Protecting VoIP from TCP Traffic

Ilpo Järvinen

Department of Computer Science
University of Helsinki

Wibra Workshop

15th October 2012

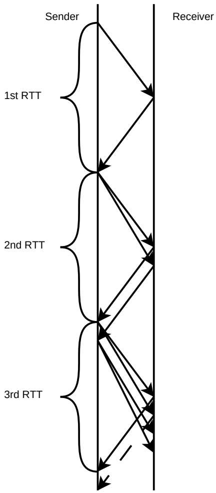
Outline

Introduction

- VoIP over empty HSPA link works reasonably well
- But not so well if TCP is competing with audio
- Most people know about the problems
  - Some are clever enough avoid using TCP while audio is used
  - But even that is not always possible (e.g, automatic software update in background)
- Could something more automated be used to mitigate problems?

TCP Congestion Control Basics

- State held in congestion window
  - Tells how much data can be outstanding in the network
- Initial probing using Slow Start with small initial congestion window (IW)
  - Congestion window grows exponentially with factor of 1.5-2 per round-trip time (RTT)
  - Actual growth rate depends on advanced TCP features such as Delayed ACKs, Appropriate Byte Counting (ABC), Initial window, etc.
- Continue increasing sending rate until losses occur, halve the congestion window (Multiplicative Decrease a.k.a. MD), and recover the lost packets . . .
- . . . continue in Congestion avoidance increasing window by one packet per RTT (Additive Increase a.k.a. AI)

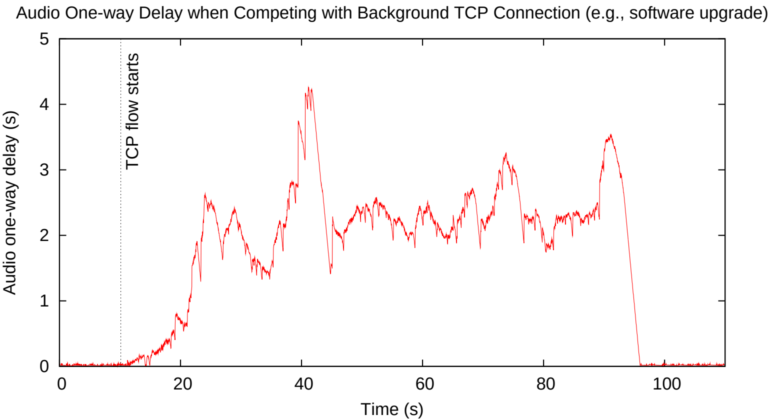


TCP Slow start

TCP and Buffers

- The link with least bandwidth on end-to-end path forms a bottleneck
  - In a common case close to the end-user, the access link or in the access network
  - When rate of incoming traffic exceeds the bottleneck bandwidth, packets pile up in bottleneck router buffer
- Buffers needed mainly for two reasons
  - Handling transient bursts
    - TCP Slow Start causes bursts (injects more packets than what goes through the bottleneck at the same time)
    - Network caused burstiness
  - Avoiding under-utilization after Multiplicative Decrease (MD)
- Right buffer size to avoid under-utilization after TCP MD
  - With one flow the buffer size needs to be roughly the bottleneck bandwidth times end-to-end RTT
  - With more flows, even less is enough as effect of a single flow MD is smaller

Background TCP Effect on Audio Delay over HSPA



- Audio one-way delays is 15ms-21ms (25th-75th percentiles) when no background traffic

Background TCP and Audio: Observations

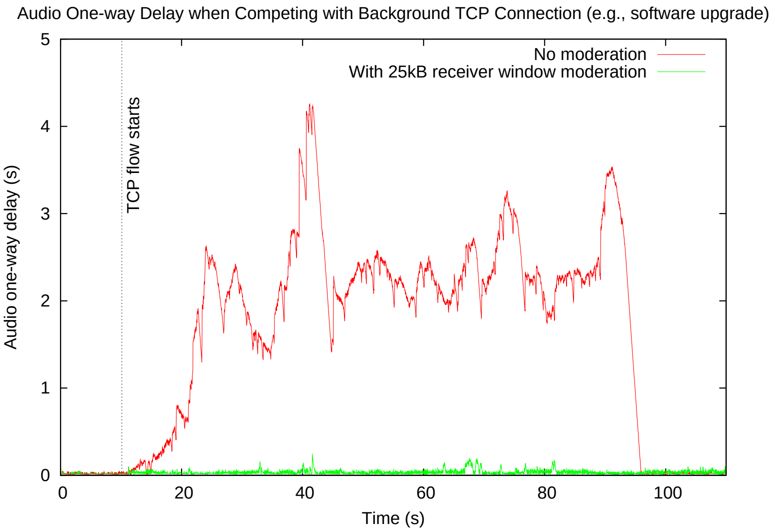
- HSPA link one-way bandwidth-delay product (BDP) around 3-13 pkts (2.7-5Mbps / 10-30ms)
- With 100ms end-to-end RTT the path BDP is 22-42 packets
- The measured buffer capacity 500+ packets
- TCP congestion control is designed to probe until losses occur
  - Without active queue management (AQM), TCP probes until the queue becomes full
  - First TCP Slow Start fills that 500+ packets buffer
  - Then, after TCP Multiplicative Decrease, 240+ packets still remain in the buffer
  - ...and TCP again proceeds to fill it up to 500+ again (and the process repeats)
- Audio is just an example, also other latency sensitive traffic has enormous problems (e.g. Web Traffic page completion time 10 times larger!)
- Can we do something?

TCP Receiver Window Moderation

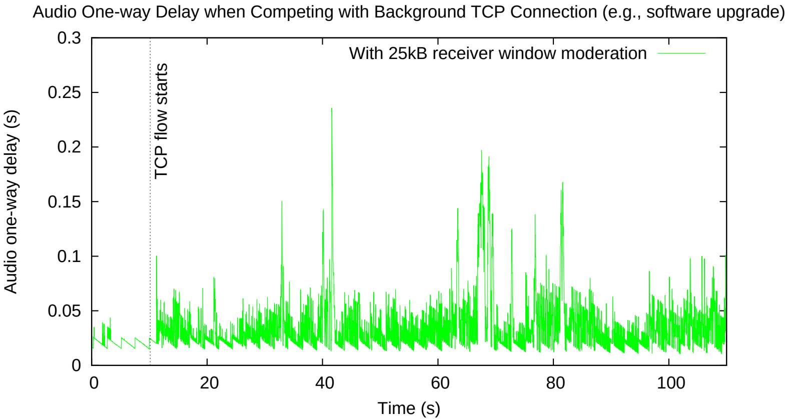
- TCP receiver has receiver advertized window (RWND) for flow control purposes
  - We rig it to limit the sender
- TCP sender is allowed up to minimum of congestion window and advertized window worth of packets outstanding
- Different from the usual TCP window capping approaches that typically occur within a TCP flow (such as implemented in Androids, iPhones, [1], etc.)
  - These approaches tend to cause standing queue
- In our approach the limit split between flows
  - Otherwise concurrent Web traffic flows would cause overcommitment

[1] Understanding Bufferbloat in Cellular Networks, CellNet 2012

Background TCP over HSPA with RWND Moderation



Background TCP with RWND Moderation (zoomed view)

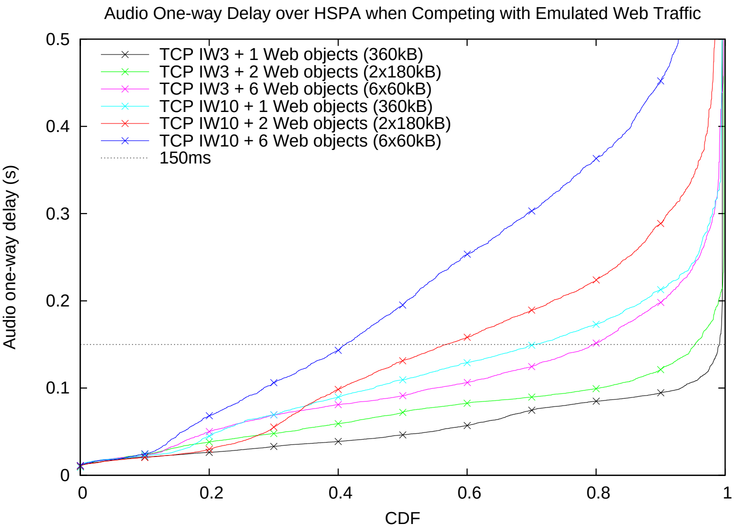


- Possible explanation for the spikes: link-level retransmissions (?)

Audio Meets Web Traffic

- Emulated Web transfers with 1, 2, and 6 parallel TCP connections
- TCP using initial window of 3 (IW3) and 10 (IW10) were tested

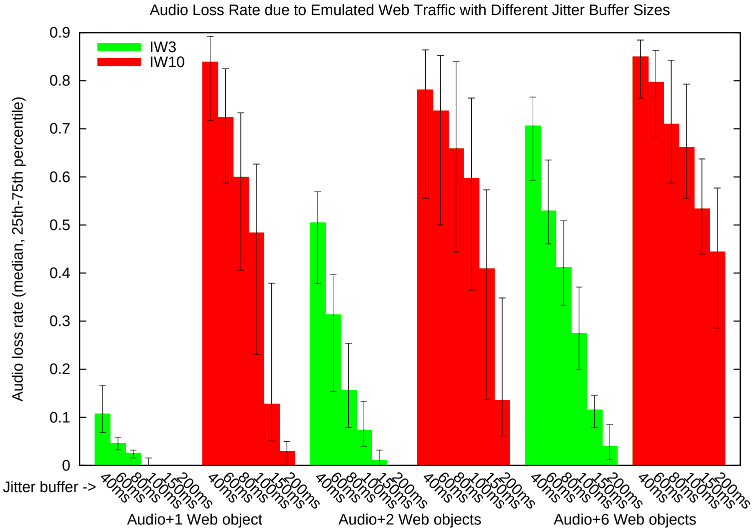
Audio One-way Delays with Emulated Web Traffic



Interactive Media, Codecs, and Jitter

- Interactive media needs to be played timely
- Codec is prepared to absorb some amount of jitter (delay variations in the packet end-to-end delay)
  - But playback sets a hard deadline
  - Packet arriving after playback deadline cannot be used, similar to loss
- Delay spikes can delay consecutive packets
  - Codecs can only conceal limited number of losses in a row

Audio Loss Effects with Different Jitter Buffer Sizes



Audio with Baseline TCP vs RWND Moderation

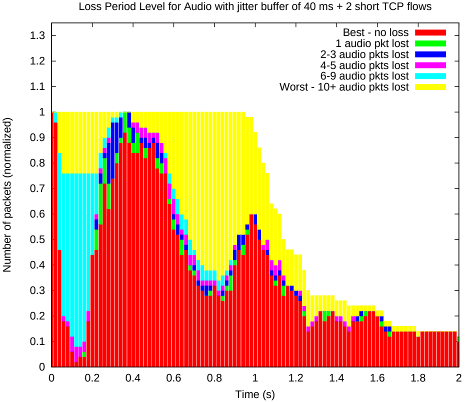


Figure: Audio with 40ms jitter buffer + 2 concurrent Web objects, no moderation

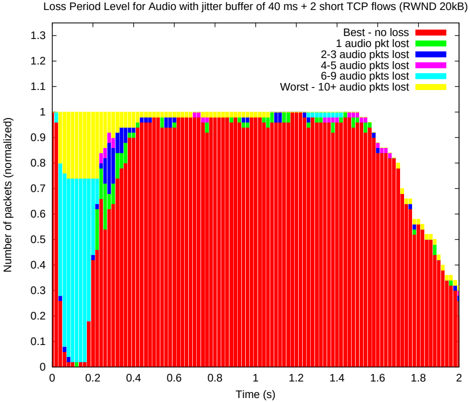


Figure: Audio with 40ms jitter buffer + 2 concurrent Web objects, RWND moderated to 20kB

Audio with Baseline TCP vs RWND Moderation (2)

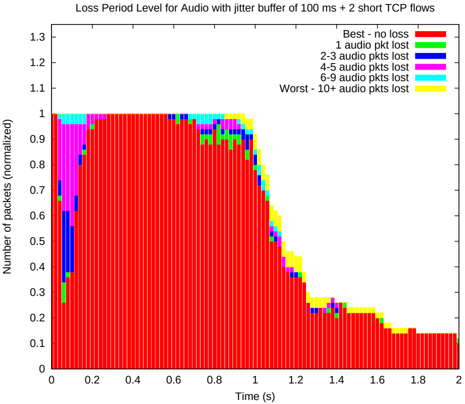


Figure: Audio with 100ms jitter buffer + 2 concurrent Web objects, no moderation

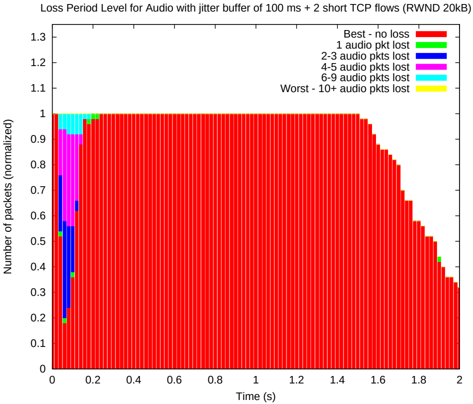
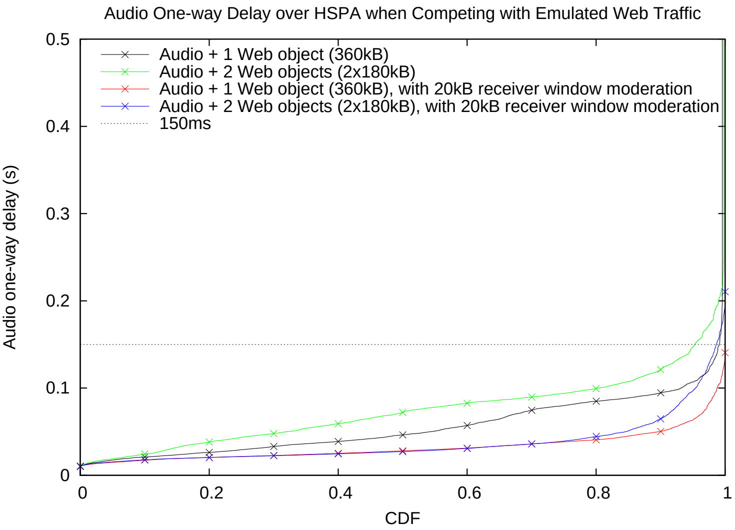
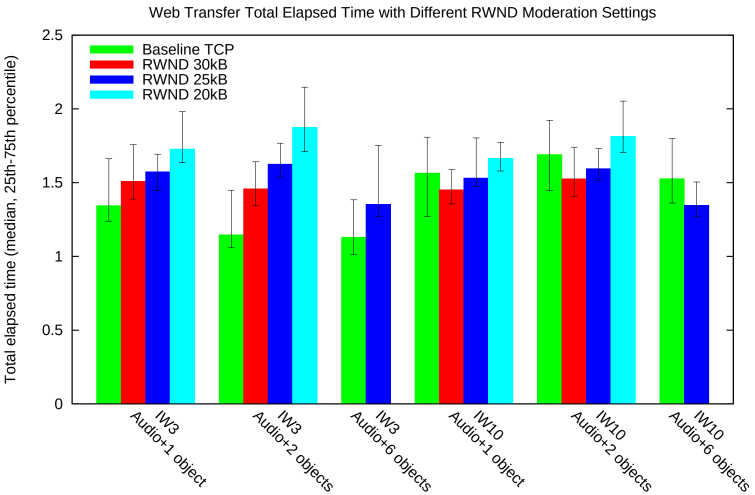


Figure: Audio with 100ms jitter buffer + 2 concurrent Web objects, RWND moderated to 20kB

Audio One-way Delay with RWND Moderation, Overview



Effect of RWND Moderation on TCP Performance



Conclusion

- Concurrent TCP traffic is harmful to interactive traffic (like VoIP)
- Presence of a long background TCP flow makes use of interactive media flow impossible
- TCP initial window size and large number of parallel flows with Web traffic contribute to the audio problems
  - Problem is getting worse with IW10 deployment
- The mobile end can use TCP receiver advertized window moderation to mitigate the problems
  - TCP IW burst still needs to be more carefully addressed
  - Slightly decreases TCP throughput, but the moderation does not entirely destroy TCP performance