Wide Area Network Traffic Optimization

Ing. Ivan Klimek

Supervisor: Assoc. Prof. Ing. František Jakab, PhD.

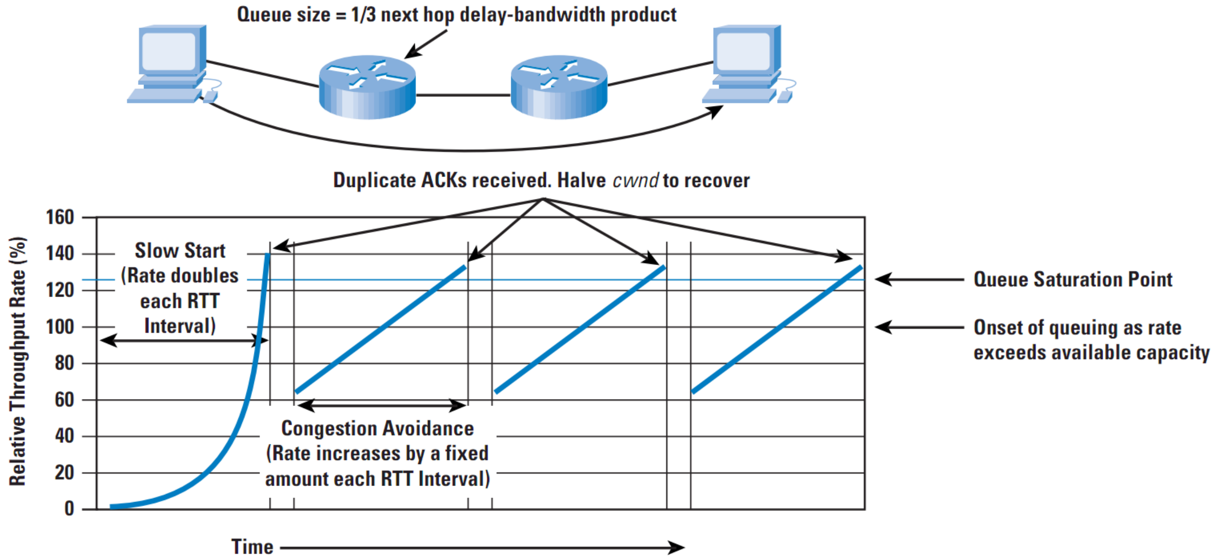
The problem definition

The transport protocol addresses the problem of achieving a <u>dynamic equilibrium</u> of maximum network efficiency, where the sending data rate is <u>maximized just prior</u> to reaching the bottleneck capacity after which congestion occurs.

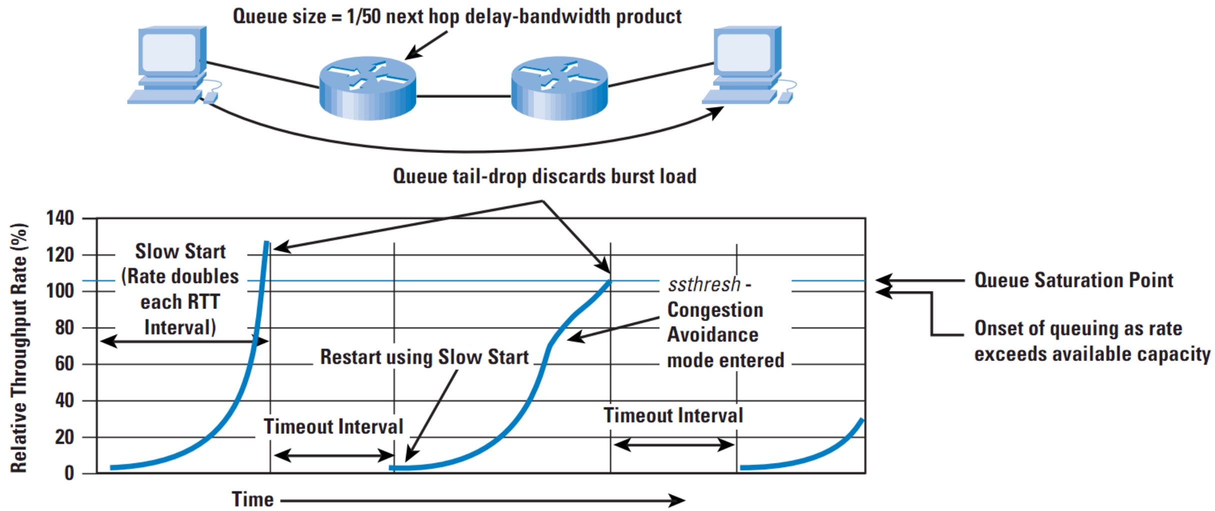
Transmission Control Protocol – packet drop is not an ideal congestion indicator

- 50-80% of TCP loss detections are triggered by <u>**timeout**</u> (RTO)

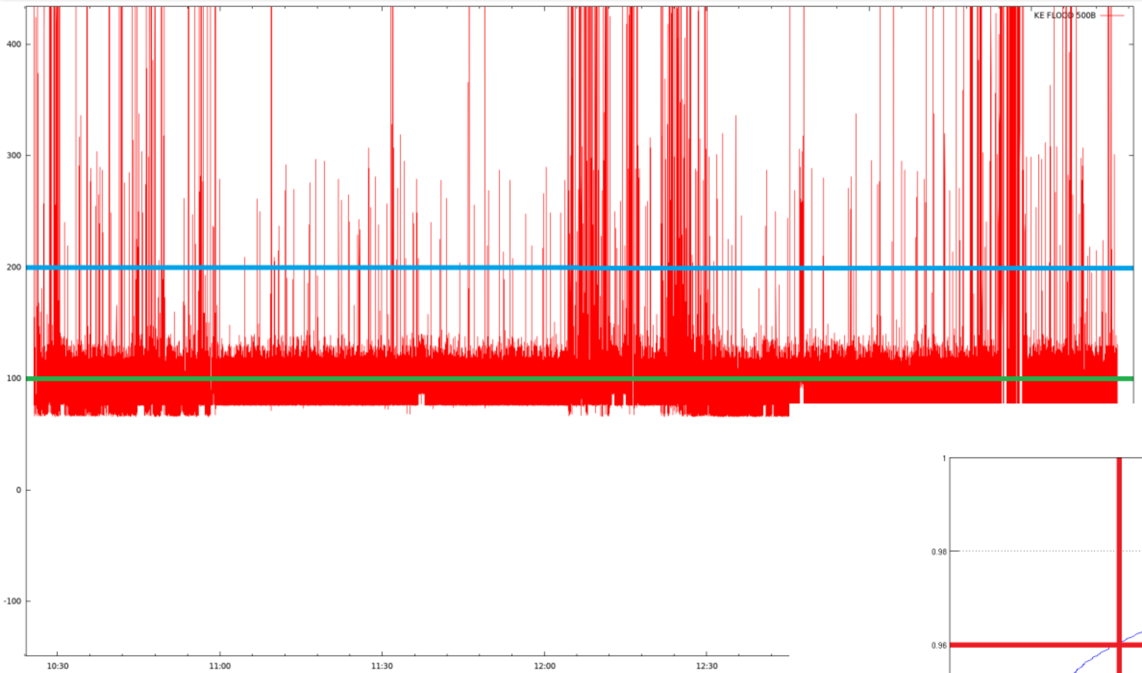
Fast Recovery ->



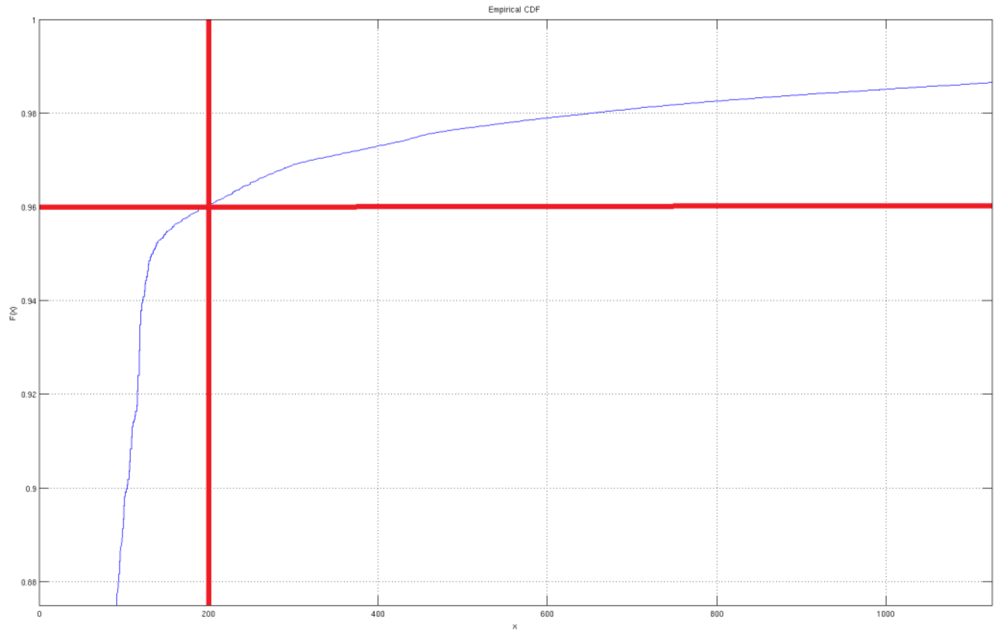
Timeout ->



Transmission Control Protocol – packet drop is not an ideal congestion indicator



Jitter on WWANs can cause spurious retransmissions



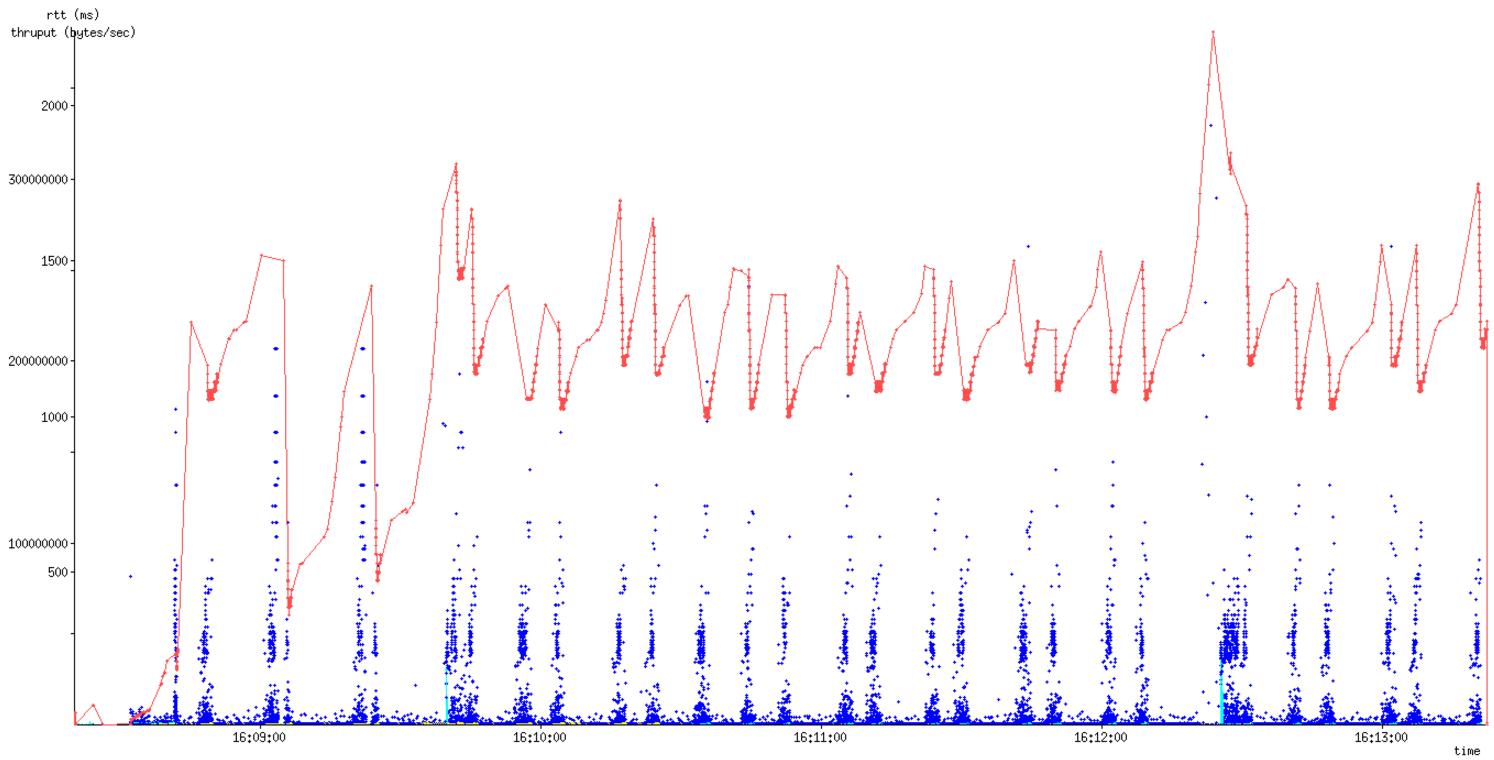
PING example on 3.5G WWAN,

4 percent of all packets have RTT > 200ms

Transmission Control Protocol – packet drop is not an ideal congestion indicator

- **Bufferbloat is a phenomenon whereby excessive buffering causes high latency/jitter, reducing the overall throughput**

Bufferbloat in action, blue marks throughput, RTT in red jumps from a few ms to several seconds

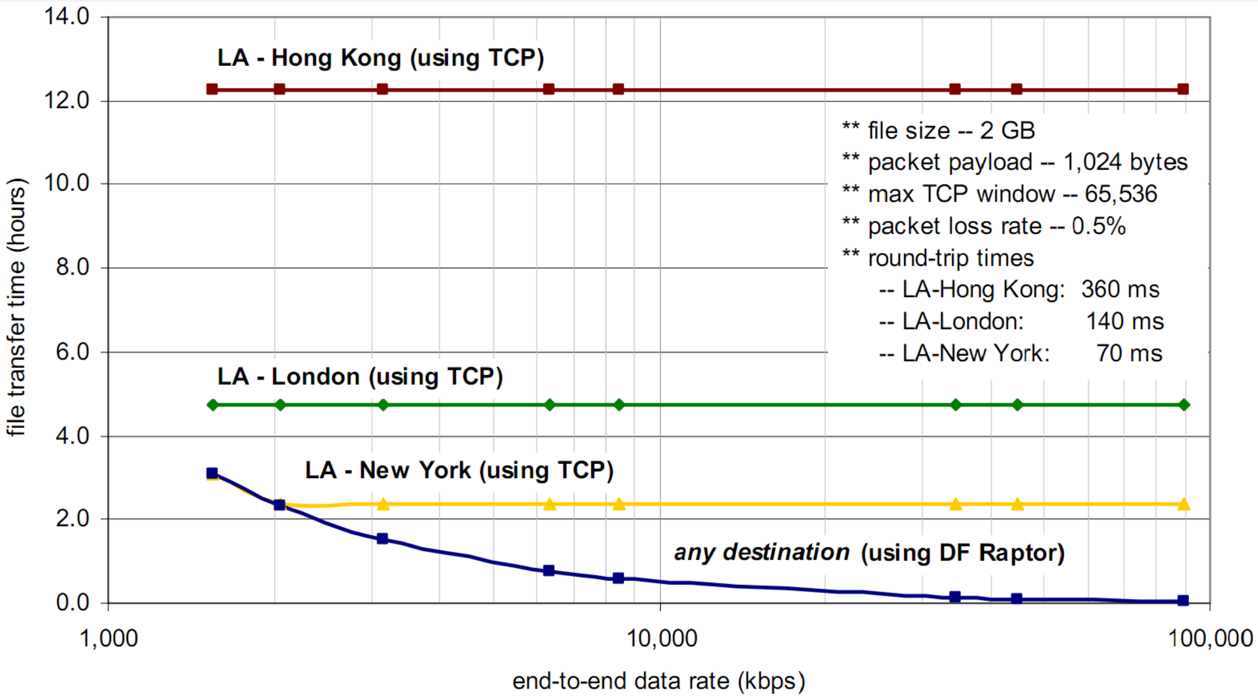


Transmission Control Protocol – packet drop is not an ideal congestion indicator

TCP rate/congestion control is almost perfect

in the <u>context</u> of Automatic Repeat reQuest (ARQ) protocols.

Forward Error Correction in Unicast Transport Protocols



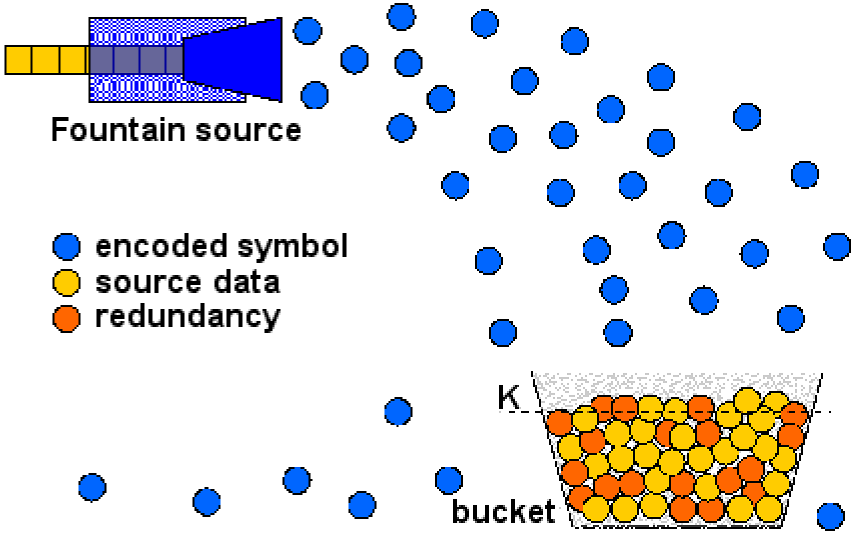
Mathis formula models TCP throughput as a <u>function of RTT and loss rate</u>

$$B(p) = \frac{MSS}{RTT} \frac{C}{\sqrt{p}}$$

- FEC-enabled protocols for throughput optimization (FBP, FECTCP …)
- FEC-enabled protocols for delay optimization (RAPID)
- Limiting factors: computational complexity, data overhead

Raptor Codes – the game changer?

- Raptor codes are the first known class of fountain codes with **linear** time encoding and decoding
- The latest generation, the RaptorQ (RQ) codes, standardized only in August 2011, drastically improve the failure-overhead properties
- The failure probability for K packets e.g. **without excess packets is 10^{-2} percent**, for one excess packet 10^{-4} and for two excess packets 10^{-6}



TCP-Friendliness

Under <u>high loss rate</u> regions where TCP is well-behaving, the protocol <u>must behave like TCP</u>, and under <u>low loss rate</u> regions where TCP has a low utilization problem, it can use <u>more bandwidth</u> than TCP.

TCP-Friendliness

Can a RQ-enabled transport protocol be TCP-Friendly but not share TCP's inefficiencies?

Options: packet drop, Delay Based Congestion Control, ECN

The vision - Universal Transport Protocol

- **Separate** data and control channels (RTP, UDT, FASP …)
- **Non-blocking feedback** used for flow control purposes (ECN, jitter as BW utilization indicator (FBP…), DBCC (µTP))
- **Lightweight** stream structure (SST)
- RQ coded stream enables a **more aggressive and stable** BW estimation (better estimate thru allowing for small packet loss, while rapid packet loss increase indicates congestion -> enable TCP-Friendly mode)
- Build on top of **UDP**
- UTP needs to solve existing problems **without** the need to replace the existing infrastructure (Bufferbloat, WWAN efficiency, fairness …)

Proposed theses for dissertation

- Design and prototype a Fountain (RaptorQ) code based unicast transport protocol whose performance would approach channel capacity even under difficult conditions (packet loss, long RTT, jitter) and at the same time be optimized for smallest possible delay and minimal waste redundancy
- Design and prototype rate/congestion control mechanism optimized for the prototyped protocol with the ability to coexist with legacy TCP streams
- Analyze the proposed approaches using simulations
- Experimentally verify the simulation model

THANK YOU!

In protocol design, perfection has been reached not when there is nothing left to add, but when there is nothing left to take away. (RFC 1925, originally de Saint-Exupéry)

Laboratórium počítačových sietí
Katedra počítačov a informatiky
Fakulta elektrotechniky a informatiky
Technická univerzita v Košiciach
Letná 9
042 00 Košice

VoIP @ Lab	www.cnl.tuke.sk/voip
VirtualLab @ Lab	www.cnl.tuke.sk/vlab
Video @ Lab	www.cnl.tuke.sk/video
VRVS @ Lab	www.cnl.tuke.sk/vrvs
Synets @ Lab	www.cnl.tuke.sk/synets
QoS @ Lab	www.cnl.tuke.sk/qos

Technická univerzita v Košiciach

Fakulta elektrotechniky a informatiky

Katedra počítačov a informatiky

OPONENT QUESTIONS

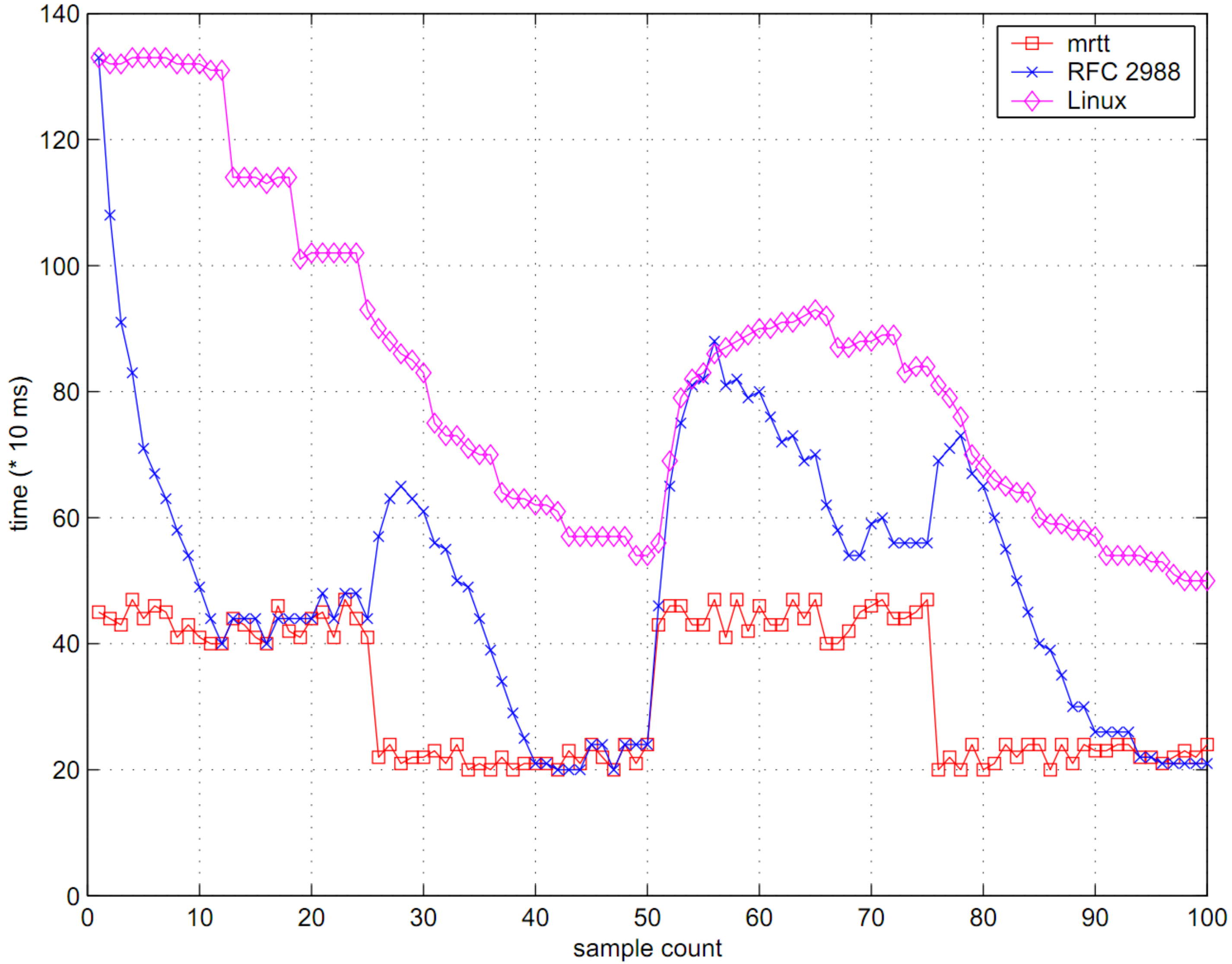
What does the vartt variable in Table 2 mean?

- Linux uses non-standard RTO calculation
- Less weight for the measured mean deviance (MDEV) when the measured RTT decreases significantly bellow the smoothed average

```
if (R < SRTT and |SRTT - R| > MDEV) {
        MDEV <- 31/32 * MDEV + 1/32 * |SRTT - R|
} else {
        MDEV <- 3/4 * MDEV + 1/4 * |SRTT - R|
}
```

- R is the recent RTT measurement, SRTT is the smoothed average RTT
- Linux does not directly modify the RTTVAR, but makes adjustments first on the MDEV variable which is used in adjusting the RTTVAR which determines the RTO
- The SRTT and RTO are calculated standardly

What does the vartt variable in Table 2 mean?



What transport protocol do the SSH control messages of FASP use?

- FASP transfers use one TCP port to establish the initial SSH connection from client to server, and one UDP port per concurrent client transfer session on the server
- By default uses TCP port 33001 for SSH

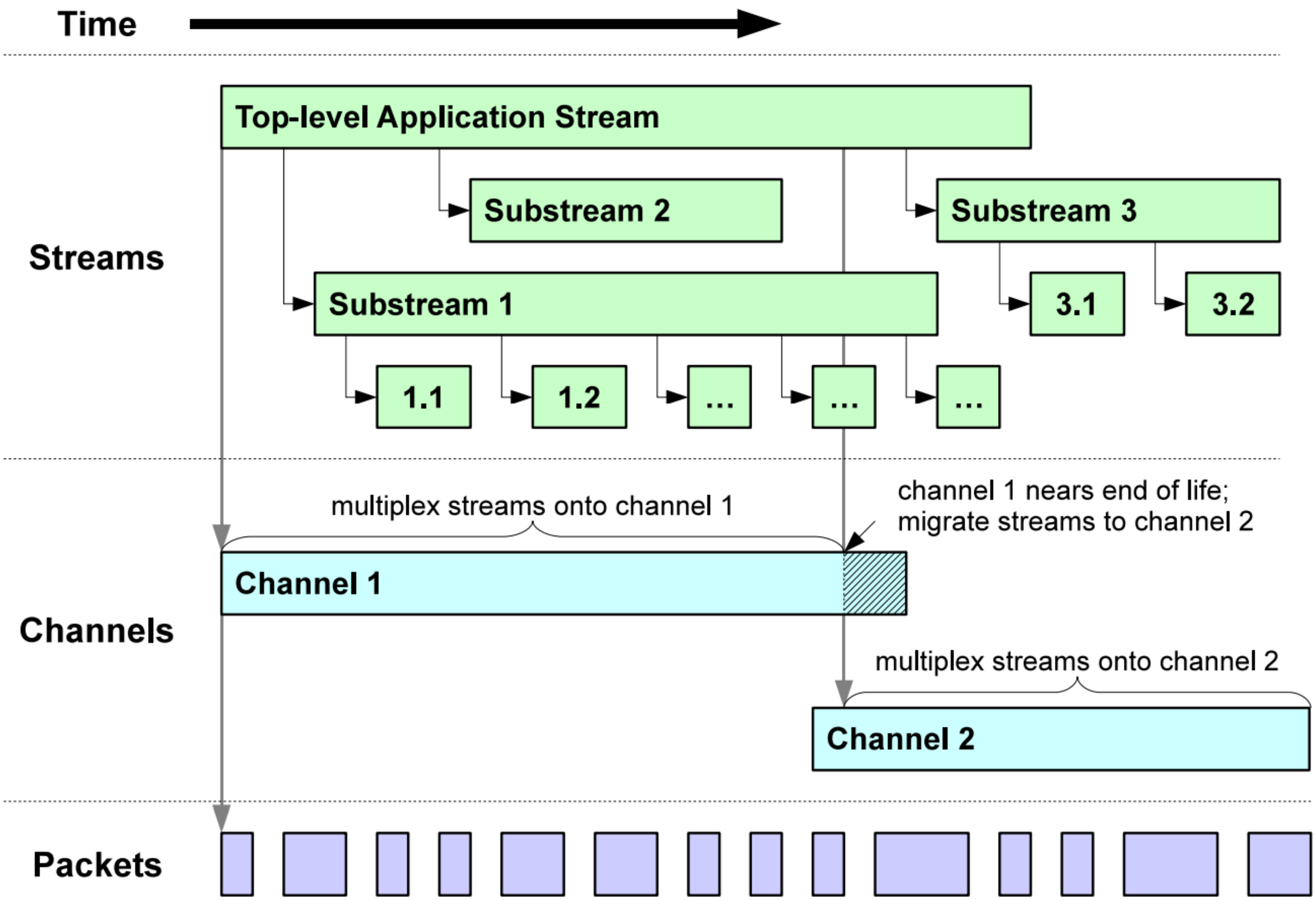
Does the asymmetry of delay have an impact on the effectivity of the transport protocol?

- Unbalanced uplink/downlink delay may downgrade QoE for applications such as VoIP
- Generally only the Round-trip time is considered

What characteristics should the proposed protocol have?

- Binary, textual wouldn’t be BW efficient
- There is no clear difference between stream/byte and message oriented protocol in our context, if the data can be delivered in one message it will be one message, if it needs more or/and if loss happens, stream of messages will be used but the protocol does not necessarily preserve any implicit structure within a transmitted byte stream, is there any value in making it message oriented?
- Multi-streaming, multi-homing – design (and maybe even code) will be inspired by the Structured Stream Protocol
- As simple as possible, focus will be given to rate/congestion issues as already discussed

What characteristics should the proposed protocol have?



What mechanisms will the design of flow/congestion control be inspired from?

- Delay Based Congestion Control – starting from simplistic implementations as seen in µTP moving to more advanced concepts
- Explicit Congestion Notificaiton
- Jitter based bandwidth estimation
- Instead of “touching” the maximal capacity as TCP does, finding it and holding it using the properties of RQ that enable to have a small packet loss without massive overhead
- More aggressive initial window than in current TCP implementations (WATCP inspired, start large, scale down if congestion detected), we still need scaling to limit the amount of waste redundancy, the scheme to be used is still to be determined