Multimedia Transport Protocols for WebRTC

Colin Perkins

http://csperkins.org/
What is WebRTC?

A framework for browser-based real-time conferencing
Includes network, audio, and video components used in voice and video chat
Accessed through Javascript API to support custom applications
(e.g., implement Google hangouts as native HTML5 application)
Signalling infrastructure

Real-time multimedia transport  NAT traversal

Peer-to-peer data
Web Server

Application download

WebRTC

Application download

WebRTC
Application provider

- JavaScript application delivered from web server
- Identity provider
- NAT traversal infrastructure (STUN/TURN)

Signalling

- Application protocol defined via Javascript API
- Offer-answer exchange of SDP (JSEP)

Media and data transport

- NAT traversal: STUN with fallback to TURN
- Media transport using secure RTP over UDP
- Secure peer-to-peer data using SCTP over DTLS/UDP
Benefits

- Standard infrastructure requirements
- Standard browser API
- Flexible application support
- Modern media codecs and transport protocols
- Peer-to-peer data channel
Challenges

API complexity vs completeness/control
NAT traversal performance
Security and identity
Media transport and congestion control

Benefits

Standard infrastructure requirements
Standard browser API
Flexible application support
Modern media codecs and transport protocols
Peer-to-peer data channel
Media transport and congestion control

What is the problem?
Why not TCP?
What are the challenges?
Directions and solutions
Media transport and congestion control

What is the problem?

Why not TCP?

What are the challenges?

Directions and solutions

Real-time media transport that is safely deployable and high-performance

Baseline RTP media transport is well defined
  › but will be deployed at very large scale
  › using modern high-rate video codecs
  › with no professional network support

Concern about potential network congestion
  › applications can use significant bandwidth, are trivial to deploy, and difficult to control
  › no appropriate congestion control algorithms
Media transport and congestion control

What is the problem?

**Why not TCP?**

What are the challenges?

Directions and solutions

Video streaming uses TCP – but high latency, and poorly suited for interactive real-time applications

TCP congestion control algorithm causes latency

- Loss-driven – relies on queue overflow
- Needs buffer to smooth abrupt changes in rate and match codec output

Retransmission with head-of-line blocking on loss

- Further latency
Media transport and congestion control

What is the problem?
Why not TCP?

What are the challenges?

Directions and solutions

Need an alternative to TCP congestion control
  › suitable for interactive multimedia

Avoid TCP-induced latency
  › for media flows
  › due to TCP cross traffic

Latency is critical; maximising throughput less so
  › media traffic has rate bounds
Media transport and congestion control

What is the problem?
Why not TCP?
What are the challenges?

Directions and solutions

Exploring three directions
- RTP circuit breaker
- Media congestion control algorithms
- Active queue management
RTP circuit breaker
How to stop errant media flows?

RTP circuit breaker
How to stop errant media flows?

Build on RTP reception quality feedback

- Media quality unusable
- Media and/or feedback timeout
- Congestion: 10× TCP throughput

\[ T = \frac{s}{R\sqrt{\frac{2p}{3}} + (t_{RTO}(3\sqrt{\frac{3p}{8}}p(1 + 32p^2)))} \]

Protects network on multi-second timescale

- Three reporting intervals
- Cease transmission or back-off 10×

Works in parallel with congestion control

- Last resort to protect network
- Does not address latency concerns


congestion control
How to adapt media to network capacity?

congestion control
How to adapt media to network capacity?

Use delay as congestion signal
- increased delay/inter-packet spacing → congestion
- Avoid standing queues in routers and packet loss
- Evolution of ideas in TCP Vegas

Google proposal draft-alvestrand-rmcat-congestion-02
- Congestion signal: filtered inter-arrival time
- Deployed in Chrome

Cisco proposal draft-zhu-rmcat-nada-03
- Congestion signal: filtered one-way delay
- Natively incorporates ECN feedback

Active discussion in IETF RMCAT working group
- Google proposal has stability issues
- Cisco proposal requires synchronised clocks
- Both could develop into reasonable protocols
Avoid standing queues in routers
Active queue management (AQM)

Can we separate media traffic from TCP?
- Delay-based congestion control will lose to TCP
- Can switch to loss-based mode, but will lose on latency

AQM gives media traffic segregated queue
- CoDel, PIE, etc.
- Latency benefit, irrespective of cross traffic
- Deployment concerns

Avoid standing queues in routers

K. Nichols and V. Jacobson. Controlling queue delay. ACM Queue, 10(5), May 2012
WebRTC deployment is starting
  ‣ Chrome and Firefox
  ‣ Increasing developer interest

Implications for network operators
  ‣ Increasing peer-to-peer UDP media and data flows
  ‣ Protected via RTP circuit breaker
  ‣ Evolving congestion control story

Implications for research
  ‣ Interactive multimedia congestion control an open issue
  ‣ Need to understand network characteristics
  ‣ Need to understand performance of RTP circuit breaker
  ‣ Need to understand performance of AQM