HTTP Adaptive Streaming over MPTCP

L. Montalvo, S. Gouache

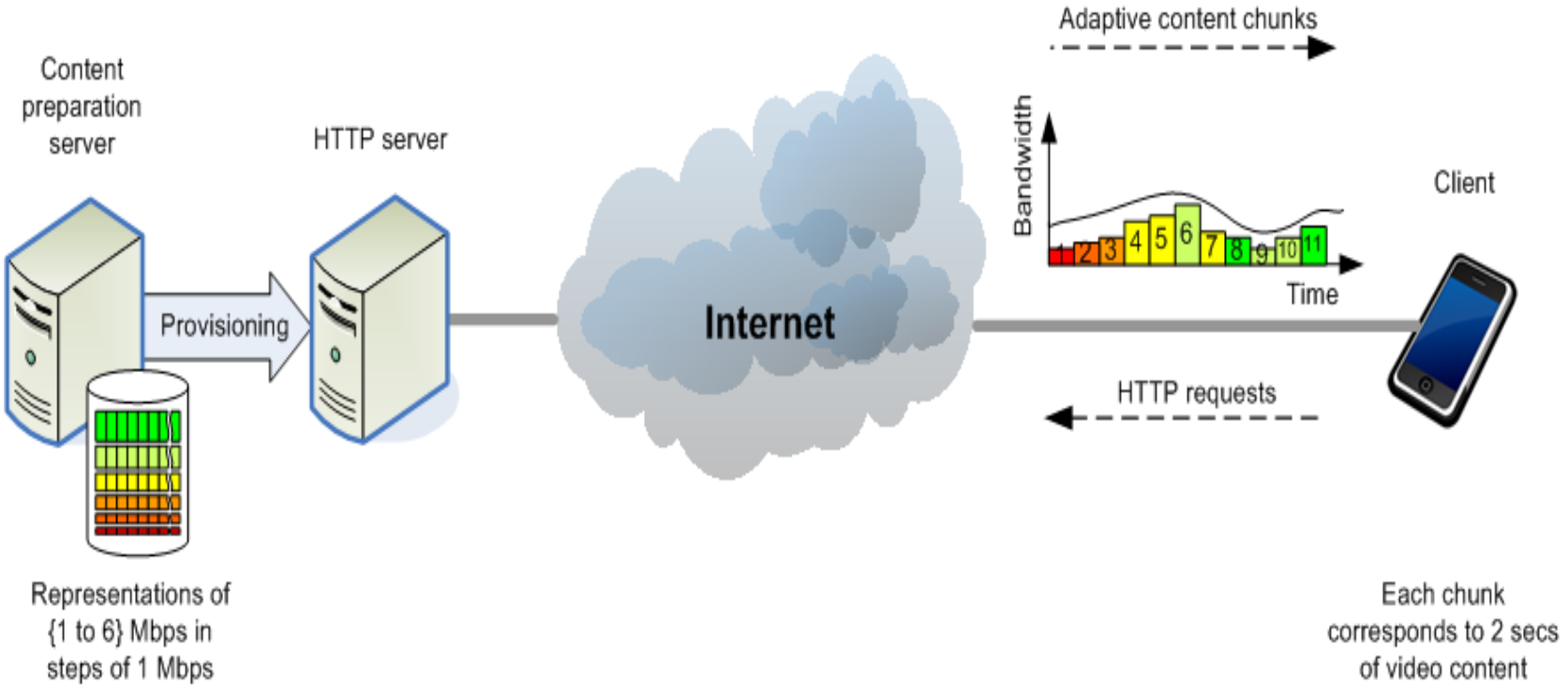
September 06, 2012

technicolor

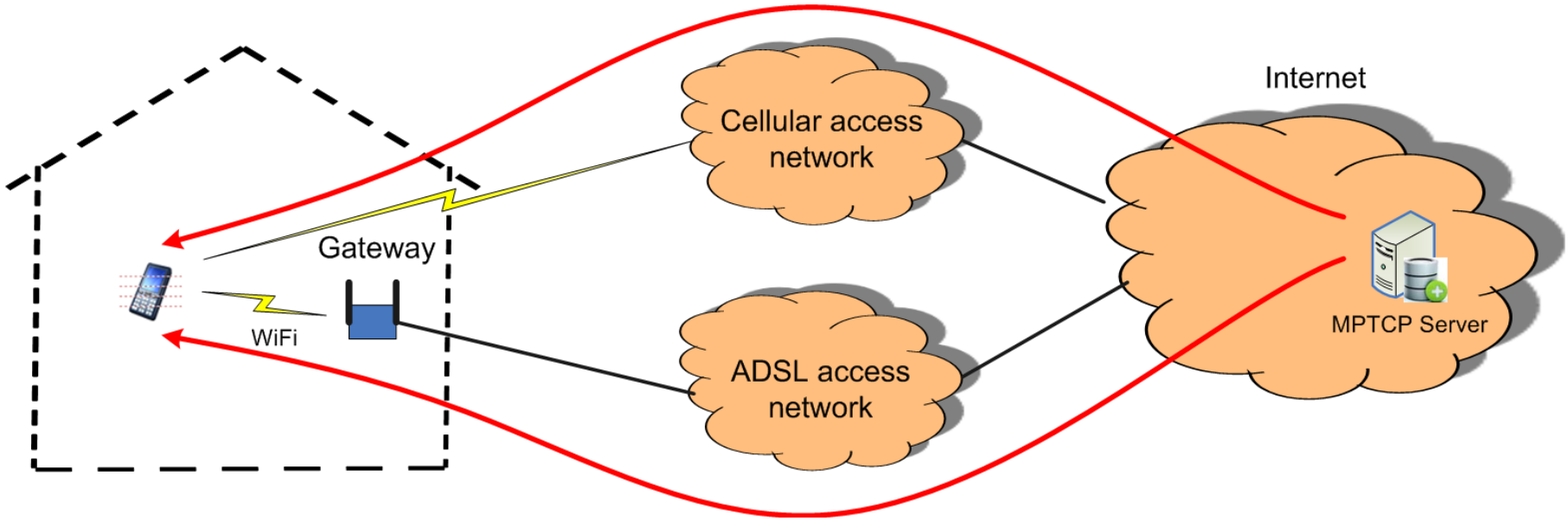
Outline

- Background: HTTP Adaptive Streaming, MPTCP.
- Our testbed
- Channel emulation
- Experimental results
- Conclusions

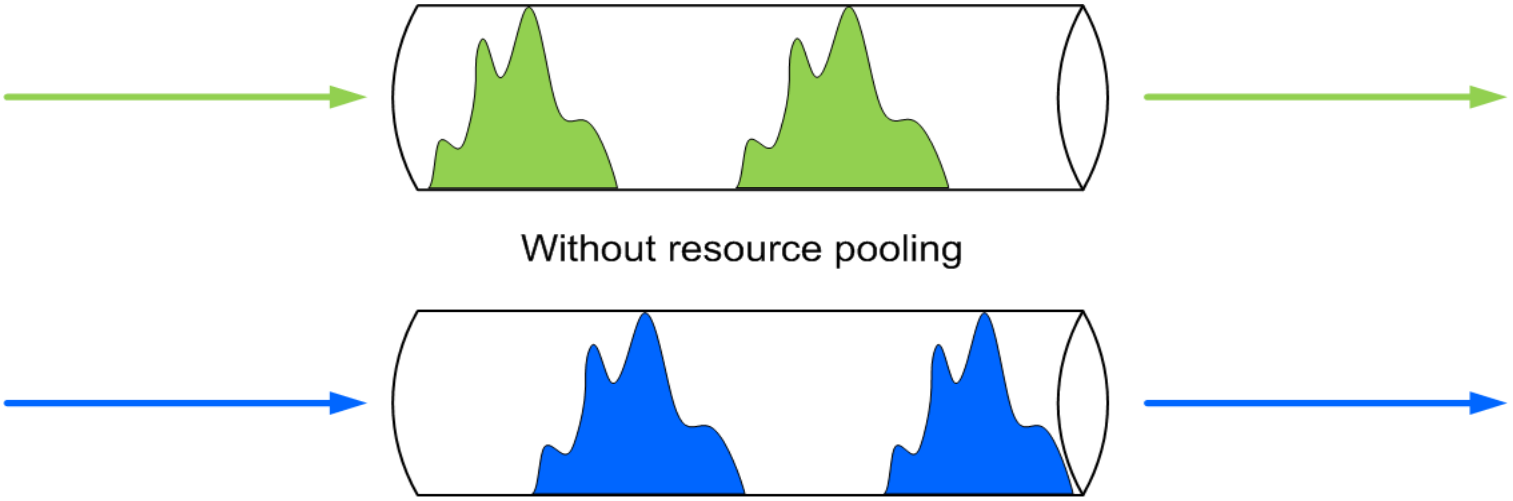
HTTP adaptive streaming

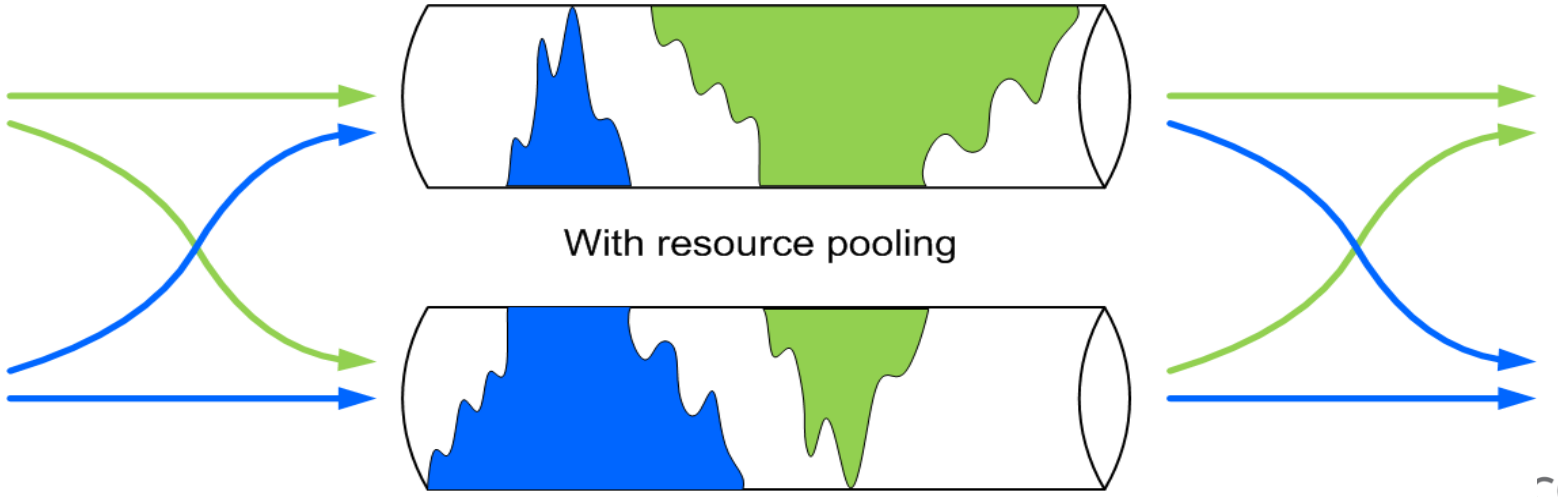


Observation: path diversity

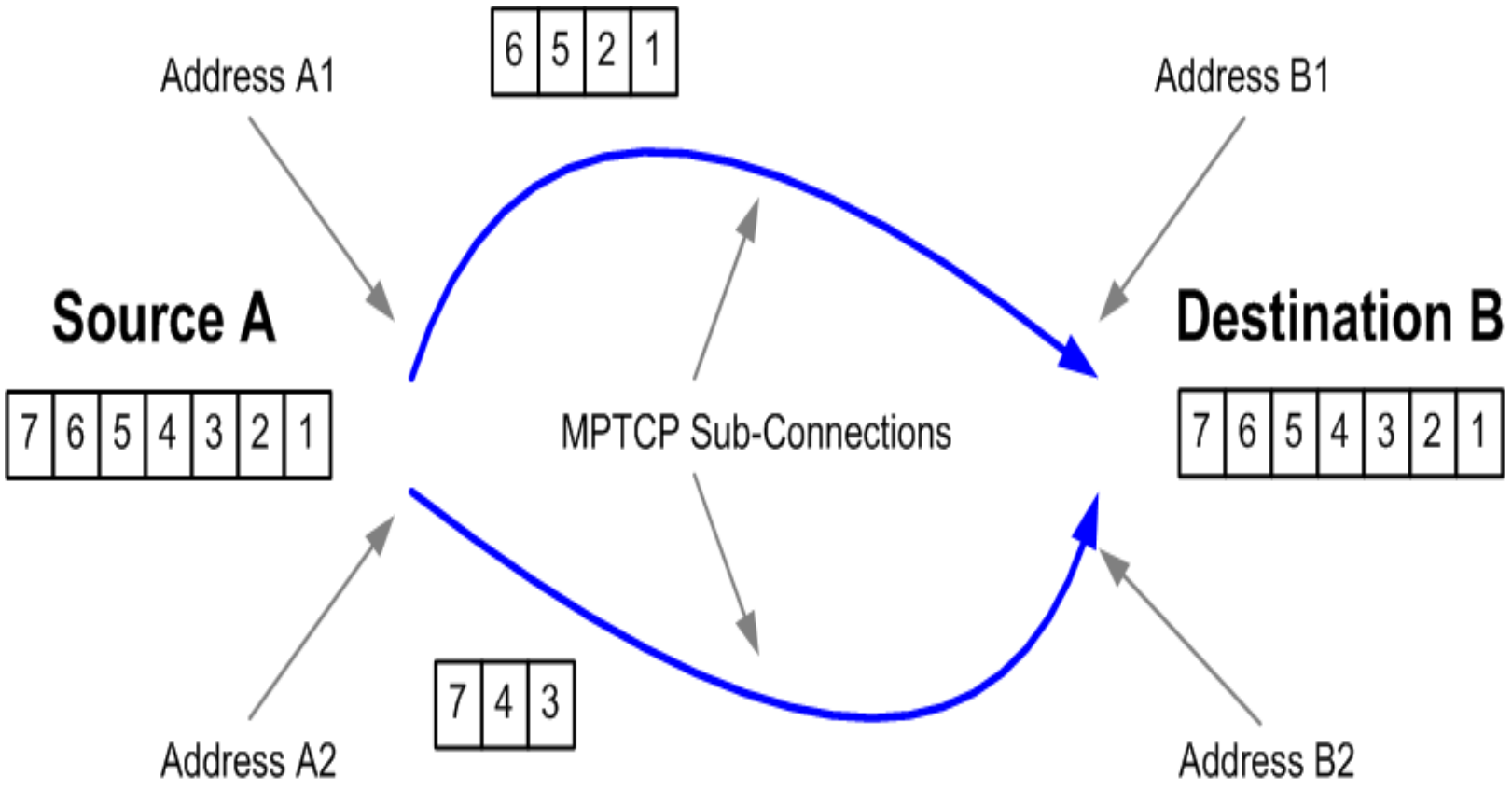


Resource pooling principle





Resource pooling applied to transport layer (MPTCP)



MPTCP design constraints

- It must be backwards-compatible with current, regular TCP, to increase its chances of deployment.
- It can be assumed that one or both hosts are multihomed and multiaddressed.

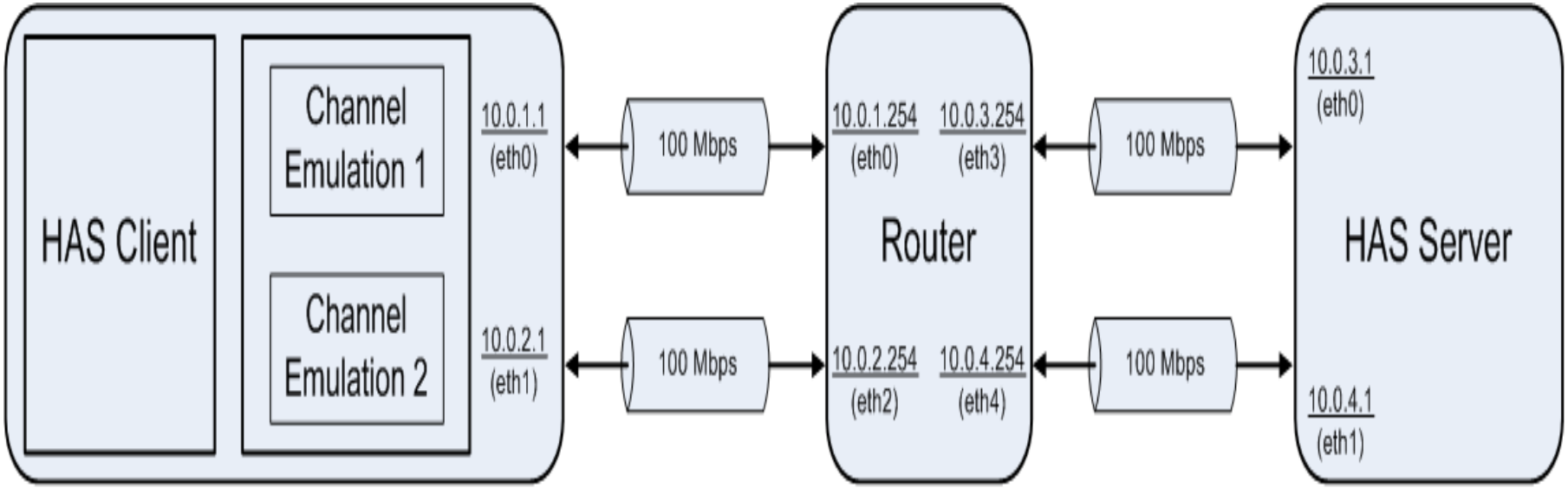
MPTCP backwards compatibility 8

- External Constraints: The protocol must function through the vast majority of existing middle-boxes such as NATs, firewalls and proxies.

- Application Constraints: The protocol must be usable with no change to existing applications that use the standard TCP API. The protocol must provide the same service model as regular TCP to the application.

- Fall-back: The protocol should be able to fall back to standard TCP with no interference from the user, to be able to communicate with legacy hosts.

Our testbed



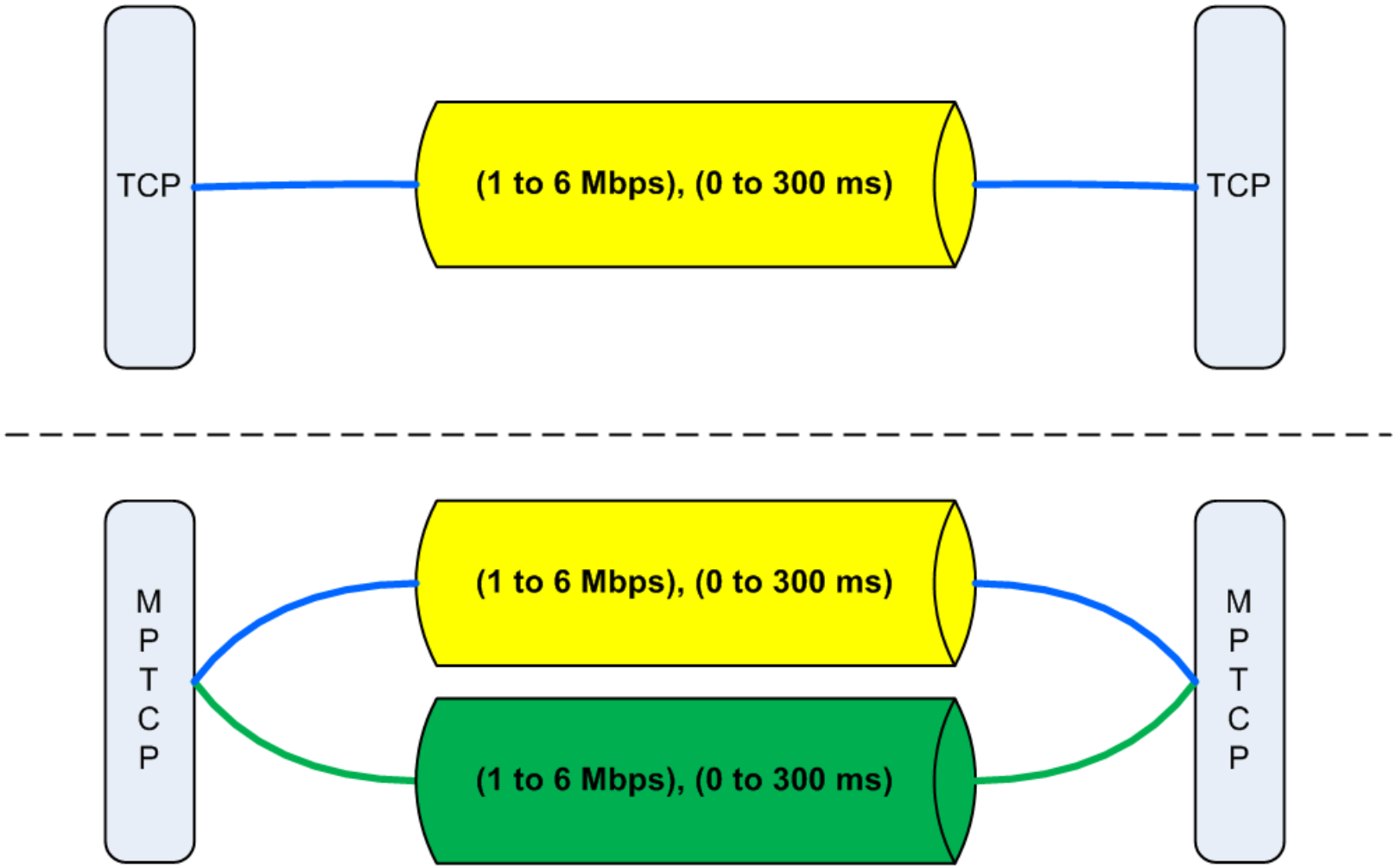
For MPTCP tests:
all three workstations run a
3.2.0 Linux Kernel, patched with
MPTCP stack from UCL (Belgium).

For TCP tests:
all three workstations run a
3.2.0 Linux Kernel, with CUBIC
congestion control.

Bandwidth prediction (Client)

- A software agent continuously measures the bandwidth available at the client's link.
- Based on the current and past values of the measured bandwidth, the client predicts the bandwidth available for the next chunk.
- The client chooses the next chunk from the most suitable representation for the predicted bandwidth.

Channel emulation



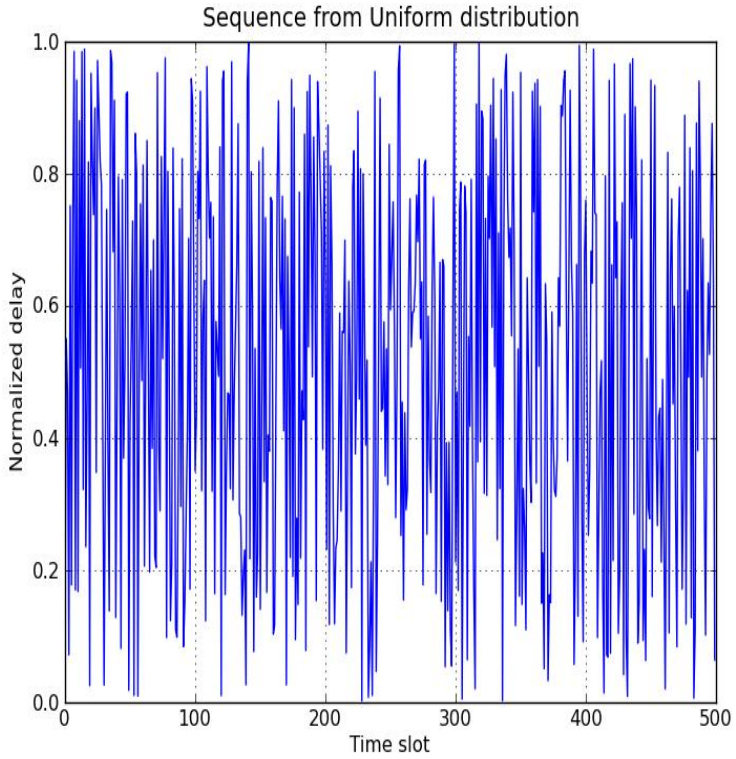
Channel emulation (bit-rate)

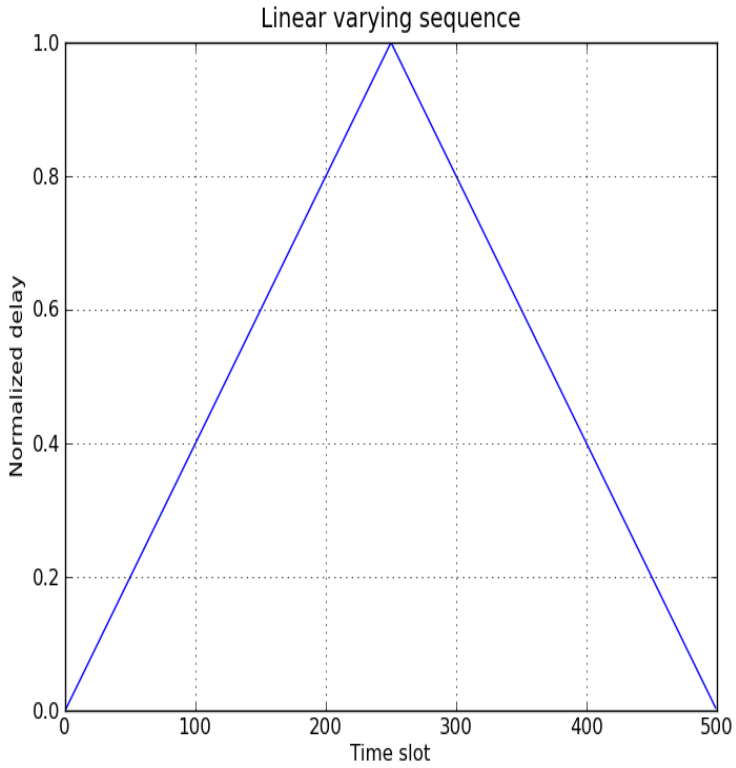
- S = {1 … 6} Mbps, step 0.25Mbps. Bitrates set.

$$C = \binom{n}{2} = \frac{n(n-1)}{2} = \frac{24x23}{2} = 276\,\text{combs.}$$

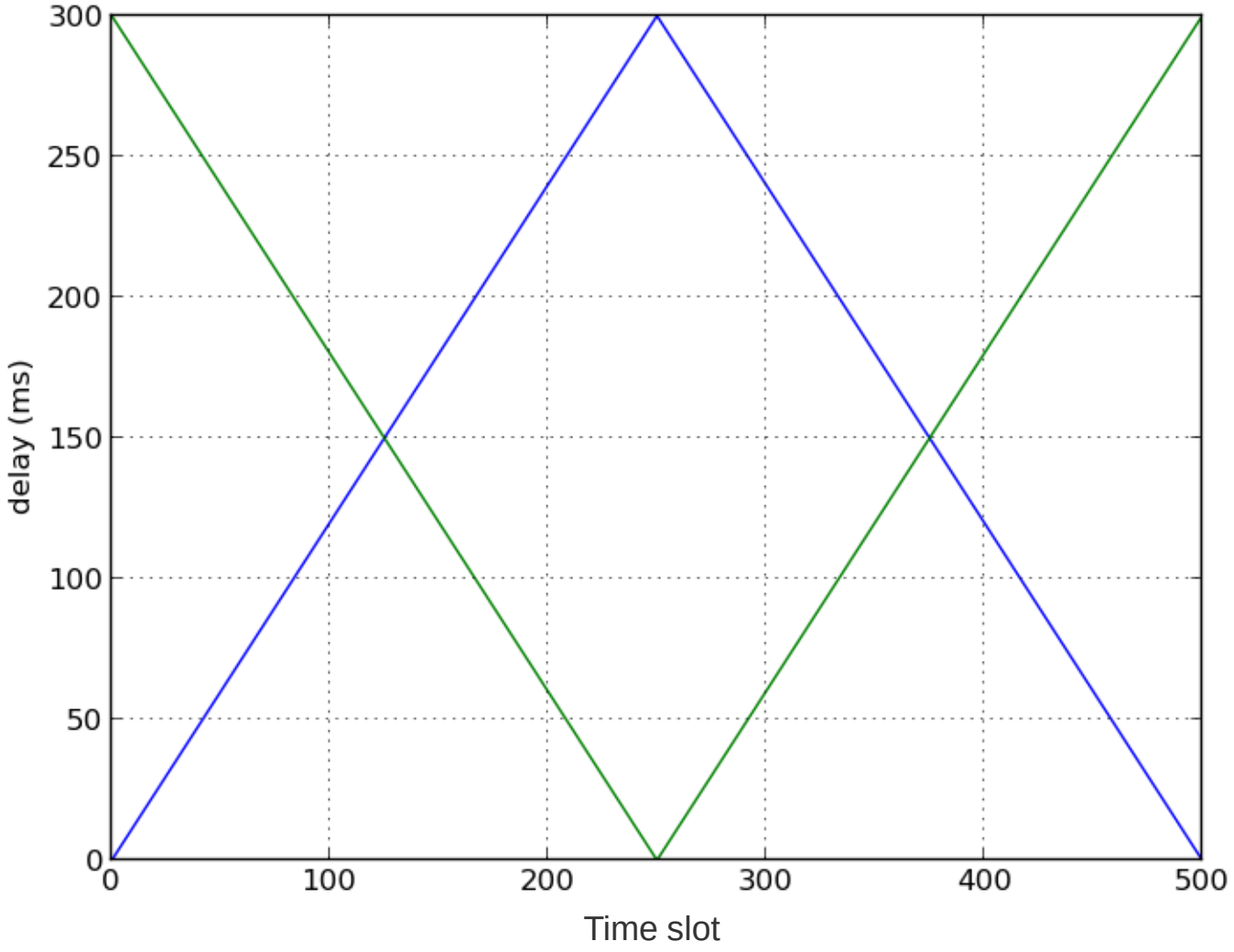
- Each bit-rate combination runs during 5 mins.
  - test duration = 23 hrs. (for one channel type).
- For each bit-rate combination the delay changes every 150 ms. (time slot emulation).

Channel emulation (delay)

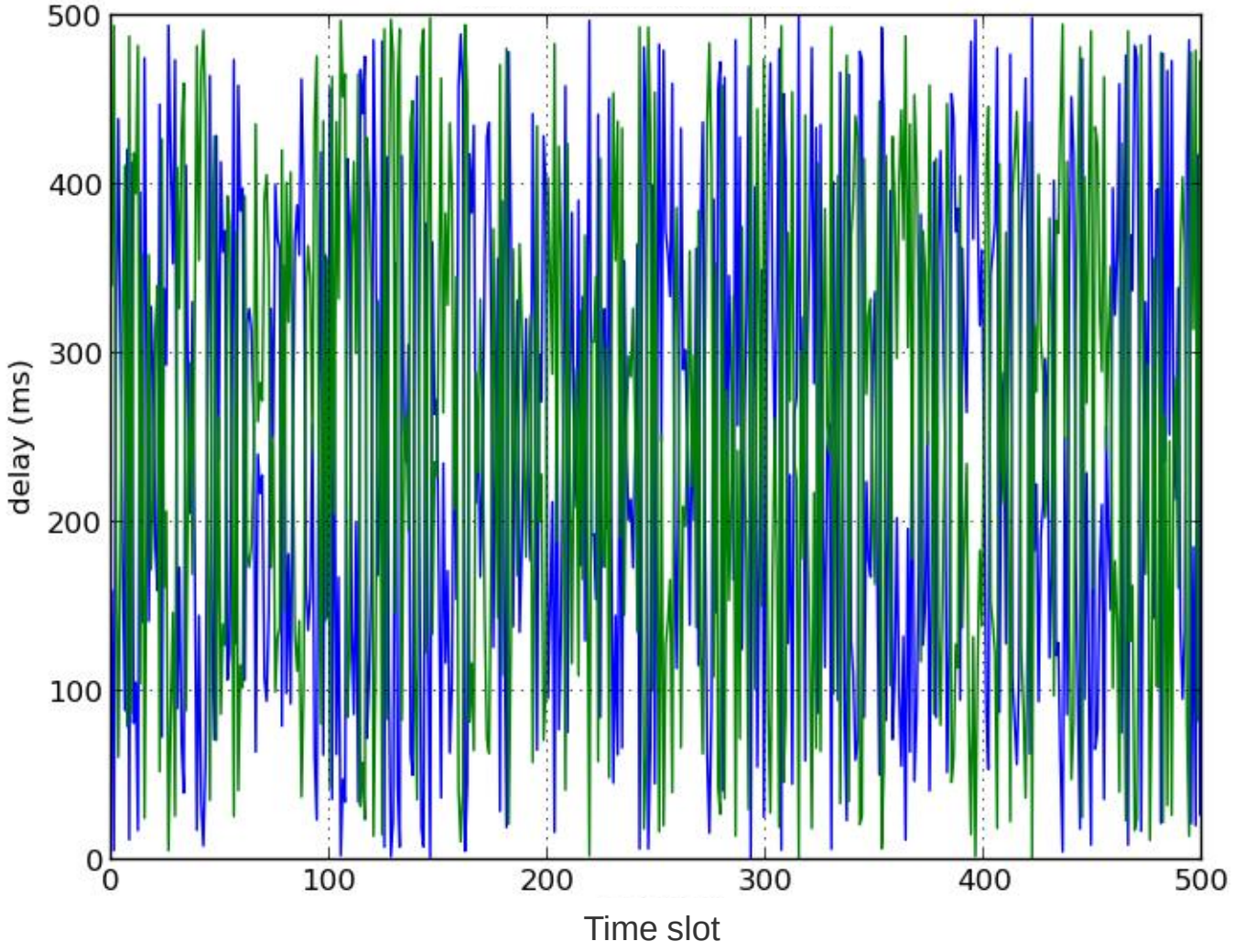




Correlated channels emulation (linear)14

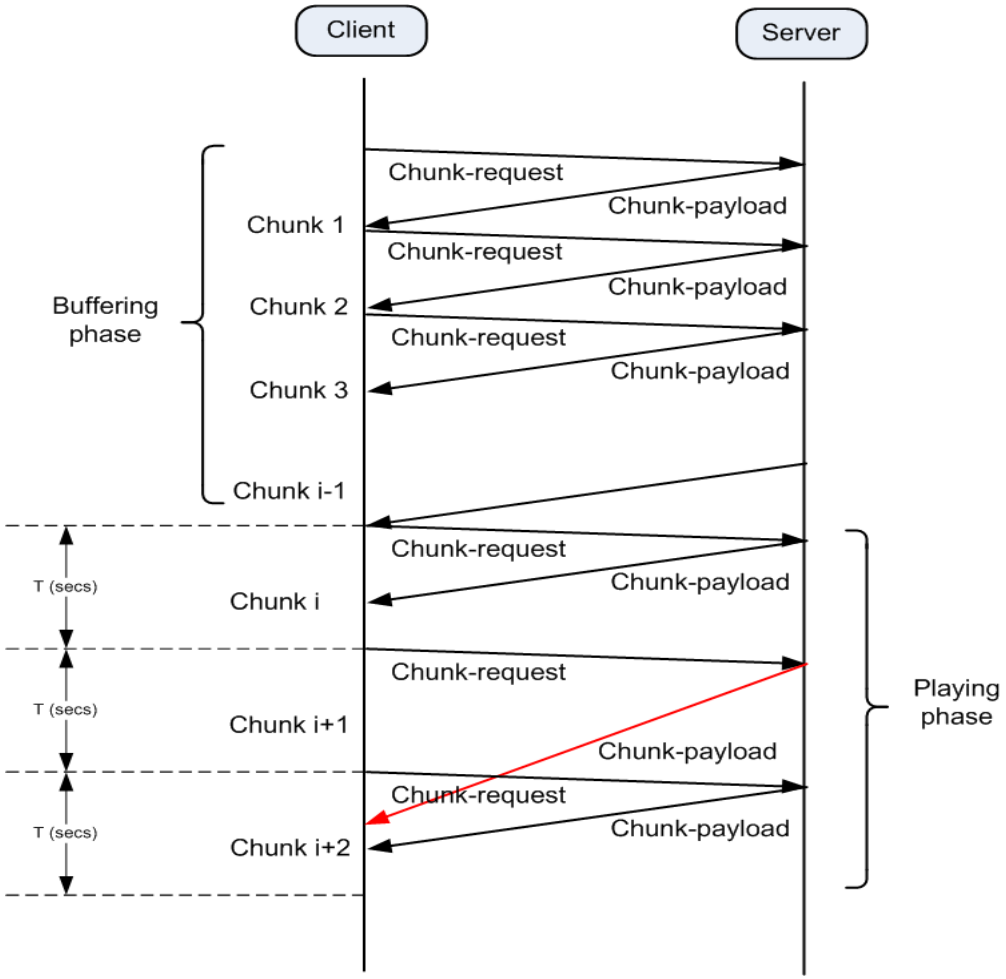


Correlated channels emulation (uniform)



Experimental results

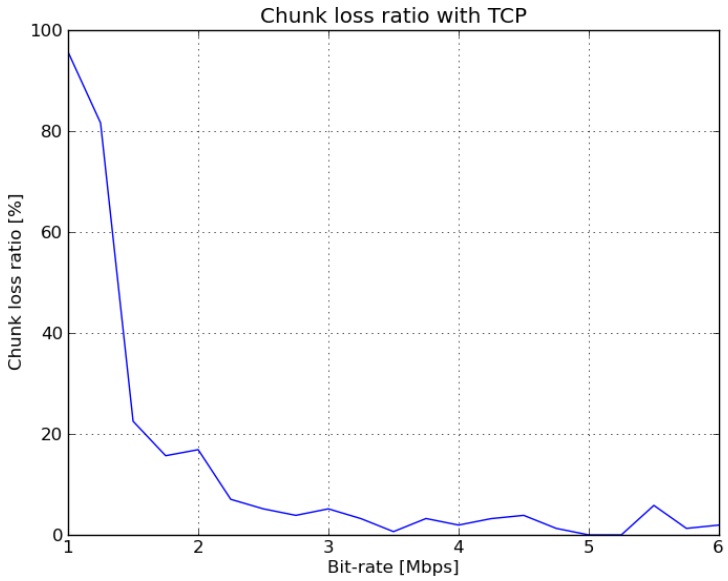
Client-server time line



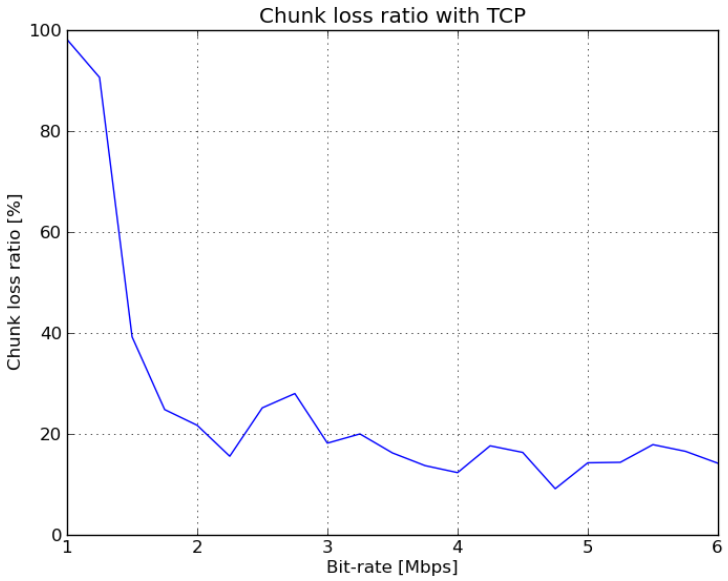
late-chunk-ratio = late-chunks/total-nb-chunks

TCP late-chunks-ratio

Linear distribution

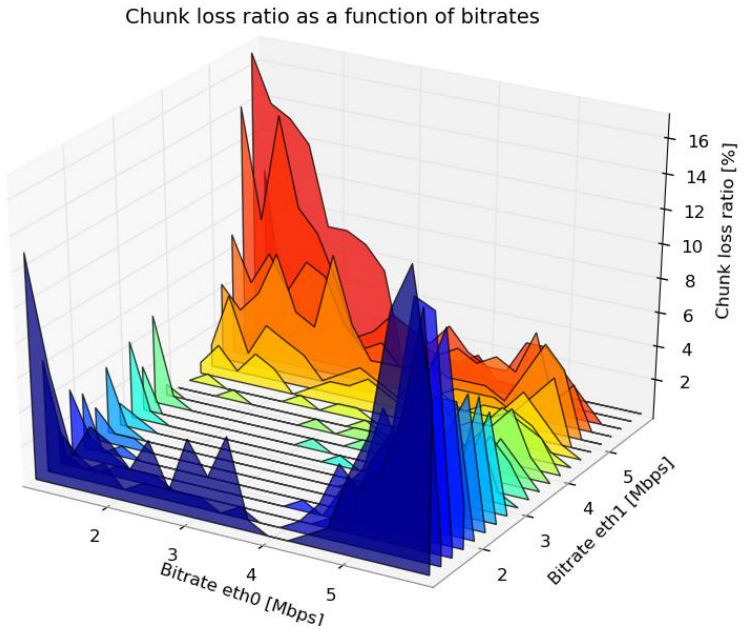


Uniform distribution

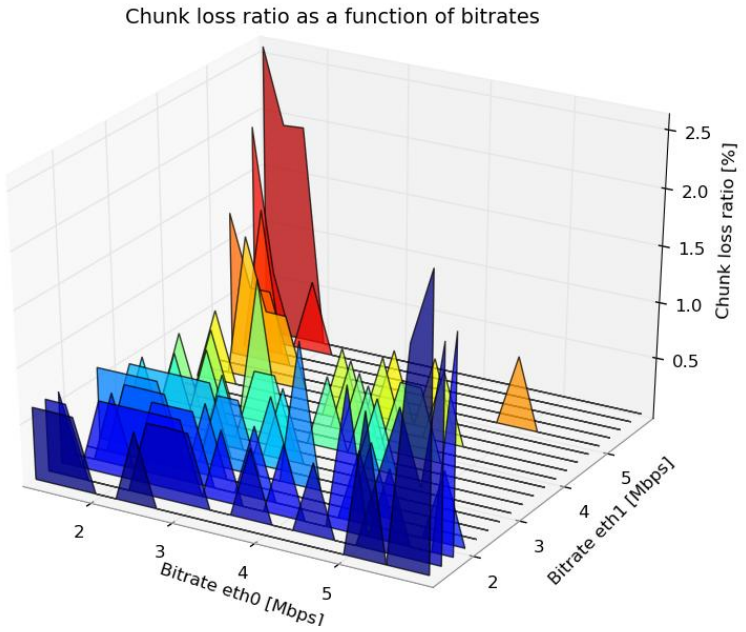


MPTCP late-chunks-ratio

Linear distribution



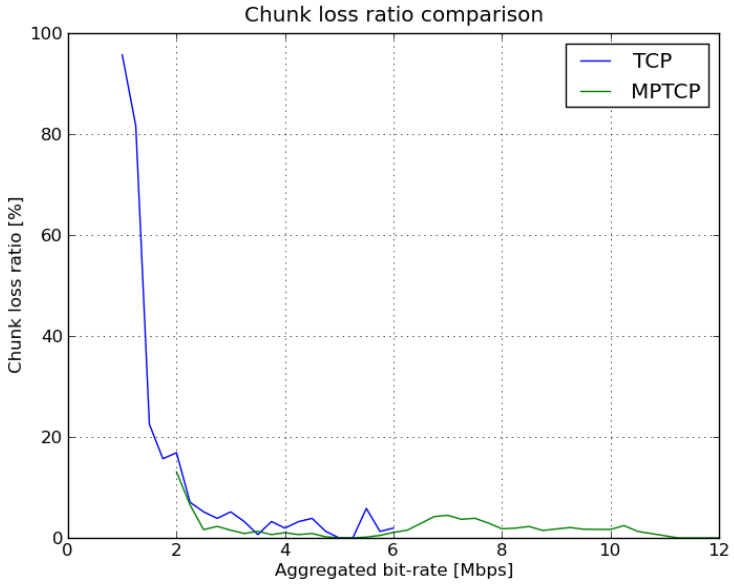
Uniform distribution



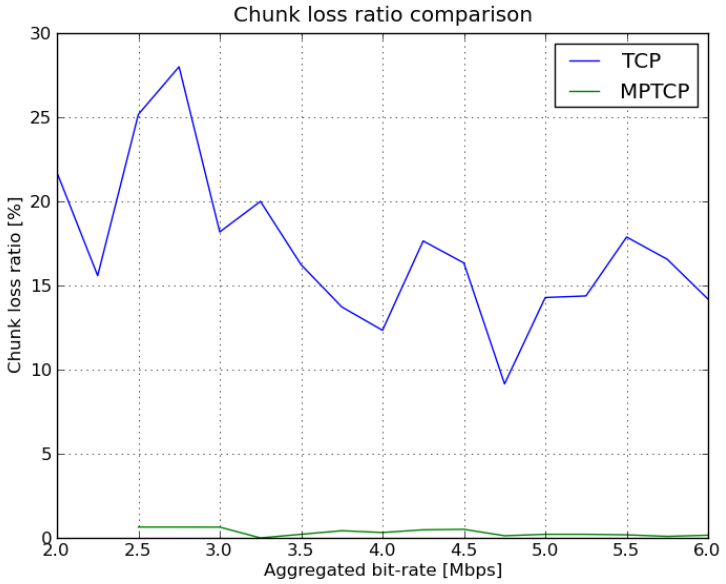
technicolor

TCP-MPTCP late-chunks-ratio (Uniform)

Whole range



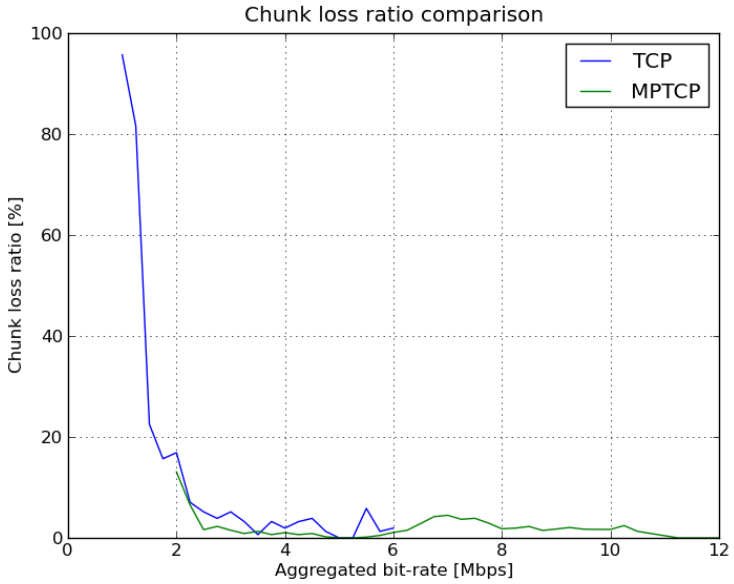
Zoom



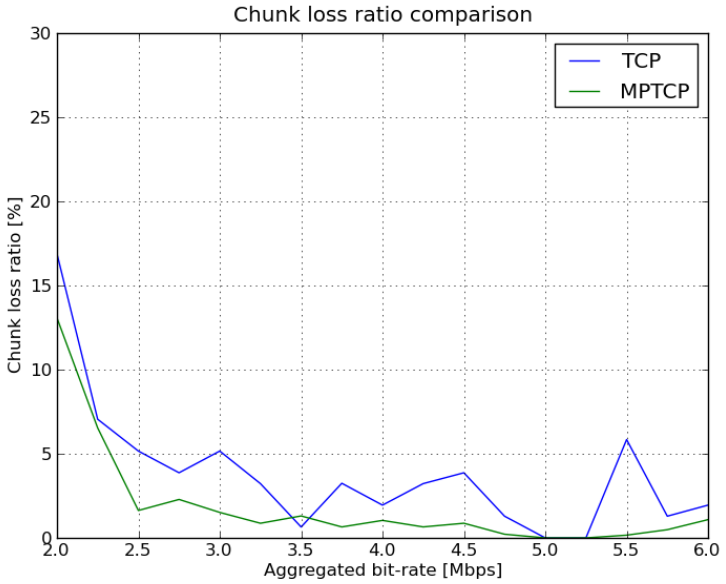
technicolor

TCP-MPTCP late-chunks-ratio (Linear)

Whole range

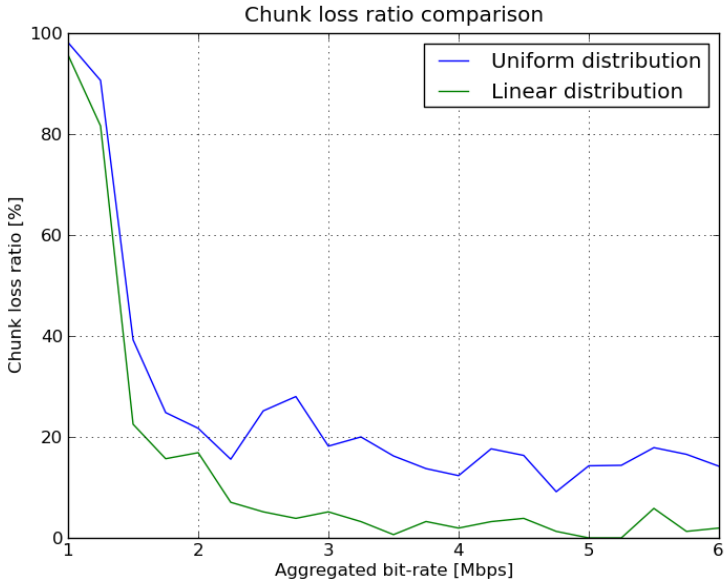


Zoom

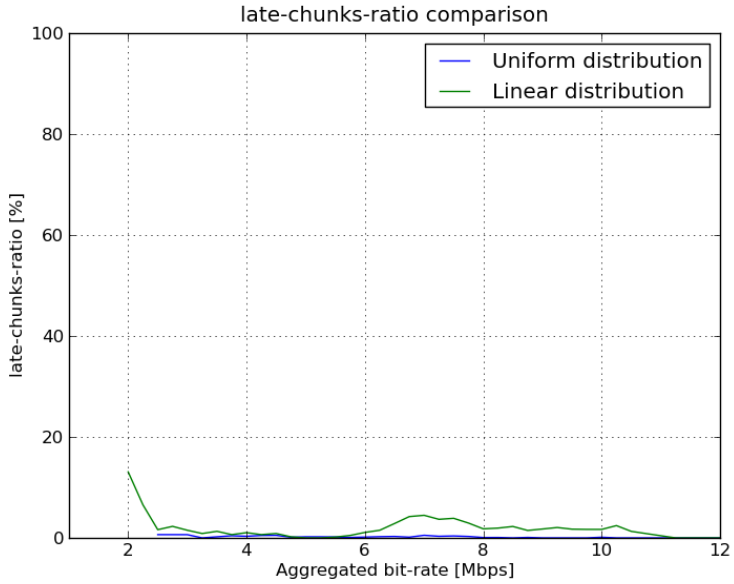


late-chunks-ratio comparison for both

TCP



MPTCP



Conclusions

- Experiment MPTCP with unmodified HAS.
- Results presented are a snapshot of ongoing tests. Other channel models are being tested.
- MPTCP provides higher reliability than TCP in all cases.
- Some results need further investigation to be explained:
  - A linear delay has more negative impact on MPTCP than a random delay.
  - A linear delay has more negative impact on MPTCP when bandwidth aggregation is around 7 Mbps.
  - Hypotesis: channel combinations with a large bit-rate disparity.