University of Glasgow | School of Computing Science

Can Congestion-controlled Interactive Multimedia Traffic Co-exist with TCP?

Colin Perkins

Context: WebRTC

- WebRTC project has been driving interest in congestion control for interactive multimedia
- Aims to make interactive multimedia conferencing a native feature of web browsers – toolkit for peer-to-peer multimedia applications
  - Protocol standards under development in Internet Engineering Task Force (IETF)
  - Standard Javascript APIs being devised in World-wide Web Consortium (W3C)
- Strong involvement from Google, Mozilla, and Microsoft, amongst others – prototype code shipping now, wide deployment soon

Changing Environment

- Expect WebRTC to further shift Internet traffic mix towards latency sensitive traffic
  - Ubiquitous deployment in web browsers
  - Strong interest from web developers
  - High quality due to wide availability of HD cameras and encoders
- How will network and new applications interact?
  - Video coding – constraints, quality of experience, requirements
  - Network behaviour – misbehaviours, challenges
  - Transport protocols – behaviour, impact

Video Coding for Interactive Multimedia

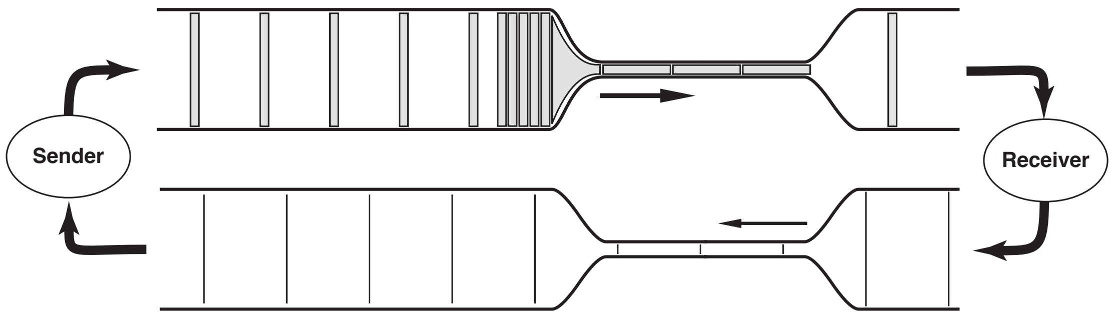
- Output of a modern video codec:
  - Can target a range of output rates, with fixed upper- and lower-bounds
    - kbps → Mbps depending on input picture size and frame-rate
    - Bit rate varies with content, not network availability – moderately bursty
  - Somewhat elastic – vary target coding rate
    - Output rate variable over (roughly) order-of-magnitude for given input
    - Constrained set of possible output rates
    - Constrained adaptation times – cannot change rate immediately
    - Some *limited* scope for channel-aware media coding – e.g., scheduled I-frames
- Interactive conferencing – QoE requirements
  - Critical to bound latency ~100s of milliseconds
    - ITU G.114 – quality of experience impacted ≥150ms mouth-to-ear latency
  - Predictable media *quality* highly desirable

Network

- Misbehaviours and challenges:
  - Congestion common at network edge – *some* times, *some* directions
  - Links often asymmetric and/or variable capacity
  - Over-buffering commonplace
  - Ubiquitous NAT
    - Perception of limited NAT resources by interactive multimedia community
    - Multiple flows multiplexed on a port (e.g., RTP, STUN, DTLS, SCTP/UDP) despite different requirements
- Cross traffic plentiful, potentially disruptive
  - Other interactive multimedia flows, web, file-sharing, IPTV, gaming, etc.

Competing Traffic – Impact on Latency

- TCP relies on packet loss as congestion signal
  - TCP probes for capacity by increasing window
  - Queues build at bottleneck – TCP dynamics rely on queues and in-network buffering
  - Queues smooth output onto bottleneck link



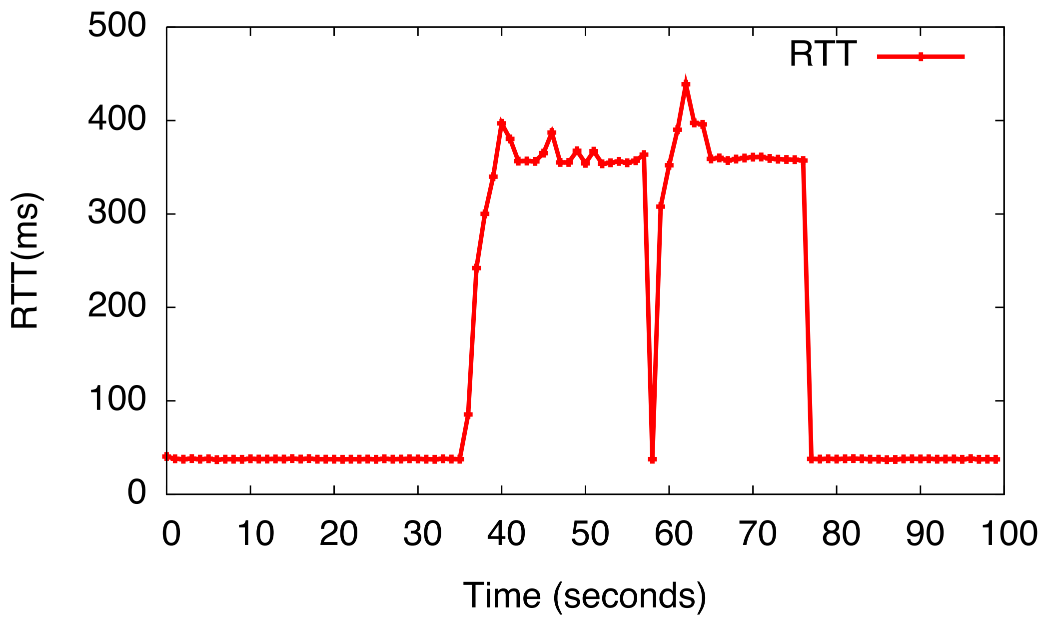
Source: Nichols & Jacobson, ACM Queue

- Long-lived TCP flows cause long-duration standing queues
  - e.g., MPEG DASH – in-network queues somewhat desirable
- Transients due to web browsers opening multiple connections encourage over-buffering
  - Impact of IW=10
- Network edge acts as if packets are precious; transport dynamics encourage this perception, yet latency is often bigger concern

Interactive Multimedia Transport

- Interactive multimedia flows avoid TCP, run over RTP/UDP/IP
  - Latency sensitive – prefer loss that can be concealed to latency that cannot
  - Not congestion controlled

- But, share queues with TCP flows
  - Example: ping times on ADSL link, measured in parallel with a bulk TCP upload
  - 10× RTT spike → problematic for interactive



- Network is optimised for throughput, not latency – interactive applications suffer

Challenges for Interactive Multimedia

- Can we implement congestion controlled transport for interactive multimedia on today's Internet?
  - To ensure safe deployment of WebRTC applications, with suitable delay bounds for interactive QoE

- Does the network provide an appropriate service model for the long term?

WebRTC Congestion Control Design Space

- Must work on today's Internet
- Must have significant delay-based component – to control latency
- Must work with limitations on when and how coding rate can adapt

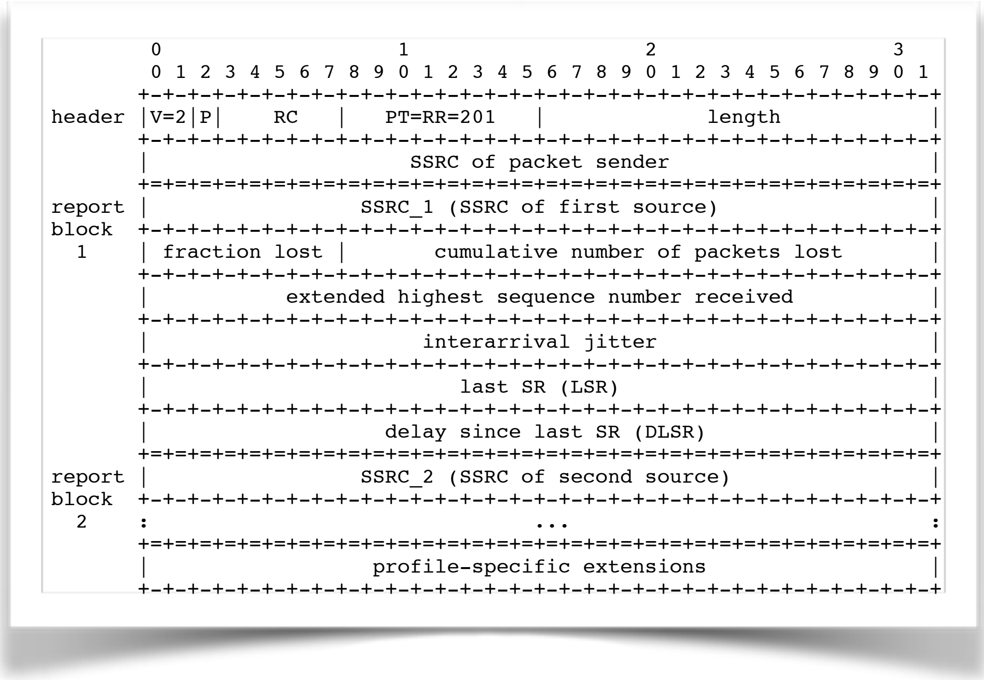
- Must it compete reasonably with cross traffic?

Type of Cross Traffic	Feasibility
Other interactive multimedia flows	Ok – delay based algorithm, fair competition
TCP flows – web browsing	Maybe – but don't overreact to loss bursts
TCP flows – long-lived	Not feasible due to queue build-up

- Must it be *fair* to TCP? Non-goal, sufficient to compete well with itself

Interactive Multimedia Transport: Basics

- Media runs over RTP
  - Framing, sequencing, and timing recovery
  - Supports wide range of codecs, loss tolerance tools
  - RTP Control Protocol (RTCP) – reception quality feedback and network management
- Low-rate feedback channel
  - Feedback packets large, infrequent; but can return semantic feedback (blocks *x* and *y* of media frame *z* were damaged)
  - Roughly every frame, not every packet
  - No explicit ACKs – continuous return traffic not guaranteed, so cannot piggyback
  - May need to re-think congestion feedback?

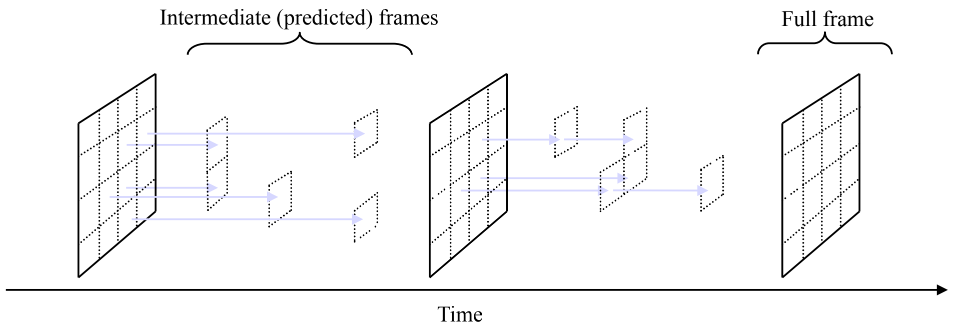


```
        0                   1                   2                   3
        0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
header |V=2|P|    RC   |   PT=RR=201   |             length            |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                     SSRC of packet sender                     |
       +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
report |                 SSRC_1 (SSRC of first source)                 |
block  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  1    | fraction lost |       cumulative number of packets lost       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |           extended highest sequence number received           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                      interarrival jitter                      |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                         last SR (LSR)                         |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                   delay since last SR (DLSR)                  |
       +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
report |                 SSRC_2 (SSRC of second source)                |
block  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  2    :                               ...                             :
       +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
       |                  profile-specific extensions                  |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Cross-flow Interactions

- How do interactive multimedia flows interact?
  - Control interactions between flows in a multimedia session – e.g., may want to prioritise audio over video, rather than fairness
  - Want reasonable behaviour when several users make video calls at once

- Predictive coding → bursts
  - Try to avoid synchronised I-frames – limits responsiveness to delay spikes

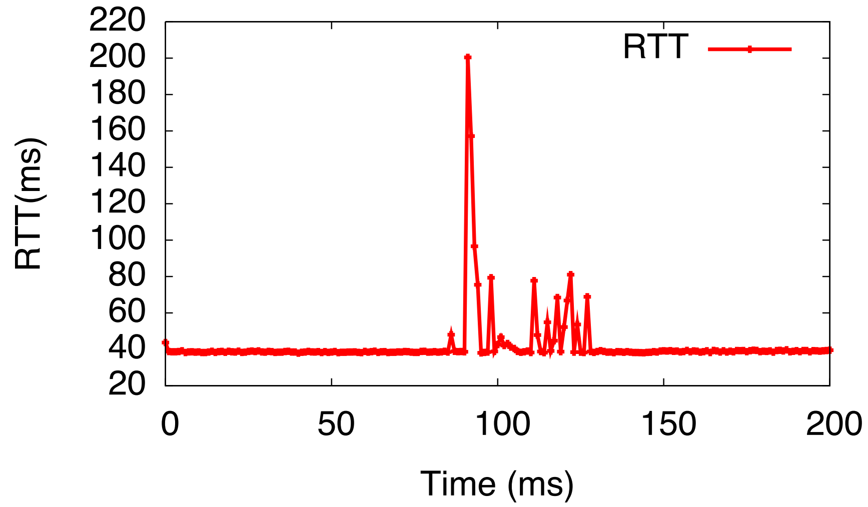


- Coupled congestion state across flows? Inter-flow prioritisation?

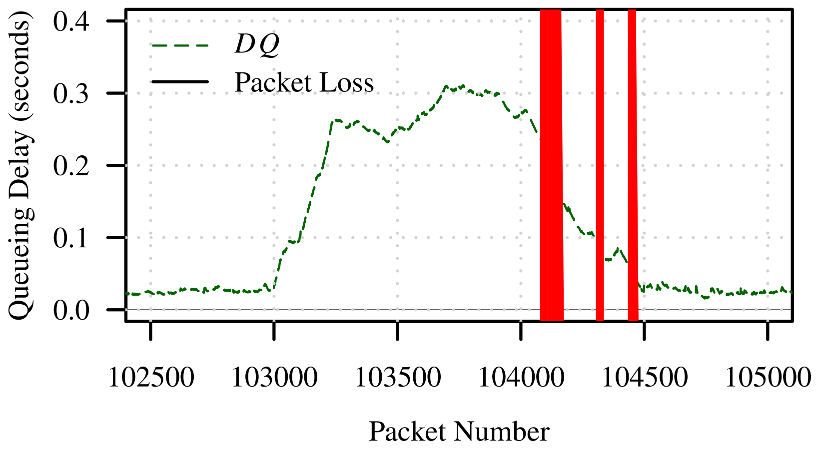
Interactions With Web Traffic

- Many short-lived parallel TCP connections
  - Average page ~1.2Mbytes, 87 requests
    http://www.httparchive.org/trends.php

- Short-term delay spikes
  - Effective jitter buffer needed – several frames delay; noticeable visual impact
  - Delay-based congestion control needs filter → less responsive but maintains usable quality



- Complex interaction with buffer bloat
  - Increased delay for several seconds; loss bursts
  - Interactive application unusable for the duration – pause rather than rate adapt?



- QoE constraints impact congestion control

Interactions with Long-lived TCP Flows

- Acceptable latency bound for interactivity is unclear
- What cut-off for congestion control is appropriate?
  - What latency causes a call to be abandoned? ITU G.114 suggests soft limit of 400ms for “network planning purposes” – is this appropriate?
  - For how long must the threshold be exceeded?
- Should interactive multimedia flows ‘fight back’ against latency insensitive flows?
  - Algorithms that switch to a more aggressive loss-based mode when delay-based mode isn’t sustaining minimum codec rate?

Next Steps for WebRTC

- IETF working group on interactive multimedia congestion control
  - http://datatracker.ietf.org/wg/rmcat/charter/
  - Desirable properties of resulting algorithm:
    - Hybrid loss-delay algorithm – but, delay as primary metric
    - Aware of user experience, codec constraints
    - Semantic feedback, working with protocol constraints
    - Coupled, cross-flow, congestion control

  - Tight charter and milestones – three algorithms proposed currently, all delay-based – unclear other properties met
  - Some months remain to influence algorithm before initial deployment; post-deployment browser updates likely rapid – Internet-scale testbed

I E T F®

Thoughts on Longer-Term Goals

- Want to prevent interactive traffic sharing a queue with loss-based congestion control

- Directions
  - Active queue management (AQM)
  - Alternate congestion control

Active Queue Management

- TCP needs standing queues, but latency-sensitive traffic must avoid queues → AQM
  - Priority queueing – separate queues for interactive and latency-insensitive
    - Better to prioritise traffic based on importance → DiffServ
    - Politics, economics, and network neutrality
  - Queue management to control latency
    - RED-like mechanisms can bound delay, if tuned correctly
    - Is there a deployable auto-tuning AQM solution – CoDel?
    - ECN interacts well with multimedia – allows application control of quality impairment (especially important at low data rates, with infrequent I-frames)
- Campaign against buffer bloat also essential

Revising TCP Congestion Control?

- Queues build because TCP tries to fill them – could we change TCP?
  - Algorithms using delay as input: TCP Vegas → FAST TCP
  - Deployment getting easier with automatic software updates – we *can* update edge devices
  - Would it help enough? Competing with loss-based TCP → queues
  - Incremental path to deployment? An algorithm that's delay-based by default, but reacts to loss and becomes more aggressive to prevent starvation when competing with TCP?

Conclusions

- Mismatch between application needs and network:
  - Interactive multimedia does not effectively co-exist with TCP in today's network
  - Desirable to reduce buffer bloat, manage queues, and mark packets to signal congestion

- Community input to standards process desirable
  - Congestion control; active queue management