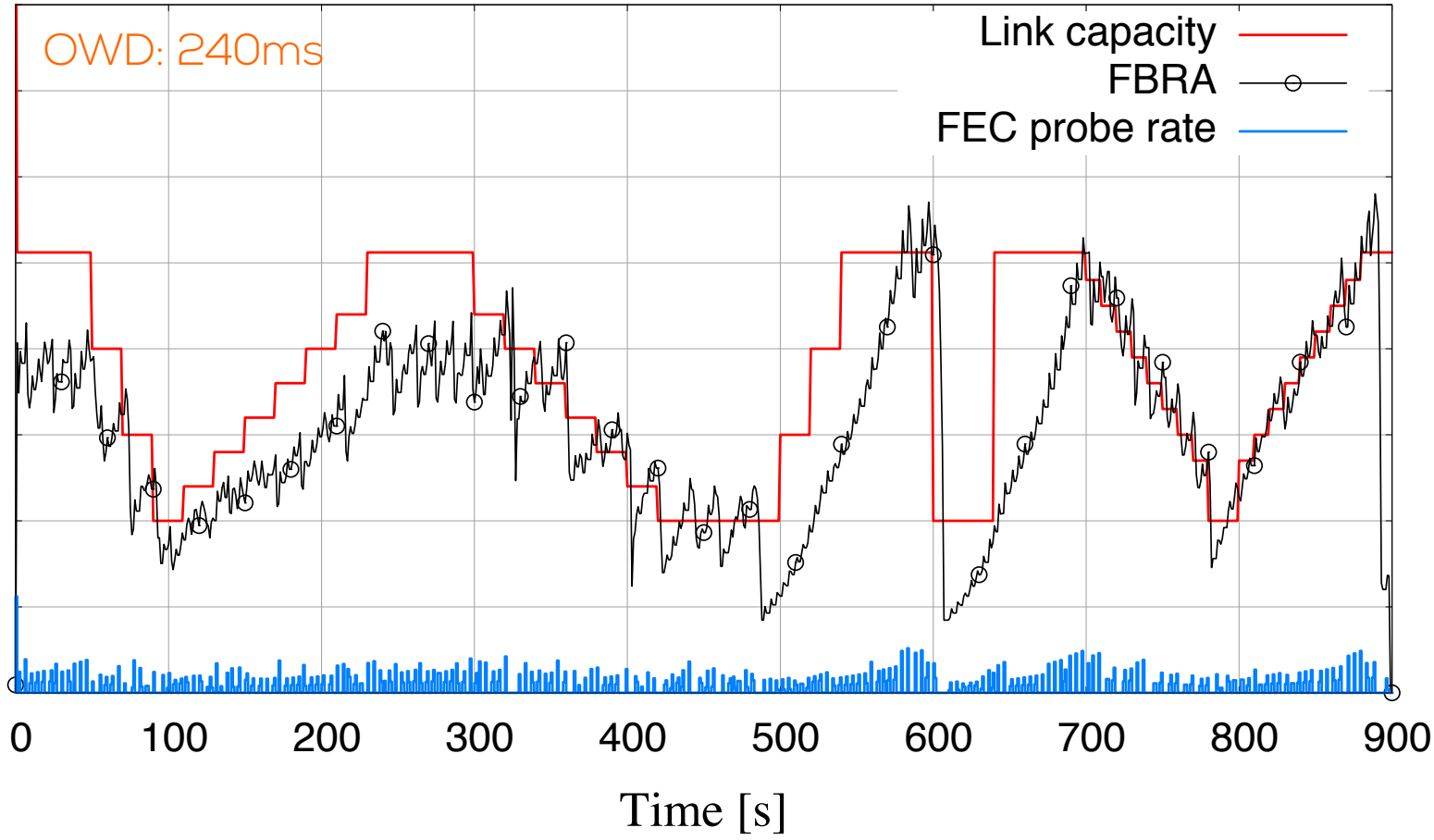


- Ns-2 simulation, Variable link capacity, Single flow on the link

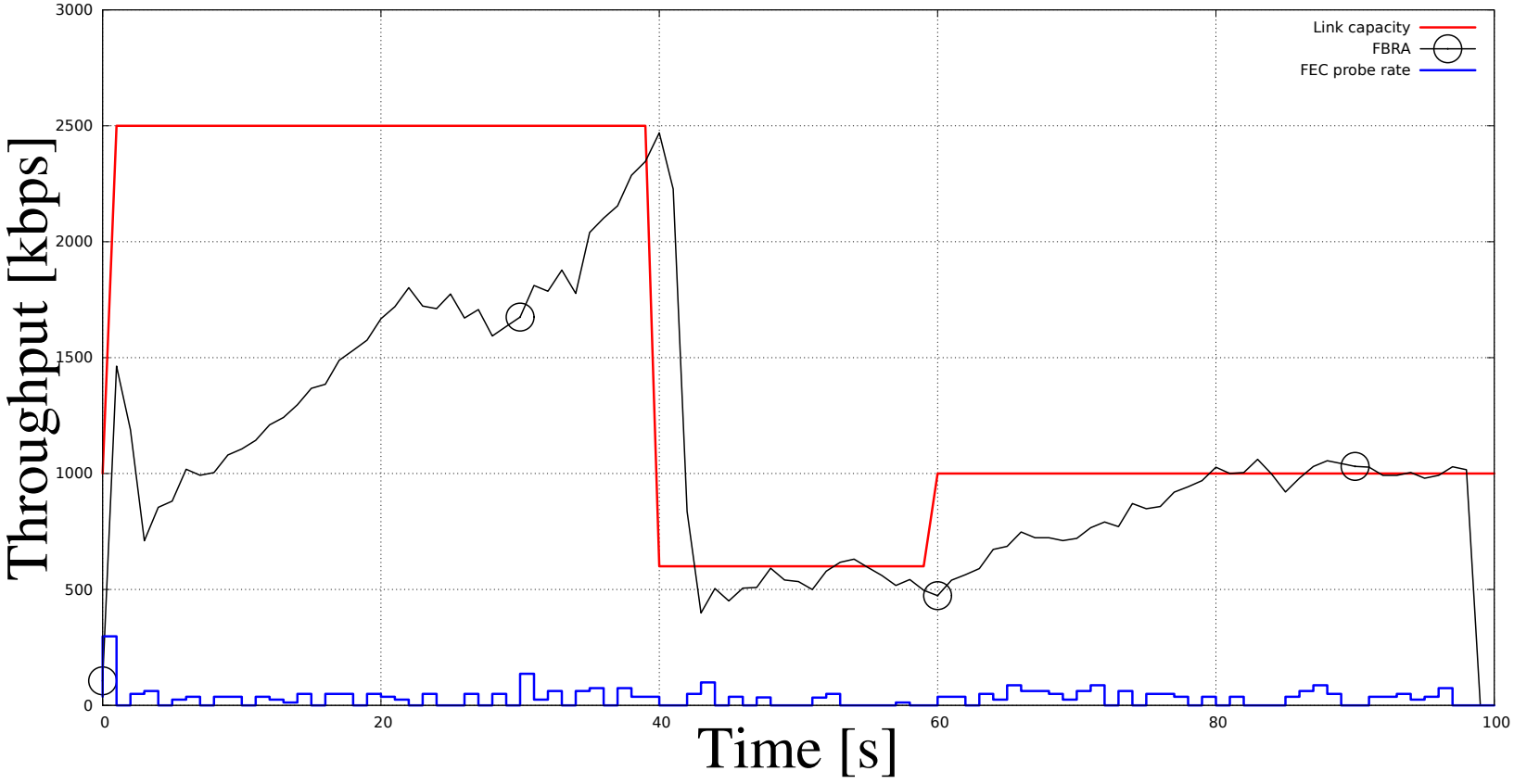
Evaluation (2/3)



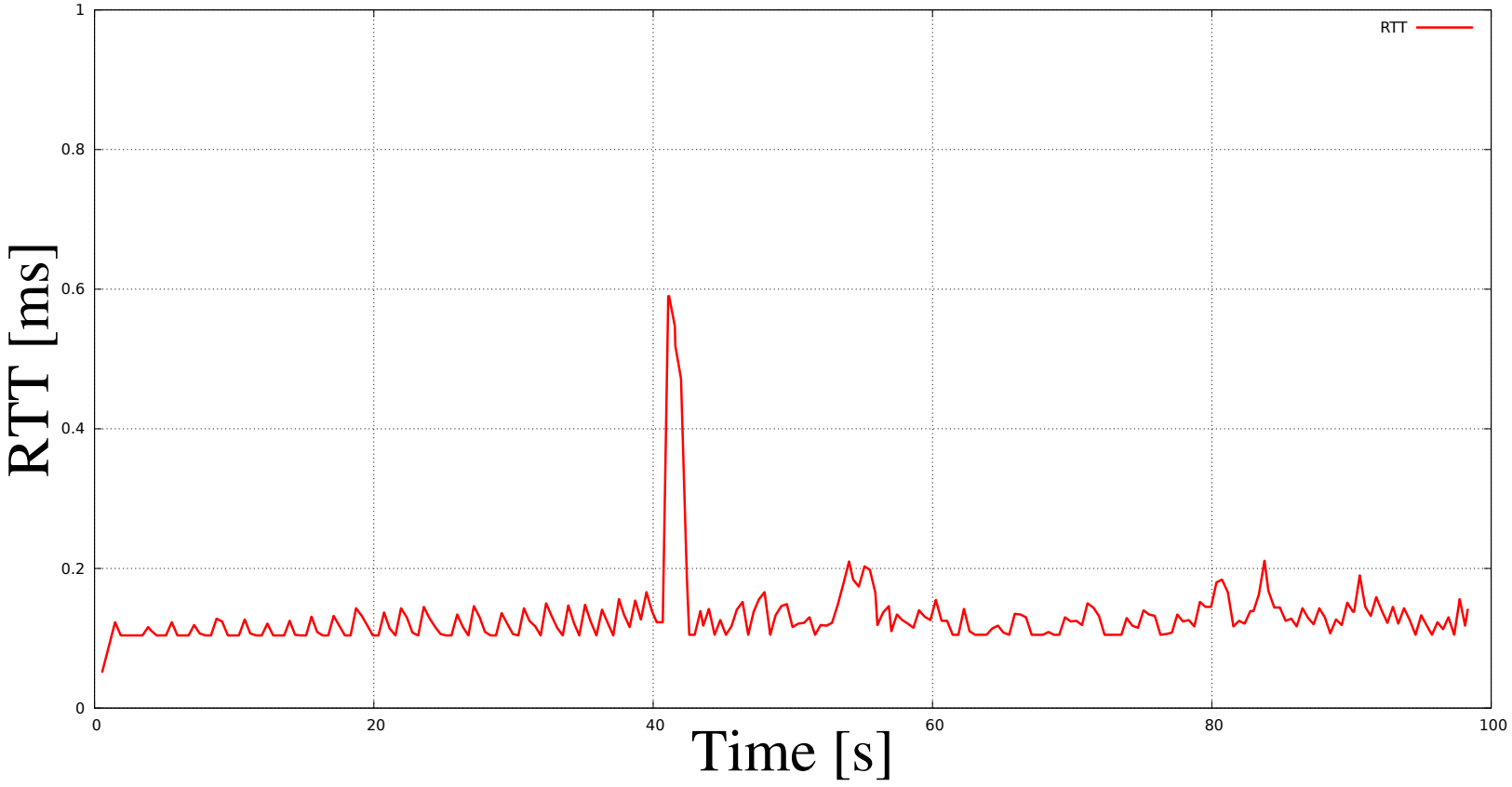
Evaluation (3/3)



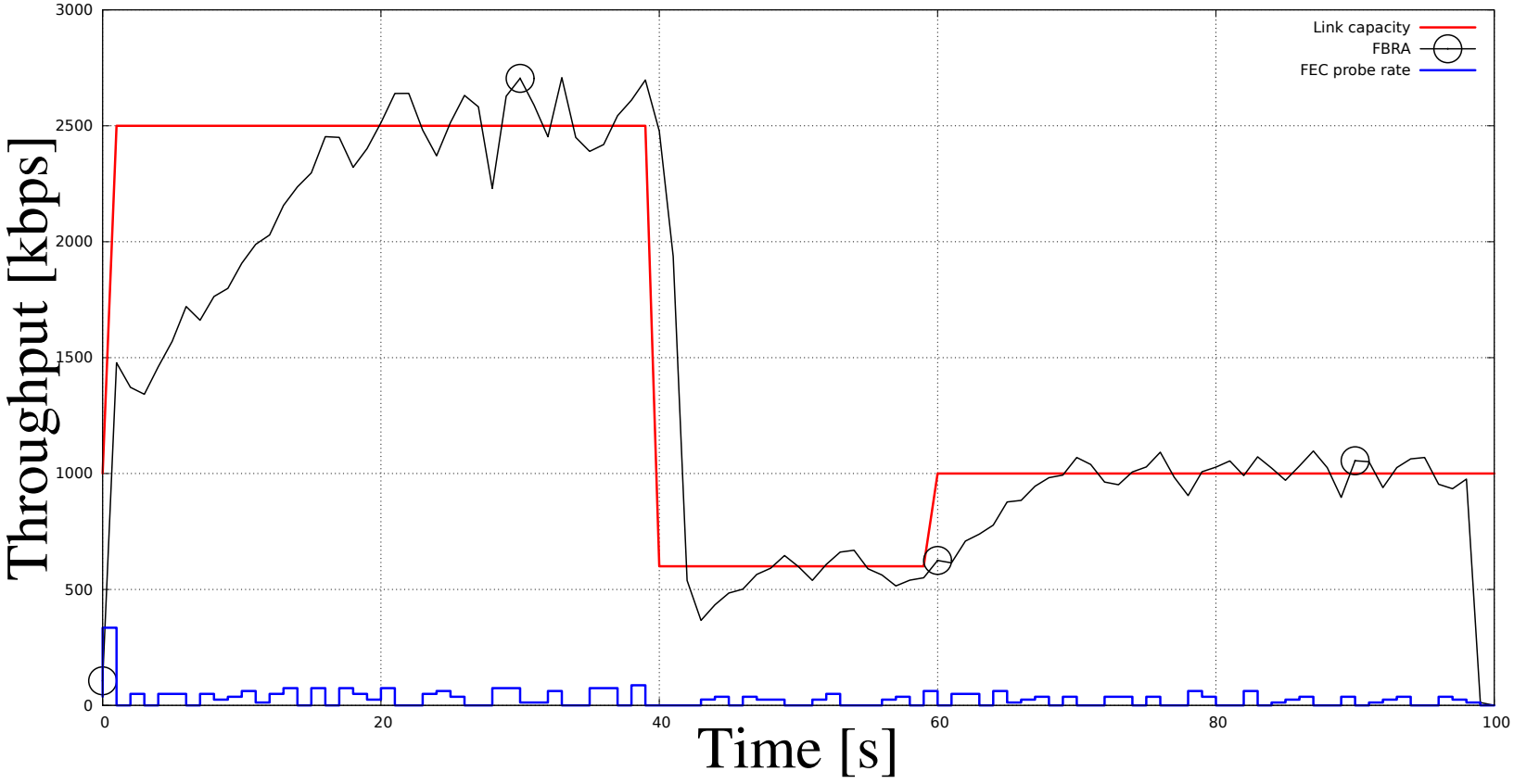
50ms


Throughput [kbps]
Time [s]
Link capacity
FBRA
FEC probe rate
0
500
1000
1500
2000
2500
3000
0
20
40
60
80
100

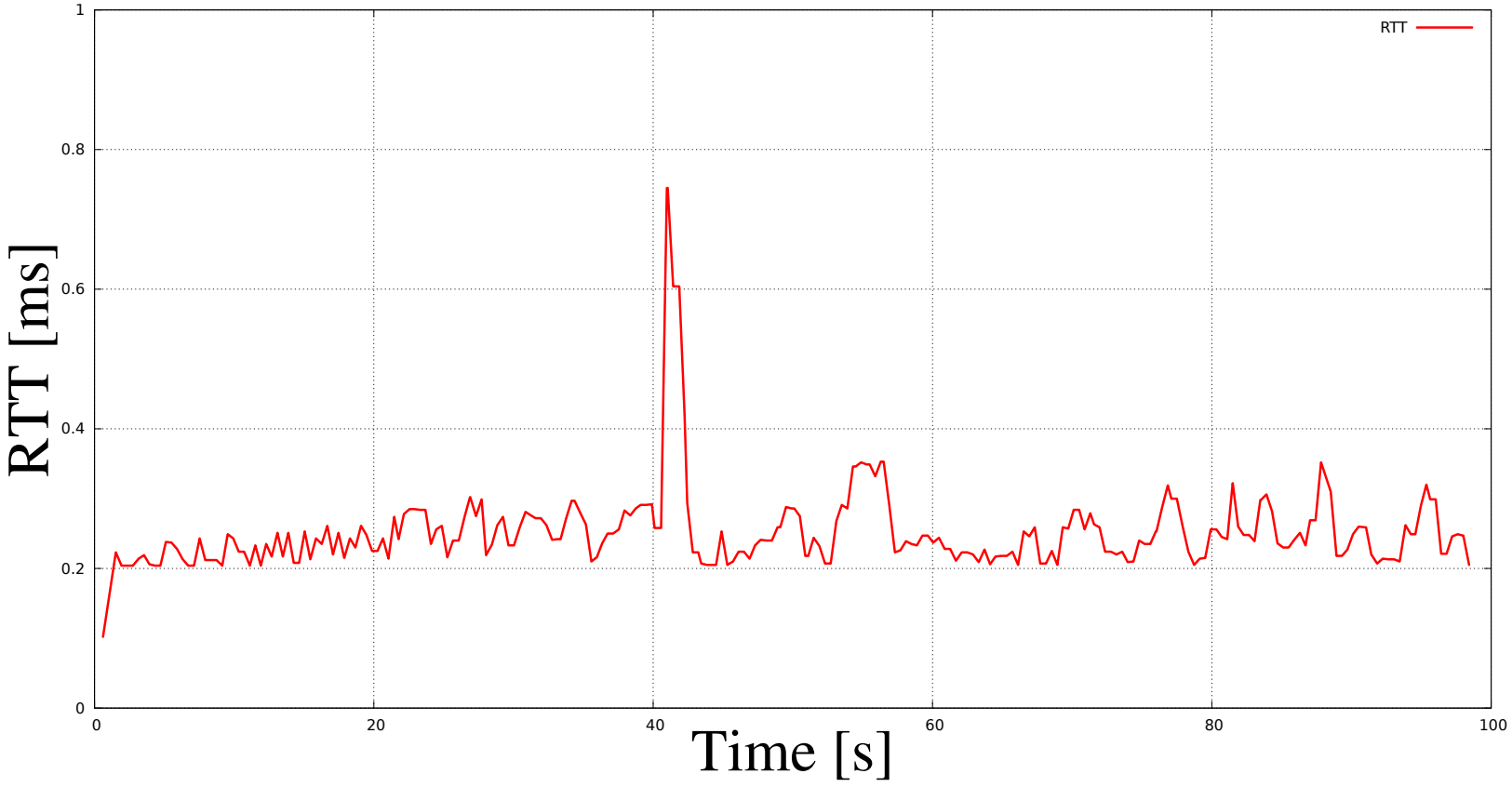
50ms



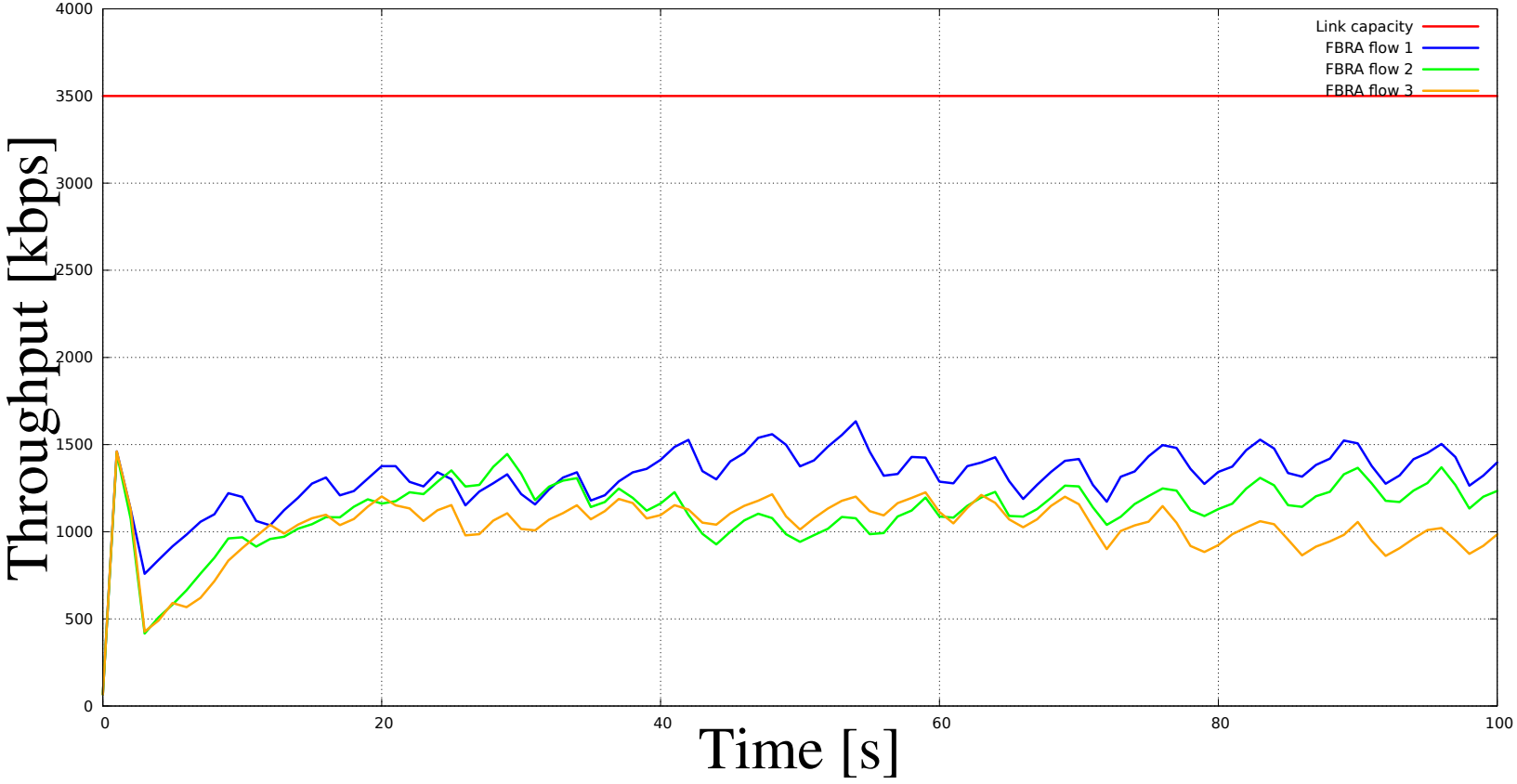
100ms


Throughput [kbps]
Time [s]
Link capacity
FBRA
FEC probe rate
0
500
1000
1500
2000
2500
3000
0
20
40
60
80
100

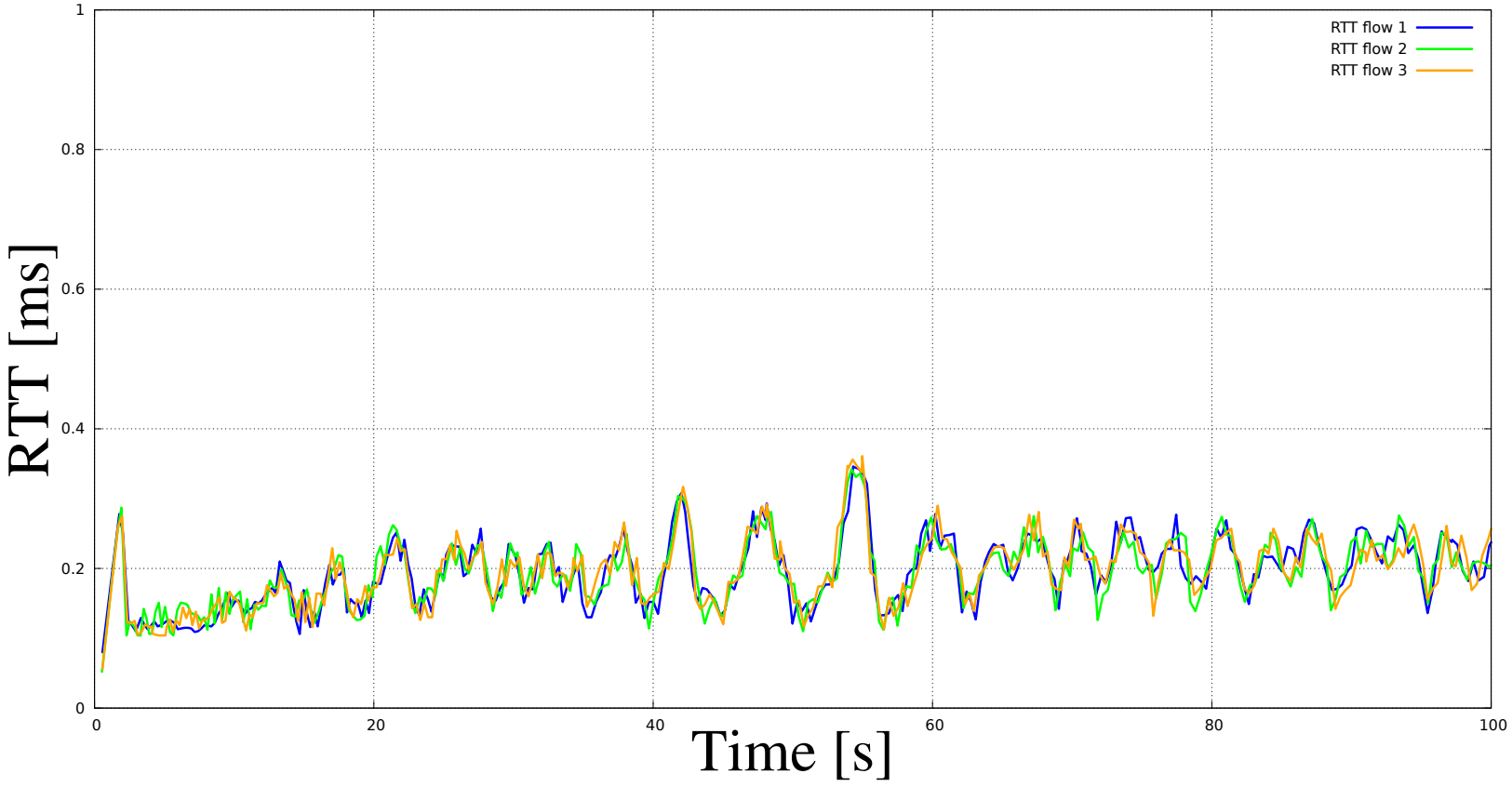
100ms



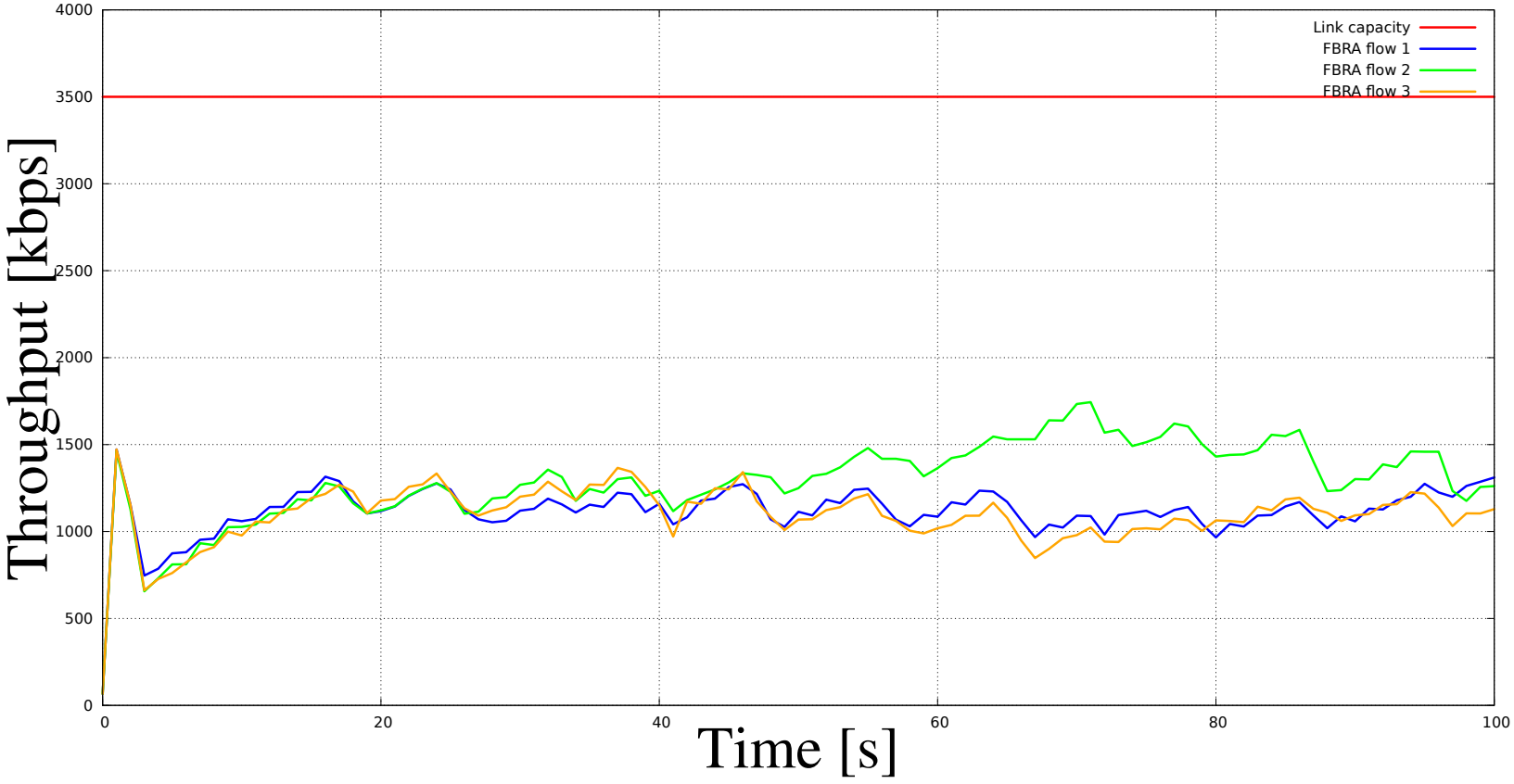
RTT fairness (50ms)



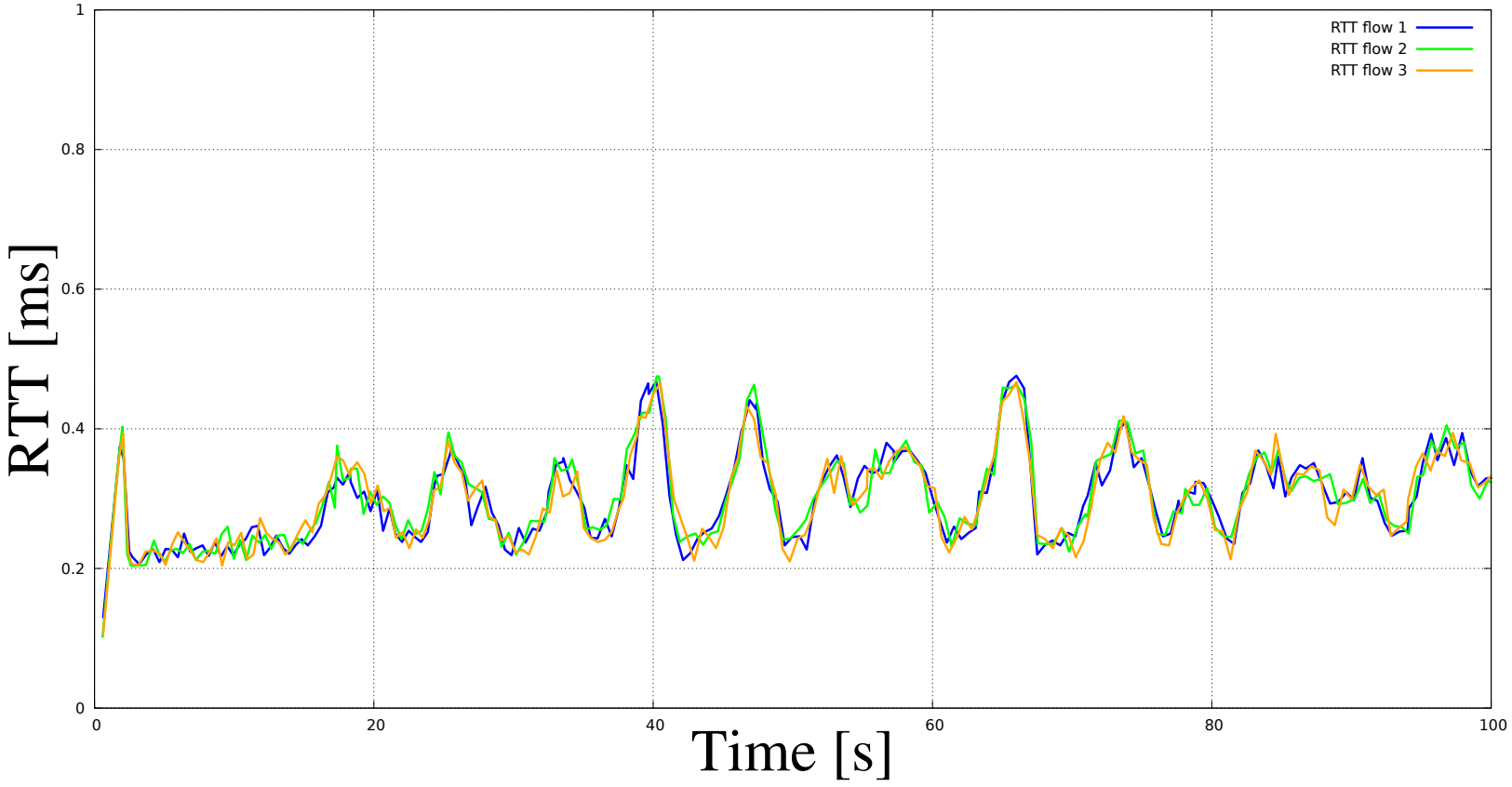
5.4: RTT fairness



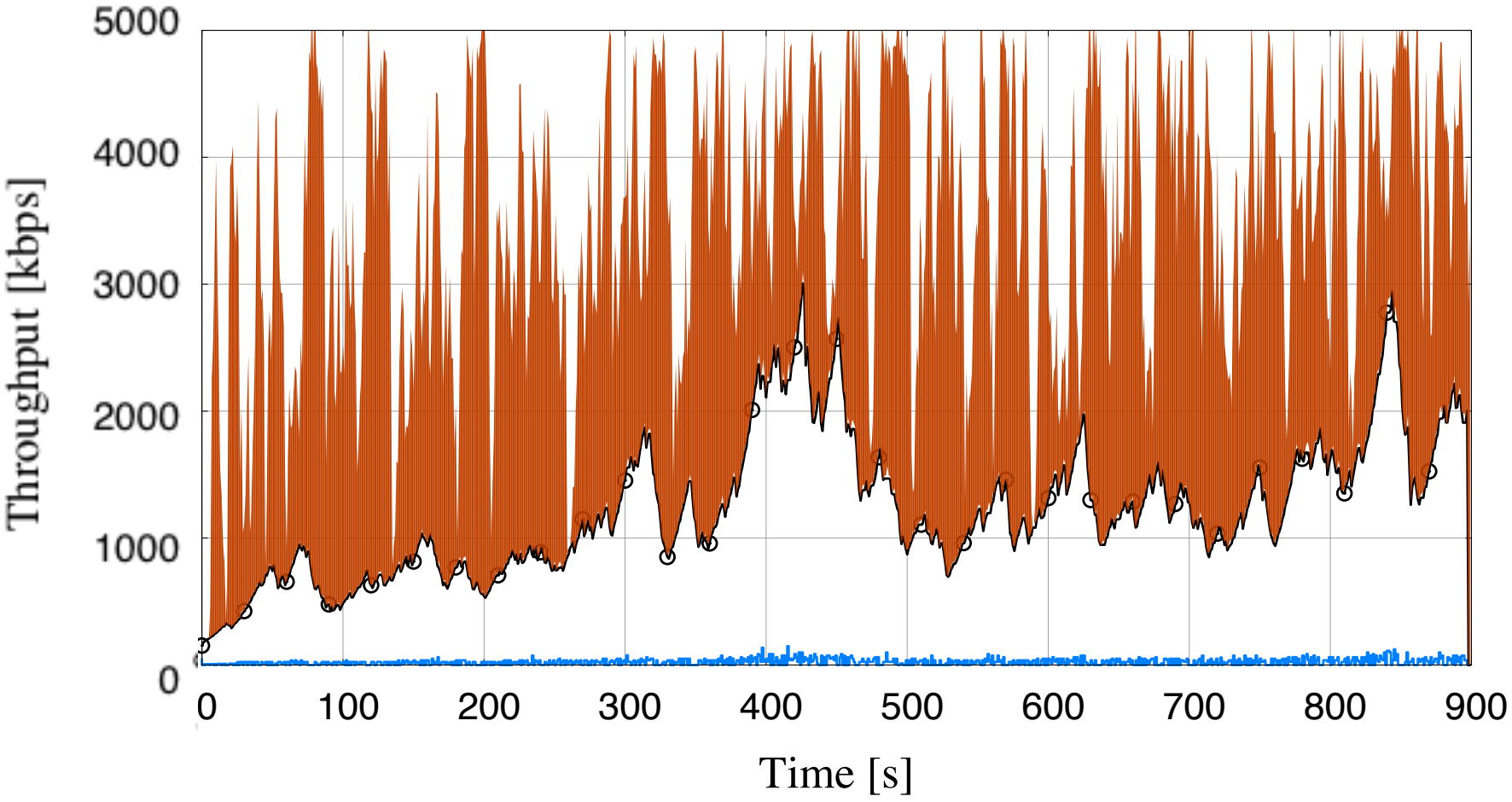
RTT fairness (100ms)



RTT fairness (100ms)

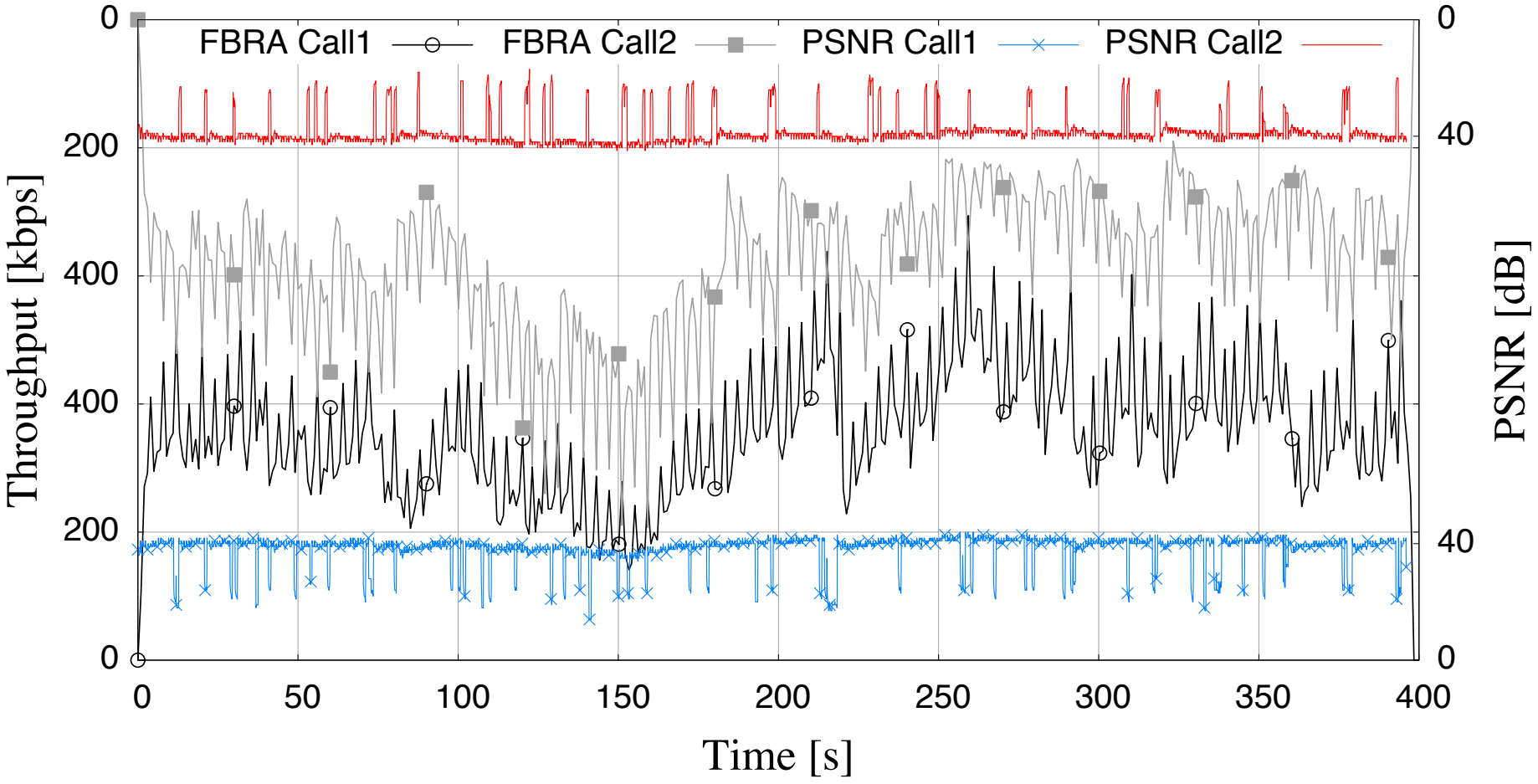


Compete with short TCPs



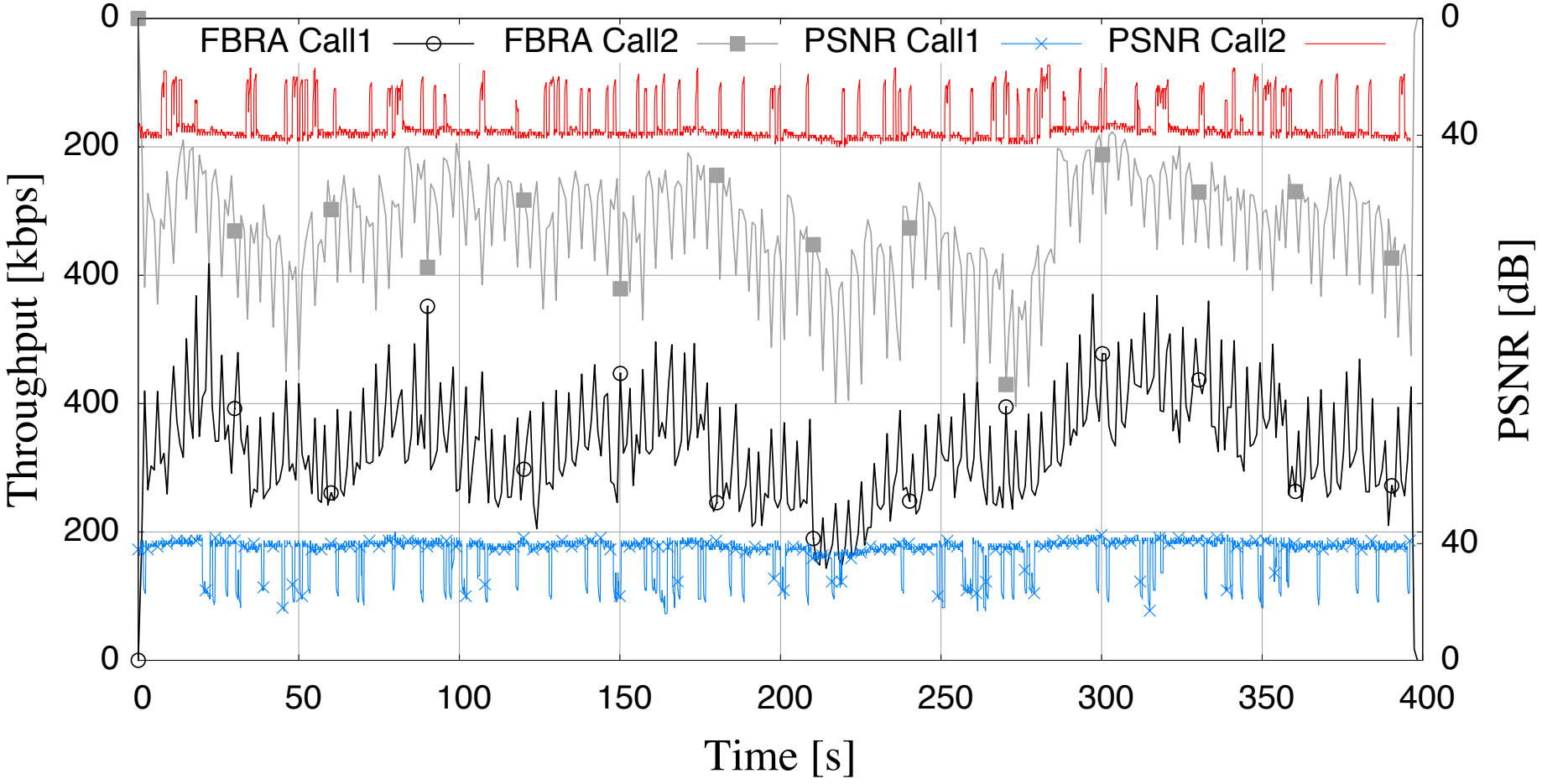
TESTBED Evaluation (1/2)

Two RTP flows on the link



- 1Mbit/s link capacity, 50ms one-way-delay,

TESTBED Evaluation (2/2)



- 1Mbit/s link capacity, 100ms one-way-delay,

Next steps

- Code: (coming soon)
  - https://github.com/protocols-comnet/rmcat-adaptive-fec-code
- Evaluation Paper:
  - Nagy M., Singh V., Ott J., Eggert L., Congestion Control using FEC for Conversational Multimedia Communication, Proc. of ACM Multimedia Systems, Singapore, SG, Mar, 2014,
- More feedback is appreciated ☺