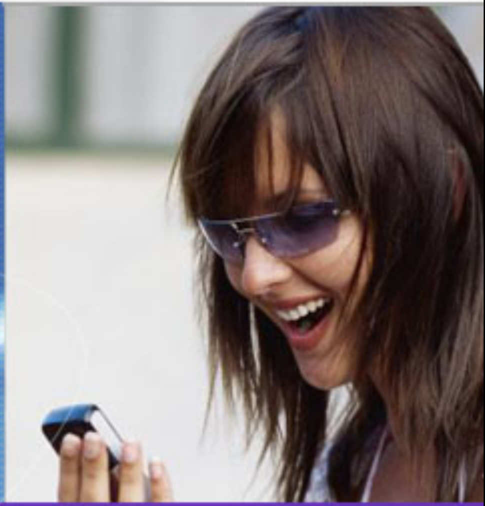
Interactions Between HTTP Adaptive Streaming and TCP
NOSSDAV 2012



Jairo Esteban,, Steven Benno, Andre Beck, Yang Guo, Volker Hilt, Ivica Rimac
Bell Labs Research

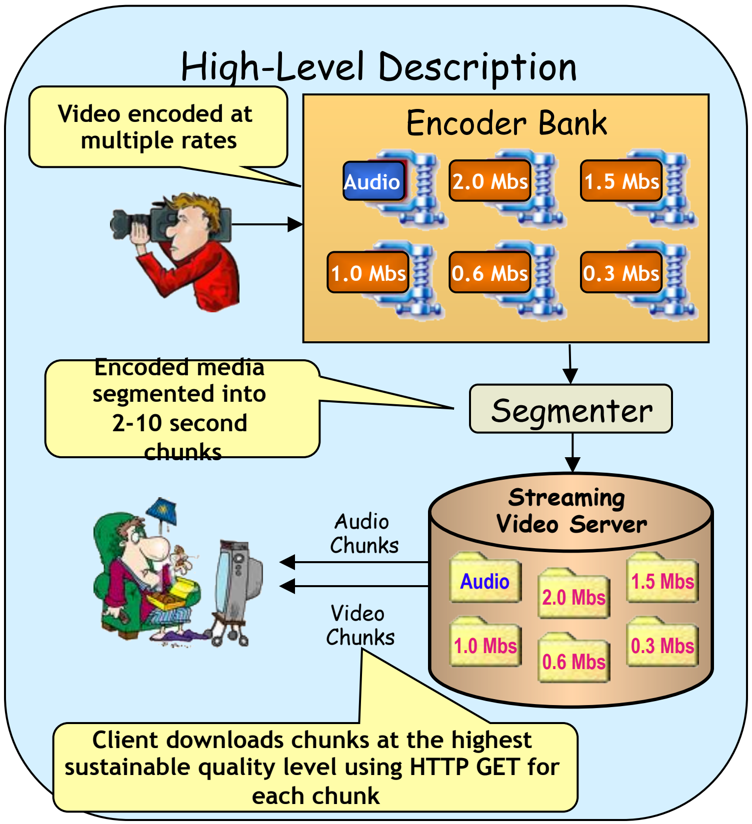
All Rights Reserved © Alcatel-Lucent 2012

HTTP Adaptive Streaming Video Overview

- Quickly becoming preferred method of video delivery:
  - Adapts to changing network conditions to give best video quality possible
  - Provides fast startup, quick seek times, and smooth playback
- Uses standard HTTP protocol/caches/proxies
  - Traverses firewalls
- Generates massive amount of data and traffic

Major HAS Players

- Pioneered by Move Networks



Embraced by Content Providers

All Rights Reserved © Alcatel-Lucent 2012

Alcatel·Lucent

Questions and Concerns

How is the client impacted by dynamic network conditions, congestion, packet loss?

Does caching have a negative impact on QoE due to different latencies between cache hits and misses?

What happens when HAS clients compete against greedy TCP cross-traffic for bandwidth?

What are the traffic patterns created by HAS flows?

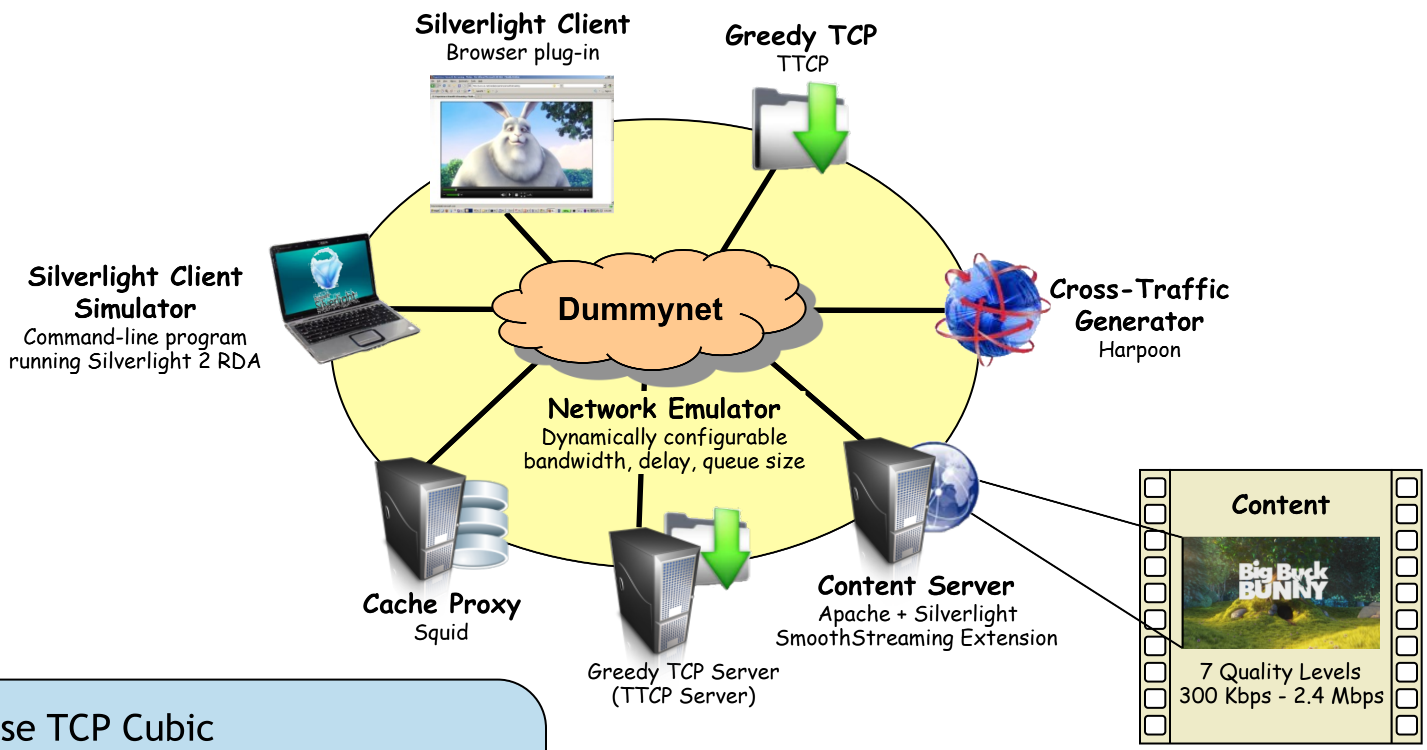
What can be done in the server/network to improve quality?

How efficiently do HAS flows use the available bandwidth?

What is the impact of delay on video quality?

All Rights Reserved © Alcatel-Lucent 2012

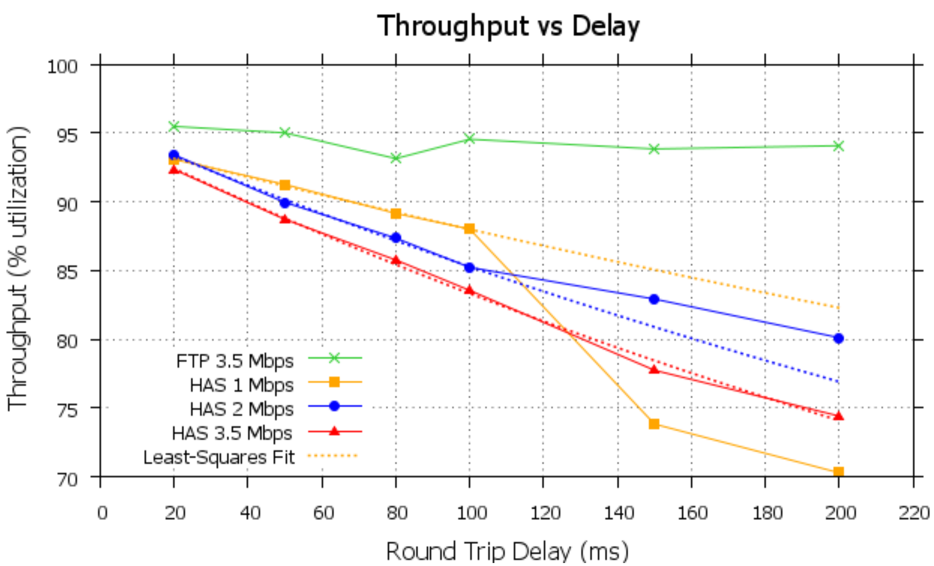
Testbed Setup

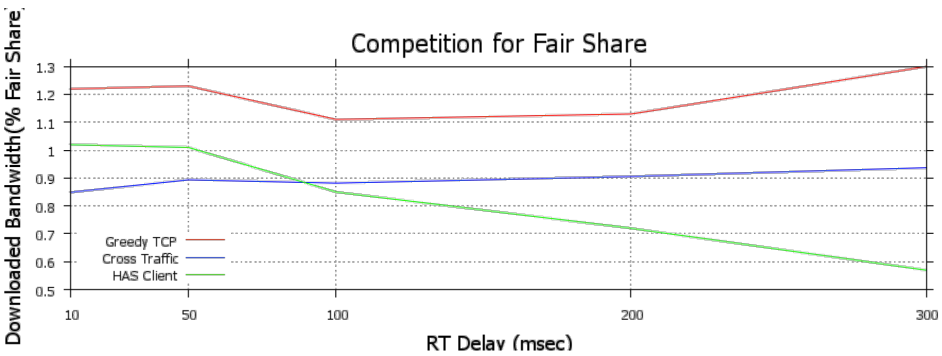


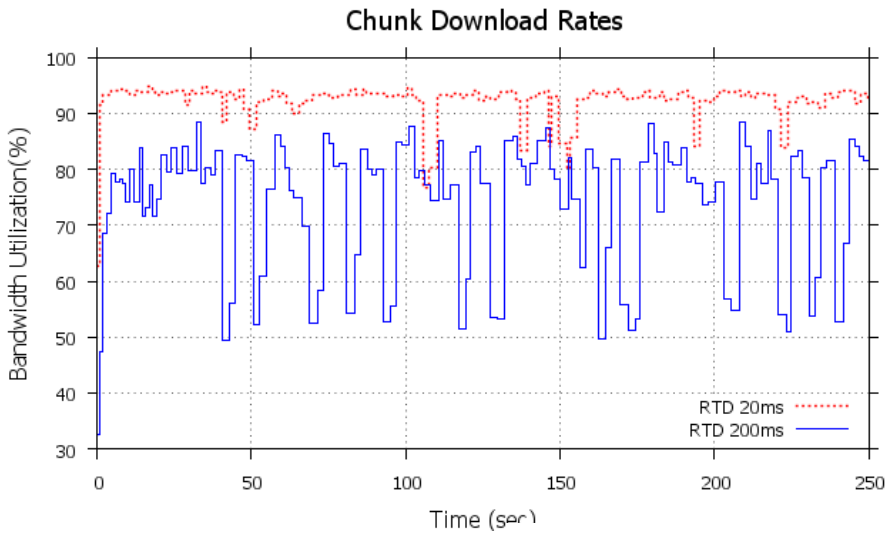
- All servers use TCP Cubic
- Content: 10-min video with 7 quality levels
- Dummynet:
  - Buffer size: 2 x BDP
  - Tail drop policy

All Rights Reserved © Alcatel-Lucent 2012

Chunk Download Penalty





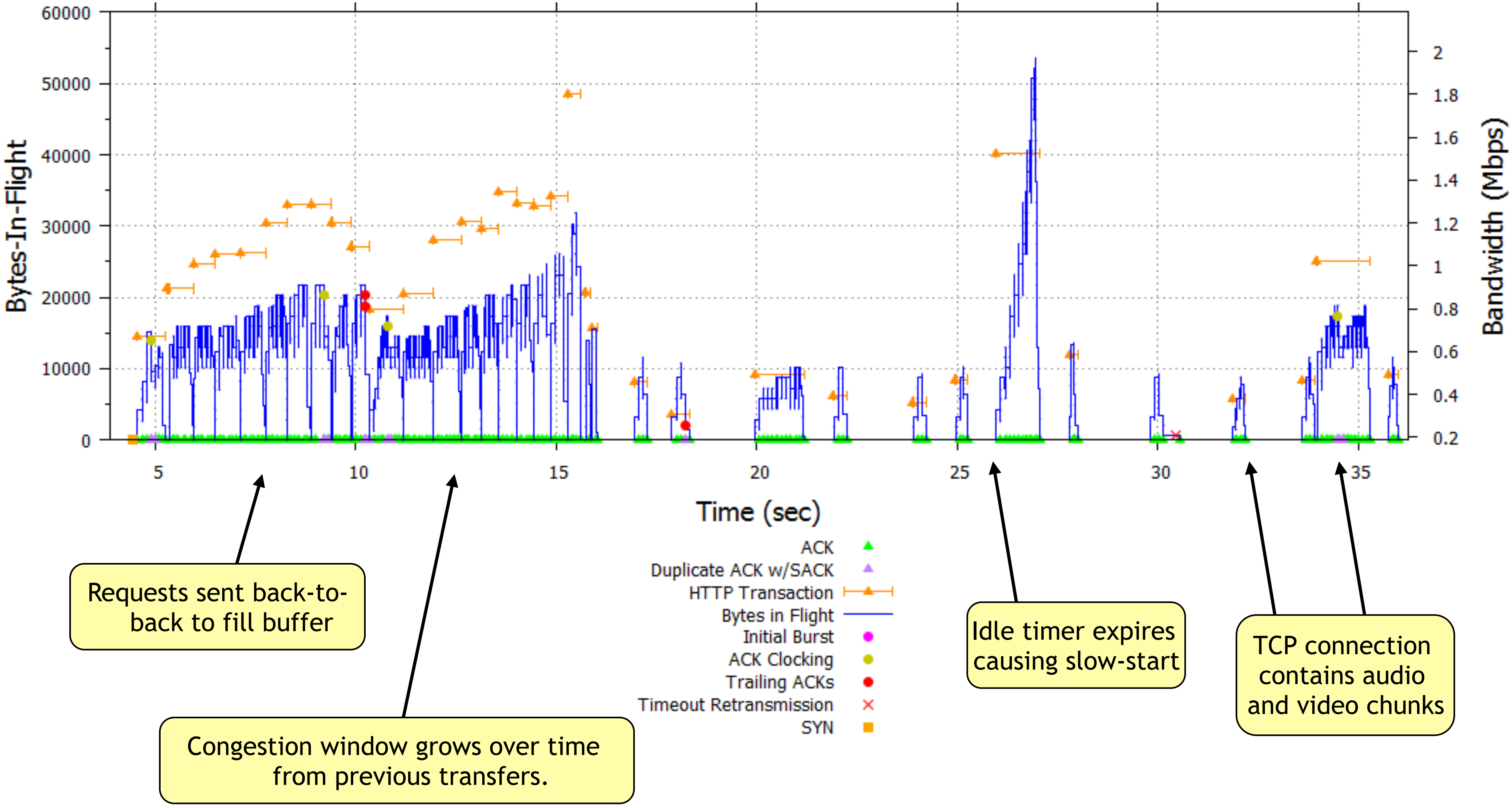


Observations

- HAS client does a reasonable job of getting its fair share but looses throughput as RTT increases.
- Downloading chunks is less efficient than large downloads, e.g. progressive video download.
- Penalty inversely proportional to RTT.
- Start-and-stop nature of HAS traffic pattern cause of TCP inefficiency.

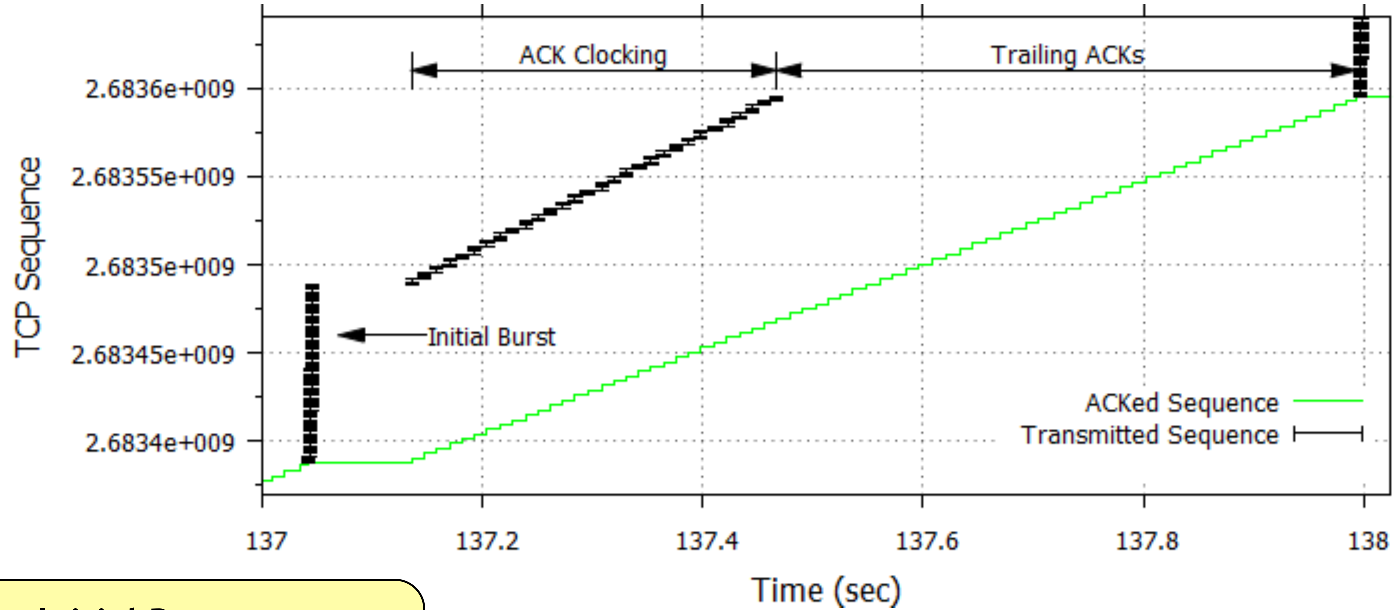
All Rights Reserved © Alcatel-Lucent 2012

HAS Traffic Pattern: Smooth Streaming Session



All Rights Reserved © Alcatel-Lucent 2012
Alcatel·Lucent

Three Phases of TCP Data Transfer



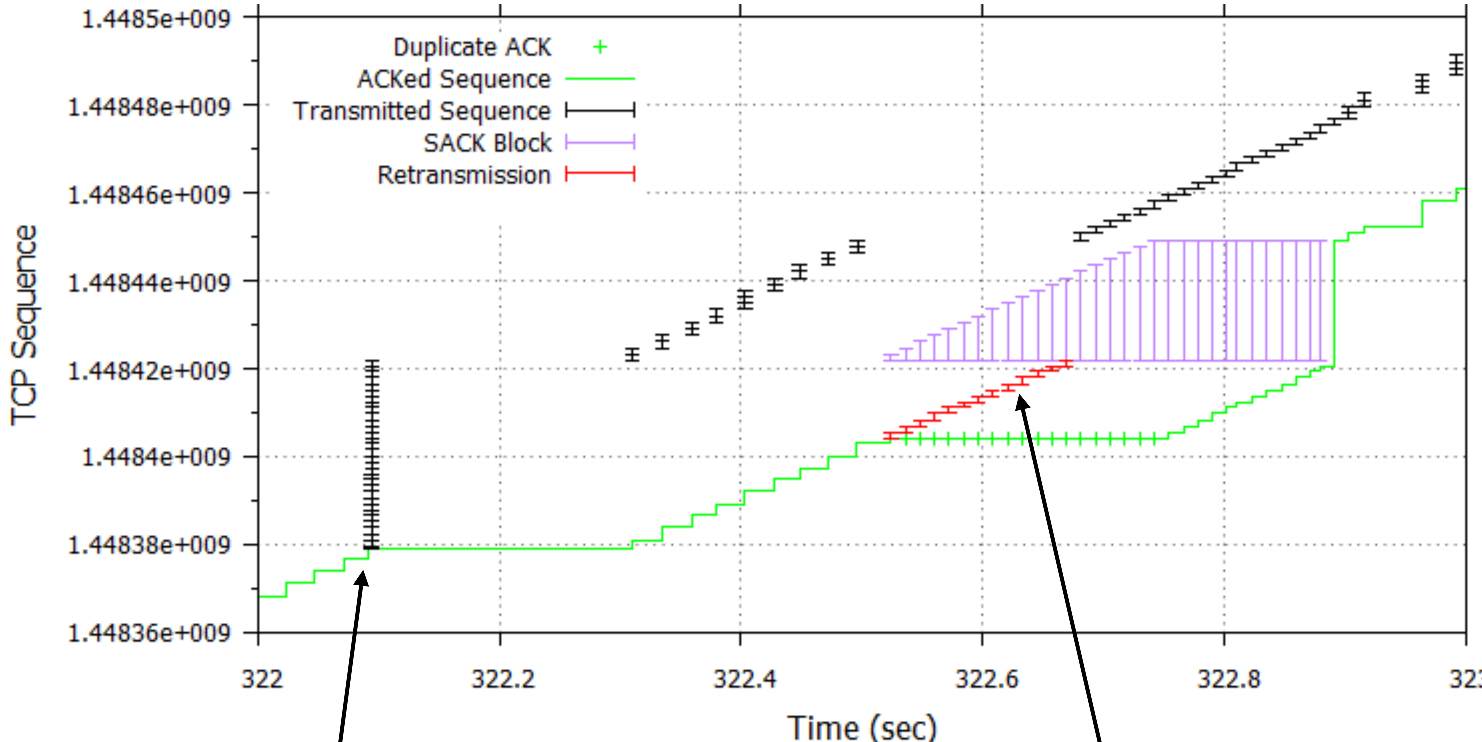
Initial Burst
Sender fills congestion window at start of each response

ACK Clocking
Sender receives ACKs from receiver and has more data to send

Trailing ACKs
Sender sent all packets at least once and is waiting for outstanding ACKs to return

All Rights Reserved © Alcatel-Lucent 2012
Alcatel·Lucent

Packet Losses: Initial Burst



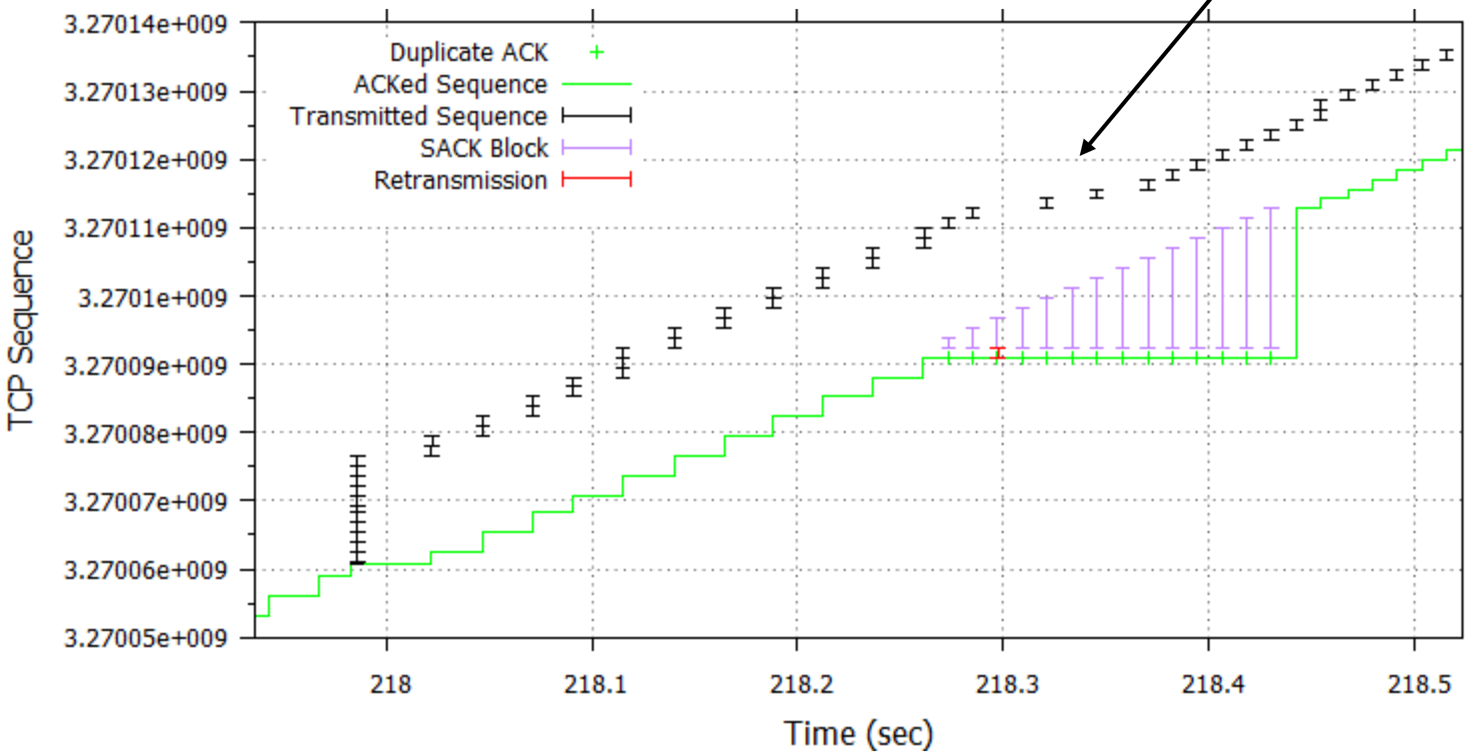
Congestion window grows over time from previous transfers.

Burst of packets overwhelms network bottleneck, causing multiple packet losses.

All Rights Reserved © Alcatel-Lucent 2012

Packet Losses: ACK Clocking

SACK, Fast retransmit, and fast recovery minimizes impact of packet loss.



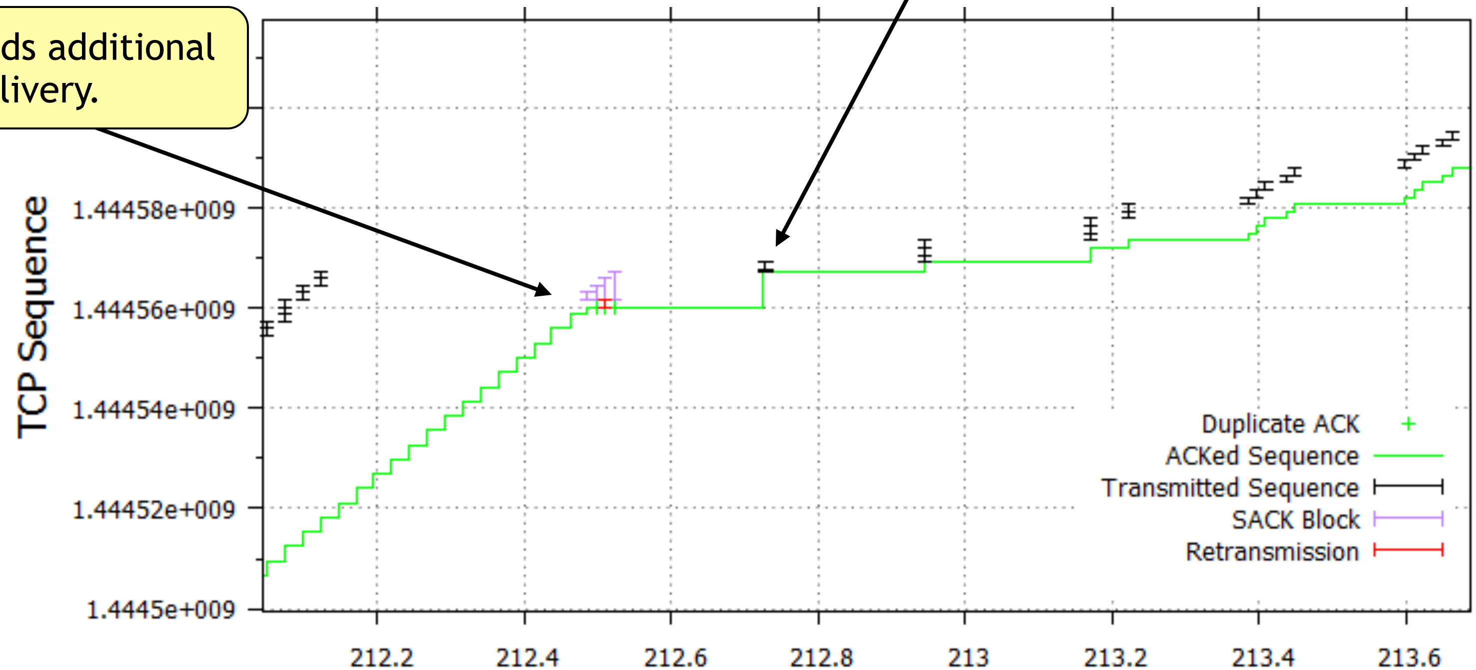
Most desirable phase for packet losses because mechanics of TCP fully utilized.

All Rights Reserved © Alcatel-Lucent 2012

Packet Losses: Trailing ACKs

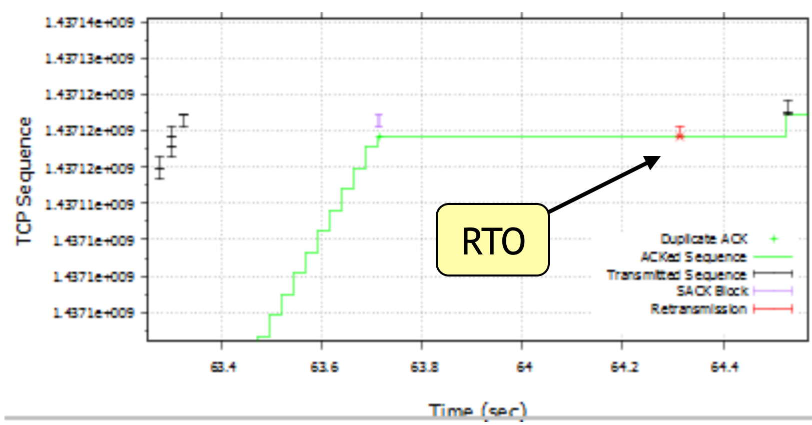
Impact of packet loss felt in next transmissions because of reduced *cwnd*.

Trailing ACK loss adds additional RTT to delivery.



Worst phase for packet losses:

- Trailing ACK packet loss increases latency
- Risk of retransmission timeout (RTO).
- Fast retransmit unavailable because of lack of new data to send.
- Congestion window severely reduced for next transmission.



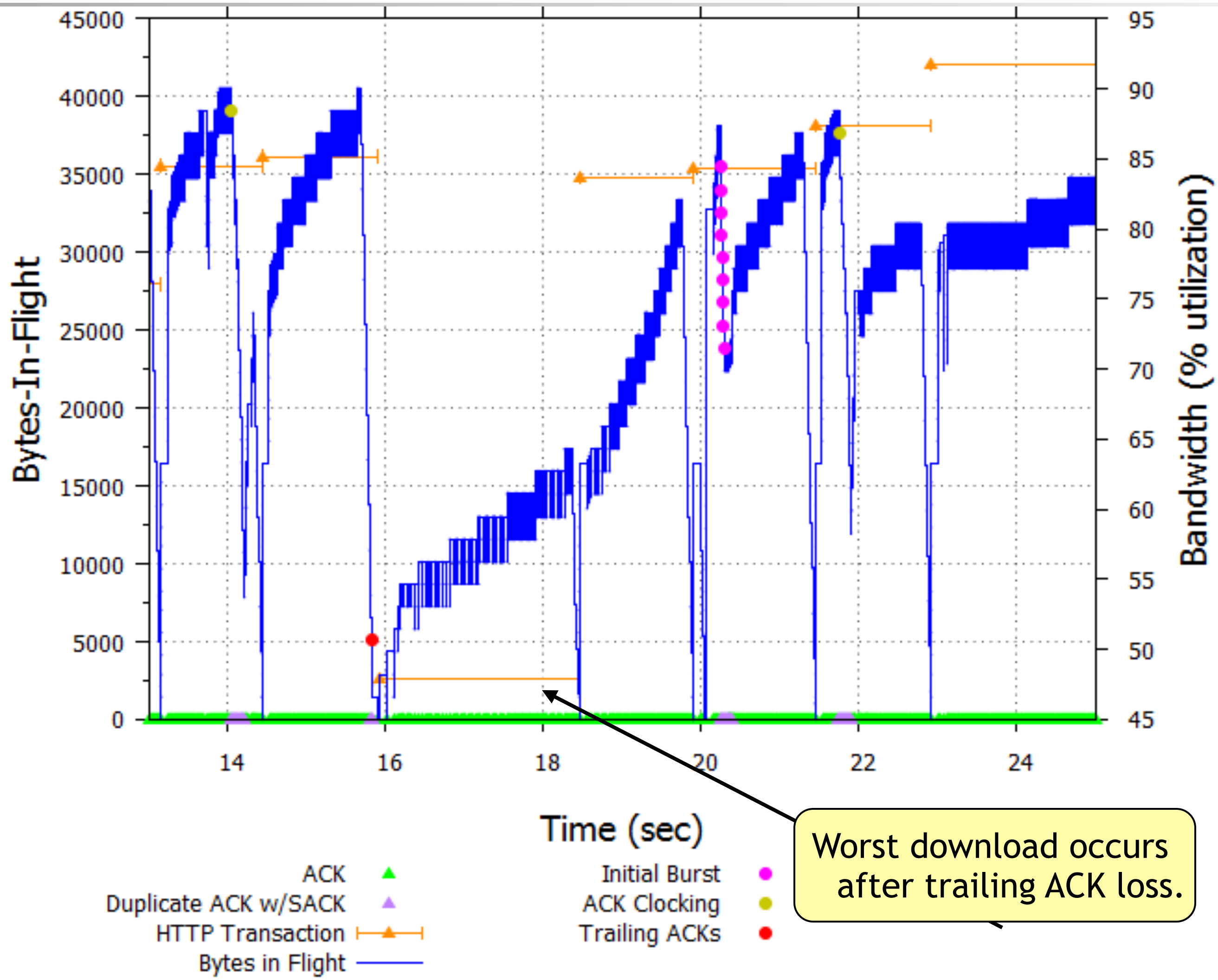
All Rights Reserved © Alcatel-Lucent 2012

Alcatel·Lucent

Packet Losses: All Three Phases

Conclusions

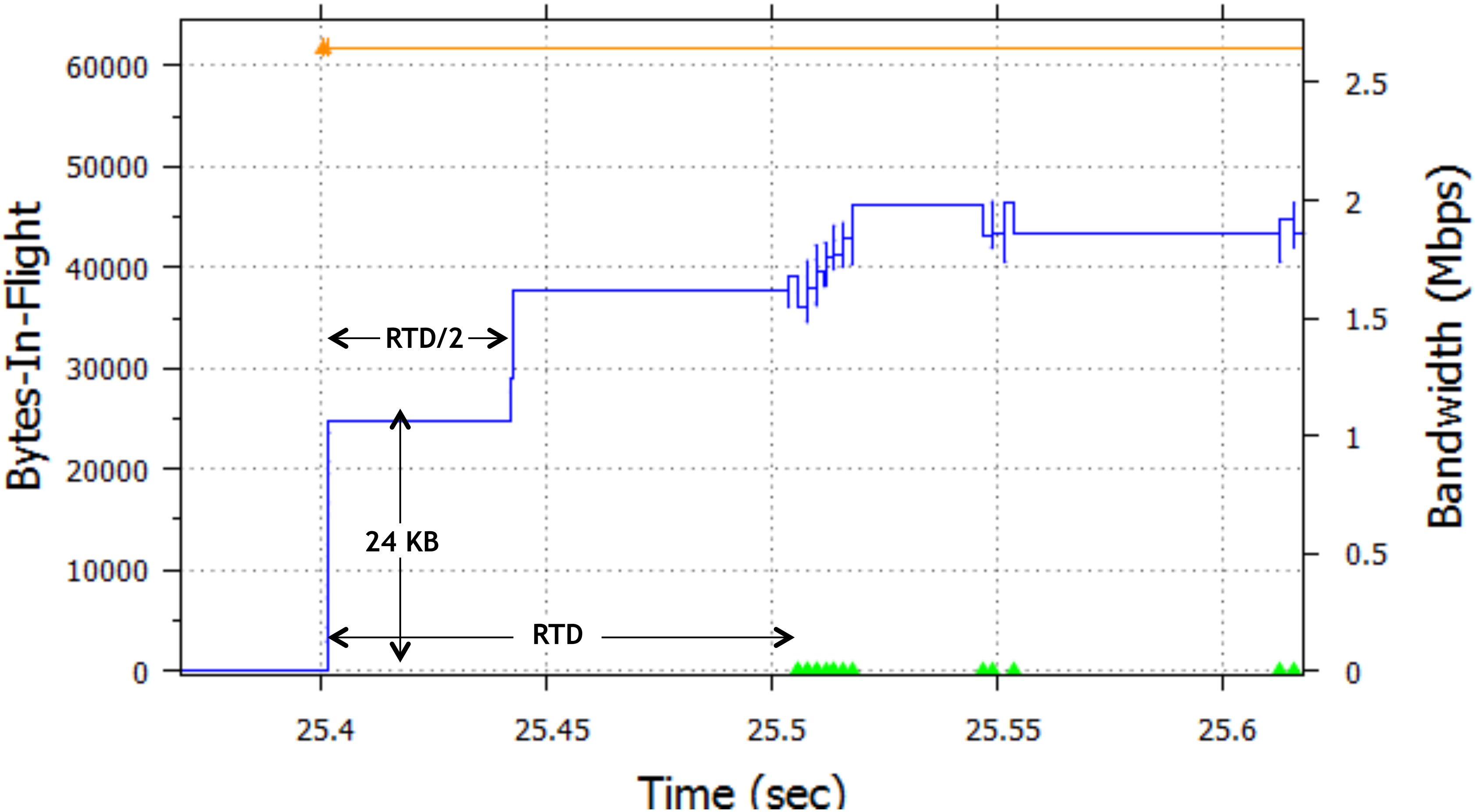
- Location of packet losses matters.
- Packet loss rate can be deceiving.
- Initial burst and ACK clocking losses recover relatively quickly.
- Trailing ACK phase worst for packet losses. Fast retransmit unavailable because of lack of new data to send.
- TCP connection penalized because it ran out of data during recovery phase.



All Rights Reserved © Alcatel-Lucent 2012
Alcatel·Lucent

Application-Level Pacing

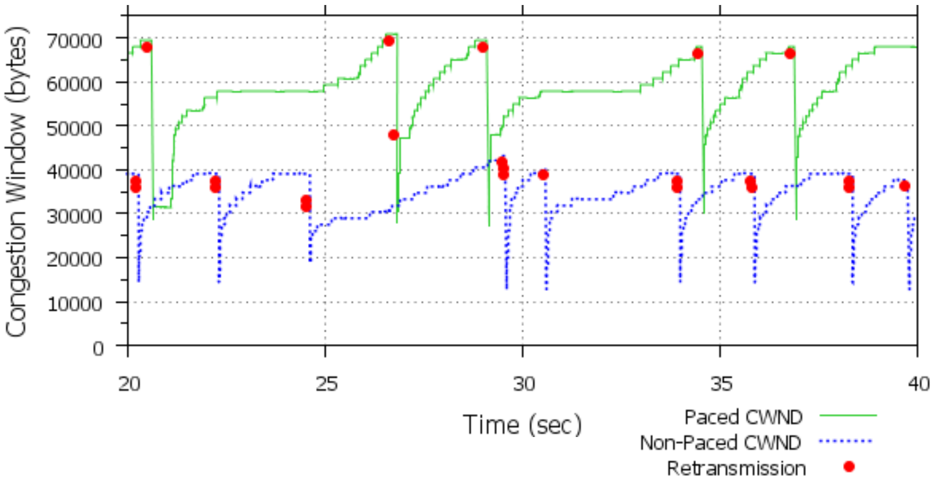
Pacing
Spreads transmission of data across RTT instead of initial burst.



All Rights Reserved © Alcatel-Lucent 2012
Alcatel·Lucent

Pacing Performance

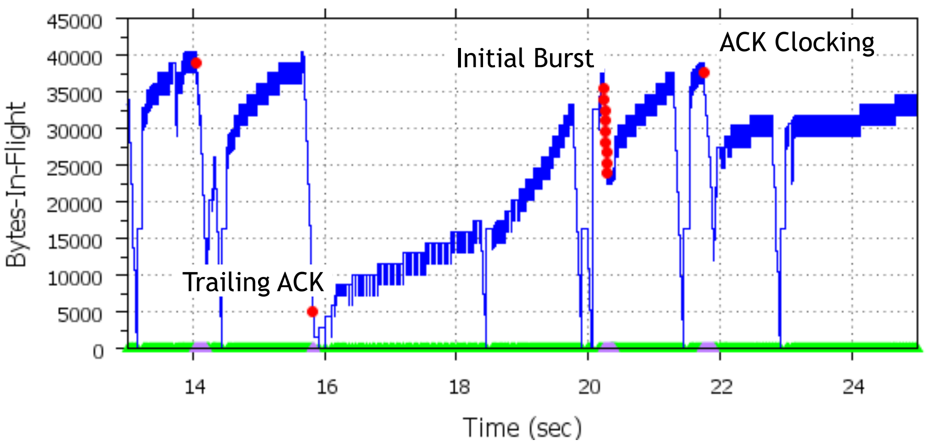
Congestion Window: Paced vs Unpaced



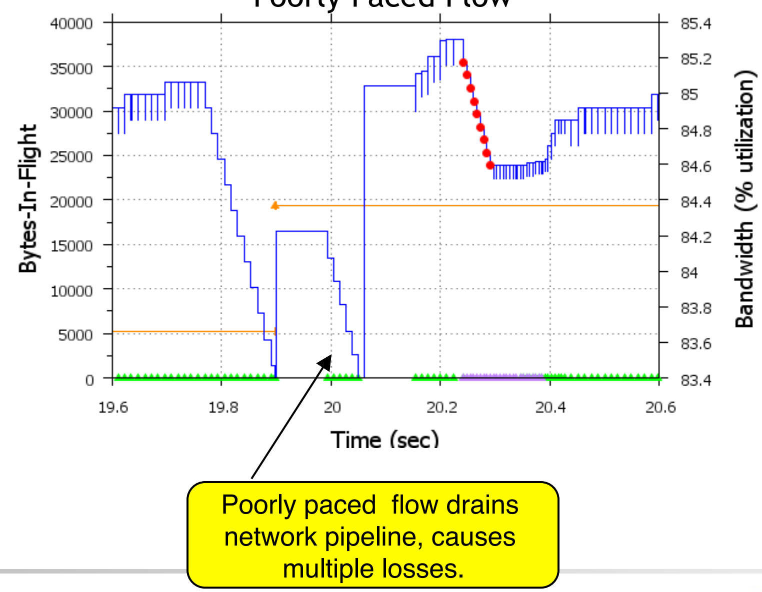
Pacing

- Pacing fills entire pipeline without packet loss, doubling bytes-in-flight.
- Pacing is no guarantee of desired behavior. Can shift packet losses to trailing ACK phase, which is undersirable.
- Poorly paced flow can reduce throughput.
- **Despite 62.2% increase of bytes-in-flight, bandwidth improved only 4.8%.**

Pacing: No Guarantees

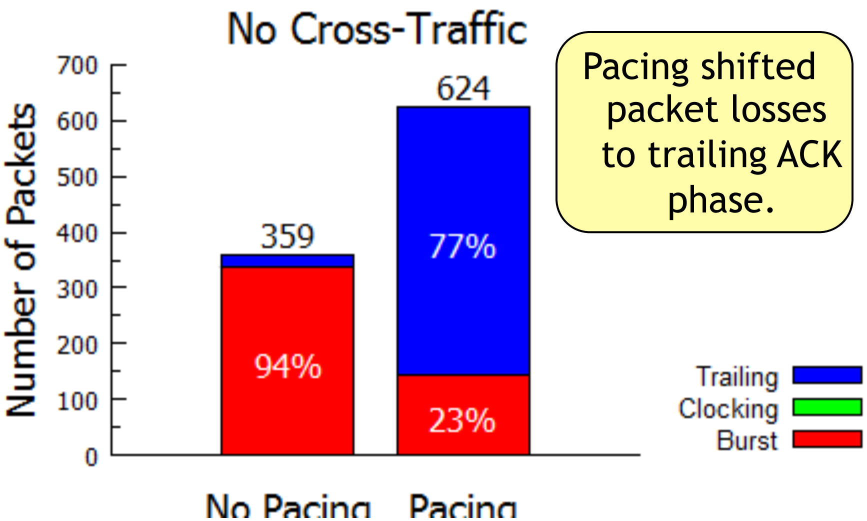


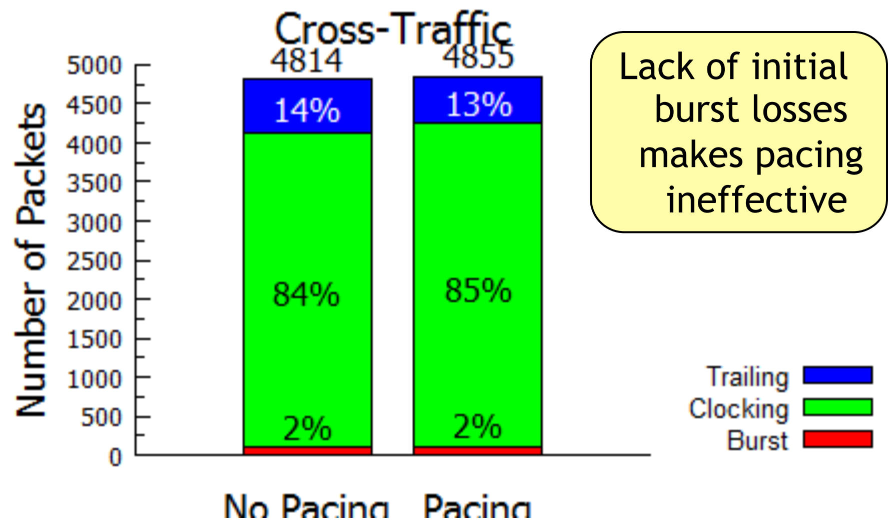
Poorly Paced Flow

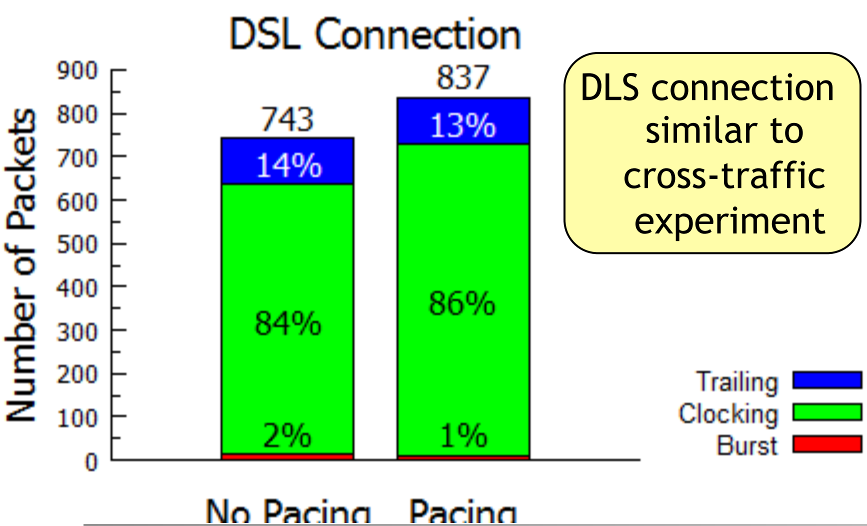


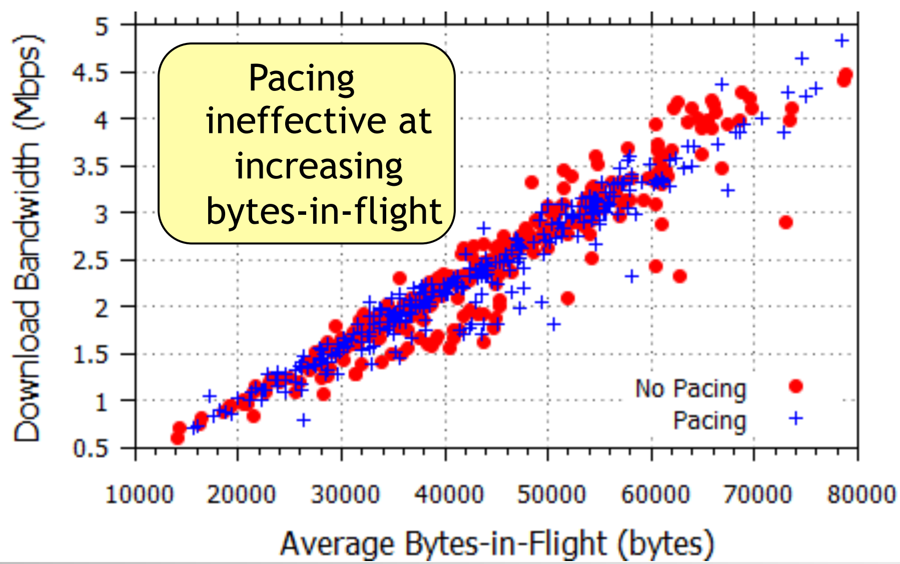
All Rights Reserved © Alcatel-Lucent 2012
Alcatel·Lucent

Pacing Performance









All Rights Reserved © Alcatel-Lucent 2012
Alcatel·Lucent

Summary and Conclusions

Summary

- Examined TCP traffic patterns created by HAS flows.
- Divided transmission into 3 phases: Initial Burst, ACK Clocking, Trailing ACKs.
- Examined effect of packet losses in each phase.
- Examined pacing as solution to increasing throughput for HAS.

Conclusions

- Location of packet losses matters.
- ACK Clocking most desirable phase for packet losses.
- Trailing ACK least desirable phase for packet losses.
- Pacing was ineffective at improving throughput for HAS in our experiments, reducing throughput when packet losses were shifted to the Trailing ACK phase.

Future Work

- Strategies to reduce the number and impact of trailing ACK losses.
- TCP modifications specific for HAS flows.
- Improved client rate determination algorithm to reduce TCP interactions.

All Rights Reserved © Alcatel-Lucent 2012