Performance Evaluation of WebRTC-based Video Conferencing

Delft University of Technology

- Bart Jansen
- Fernando Kuipers

Columbia University

- Timothy Goodwin
- Varun Gupta
- Gil Zussman

IFIP WG 7.3 Performance 2017

Bart Jansen

WebRTC

- Web Real Time Communications
- Audio, Video and Data communication
- Standard by W3C and IETF since 2012

- Benefits
  - Plugin less
  - Peer to peer
  - Cross browser/platform
  - Easy to use (API)

Columbia University

TU Delft

Research motivations

- WebRTC is a new emerging technology
  - Easy to use
  - Adobe Flash dying out
- Fast changing protocol under active development
- Study WebRTC's performance for both simulated and real world conditions

COLUMBIA UNIVERSITY

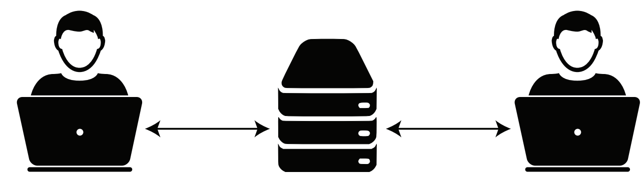
TUDelft

WebRTC – Topologies

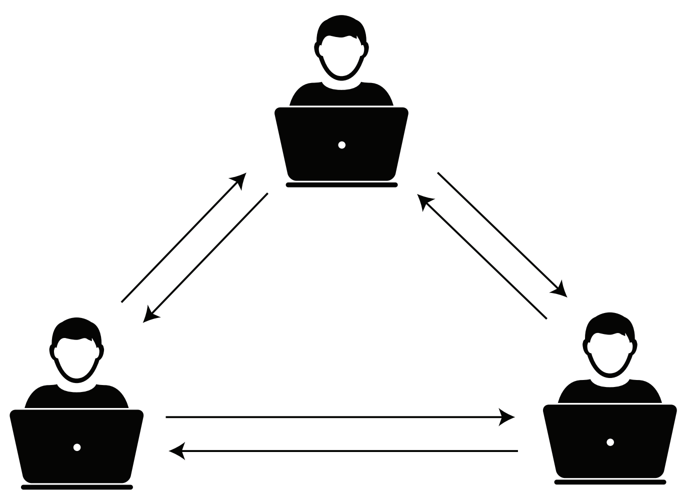
The way network nodes are arranged in a network

Peer to Peer

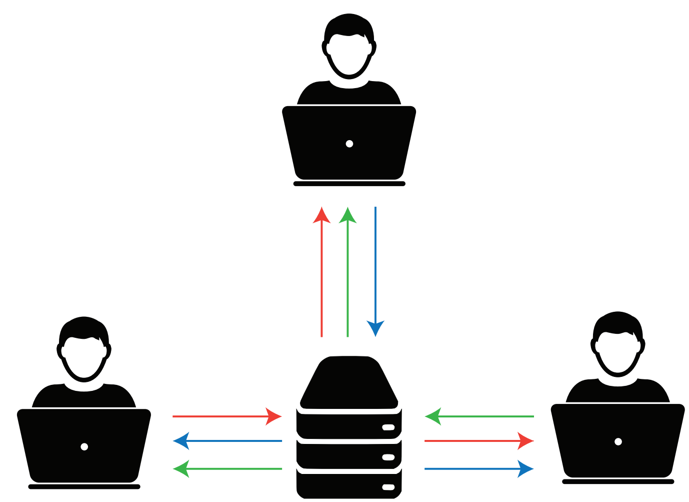
Server to Client



Meshed



SFU



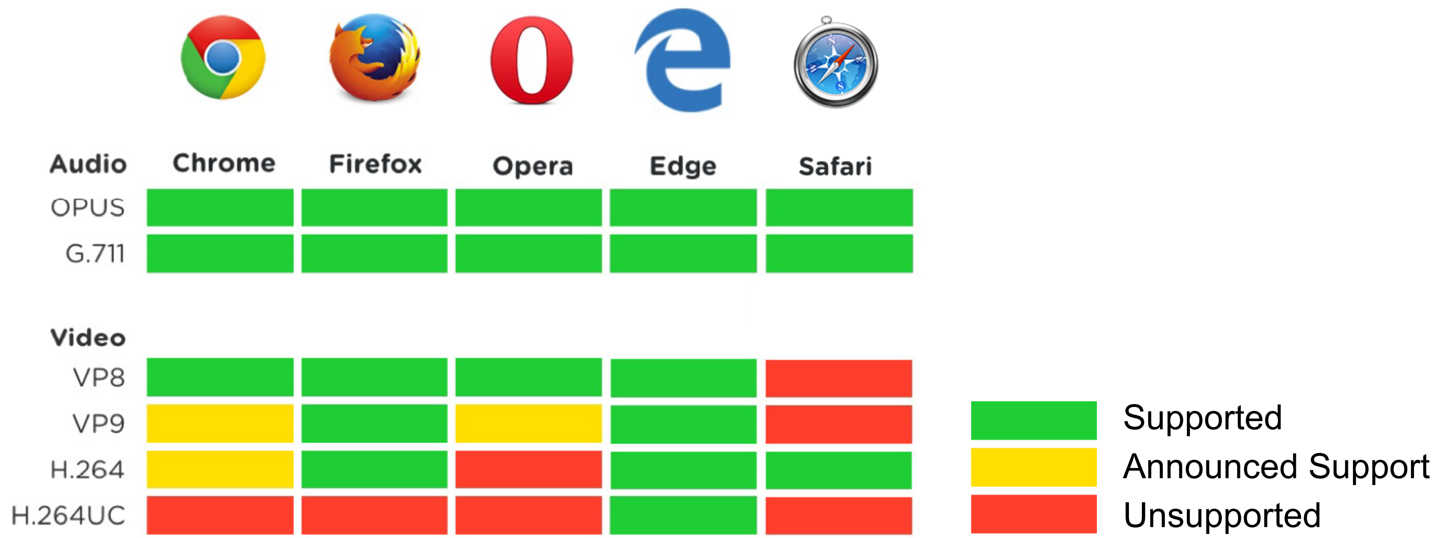
COLUMBIA UNIVERSITY
TUDelft

WebRTC – Interoperability

- Adoption

	2012	2013	2014	2015	2016
Desktop	2,01%	22,06%	30,90%	36,53%	68,51%
Mobile and Tablet	0%	1,75%	12,32%	28,94%	41,59%

- Media codecs



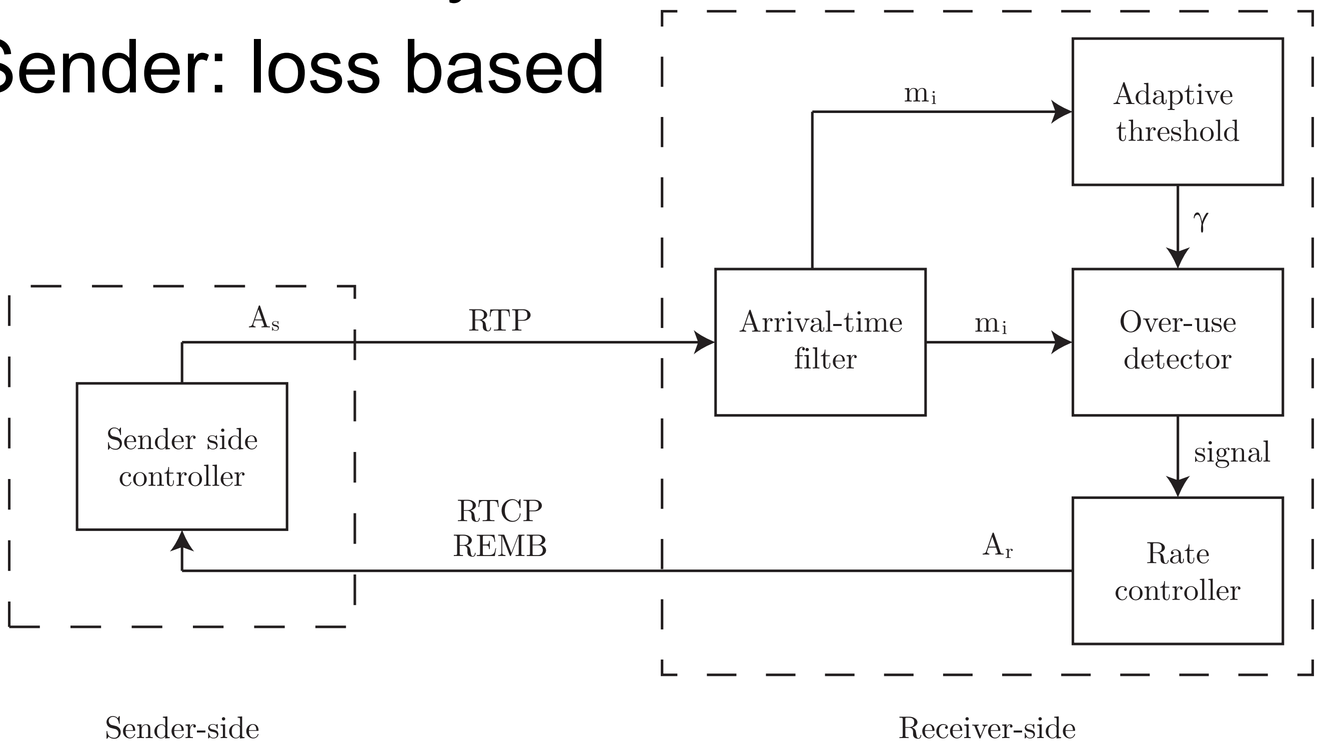
- Browser compatibility
  - Support for older browsers with plugin

Congestion Control

Congestion occurs when a node carries more data than it can handle

- Google Congestion Control Algorithm
  - Dynamically adjusts send and receive rate

- Receiver: delay based
- Sender: loss based



Congestion Control

- Receiver side (delay based):

$$A_r(i) = \begin{cases} \eta A_r(i-1) & \text{Increase} \\ \alpha R(i) & \text{Decrease} \\ A_r(i-1) & \text{Hold} \end{cases} \qquad \eta = 1.05 \qquad \alpha = 0.85$$

- Sender side (packet loss based):

$$A_s(i) = \begin{cases} A_s(i-1)(1-0.5 f_l(i)) & f_l(i) > 0.1 \\ 1.05 A_s(i-1) & f_l(i) < 0.02 \\ A_s(i-1) & \text{otherwise} \end{cases}$$

- Send rate never exceeds receive rate:

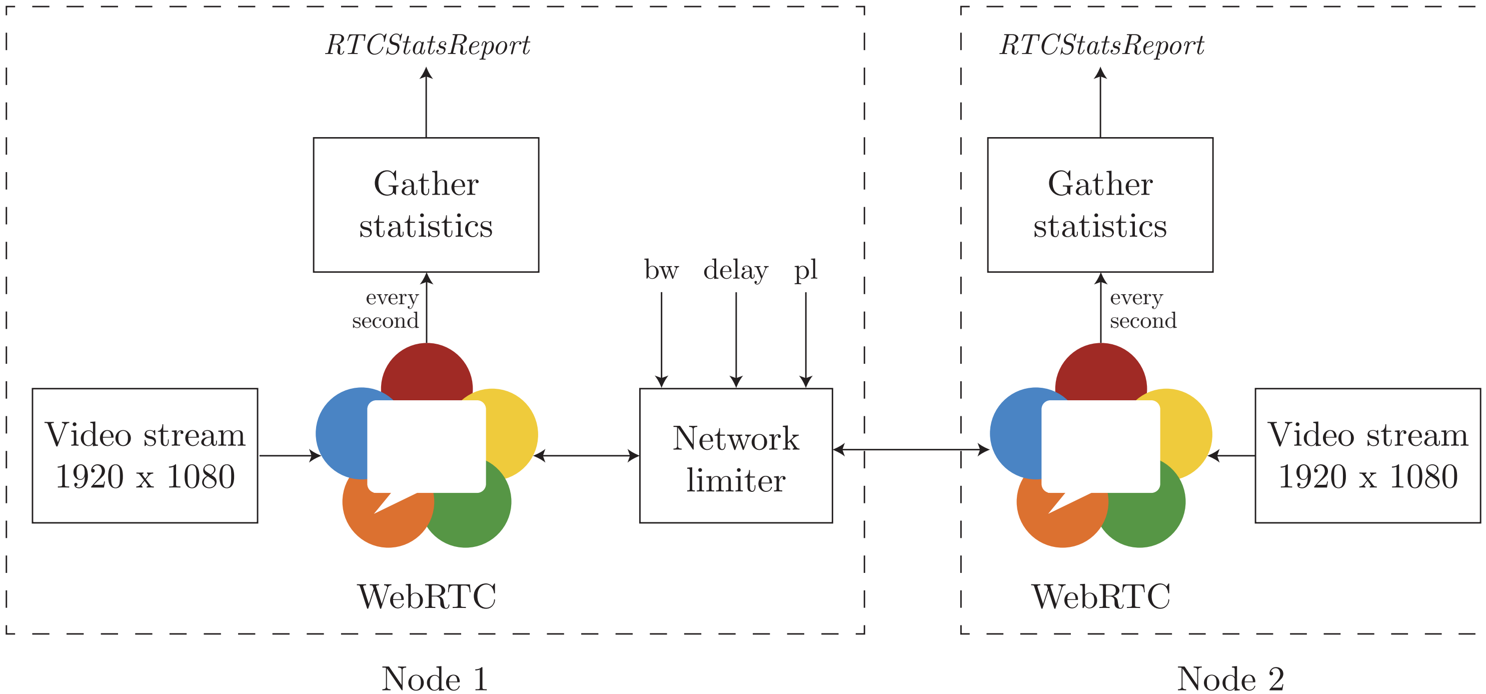
$$A_s(i) = \min(A_s(i), A_r(k))$$

COLUMBIA UNIVERSITY

TU Delft

Experimental test setup

Synthetic network conditions



- Custom video stream
- Network limiter
- Statistics via RTCStatsReport

Experimental test setup

- Avoid CPU and memory limitations
- 5 tests averaged

- Statistics:
  – Data rate (kbps)
  – Framerate (fps)
  – Resolution (width * height) (pixels)
  – Round-trip-time (ms)

- Source code available at: https://github.com/Wimnet/webrtc_performance

COLUMBIA UNIVERSITY
TUDelft

Performance Evaluation

- Baseline experiments
- Cross traffic
- Multi-party topologies
- Video codecs
- Mobile browsers
- Real wireless networks

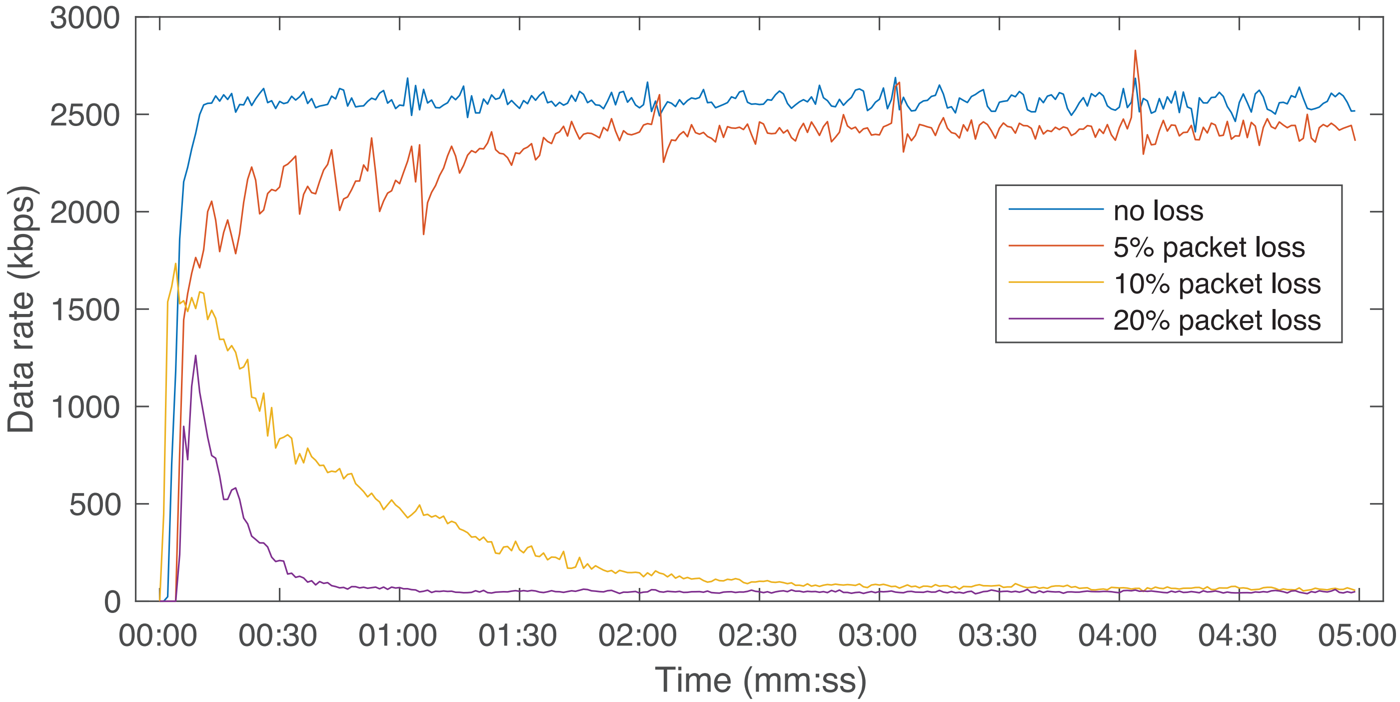
COLUMBIA UNIVERSITY

TU Delft

Performance Evaluation

Network characteristics – Packet loss

- As expected
  - 5% converges to maximum data rate
  - > 10% converge to minimum data rate



Columbia University

TU Delft

Performance Evaluation

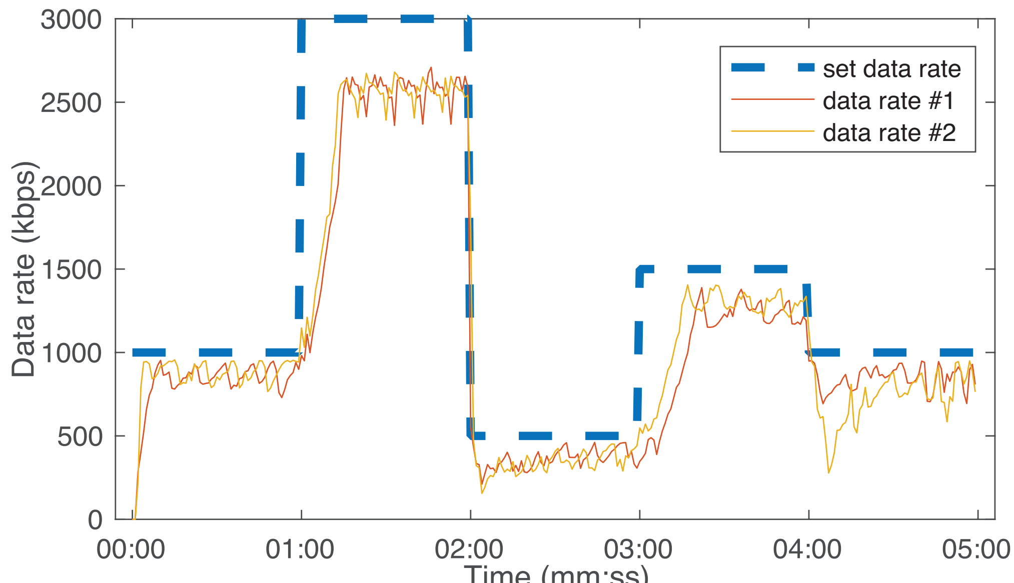
Network adaptability – Bandwidth

- 77% utilization
- Follows GCC rates after minute 1:

$$1000 * 1.05^{17} = 2300\ kbps$$

- Also after minute 3:

$$500 * 1.05^{18} = 1200\ kbps$$



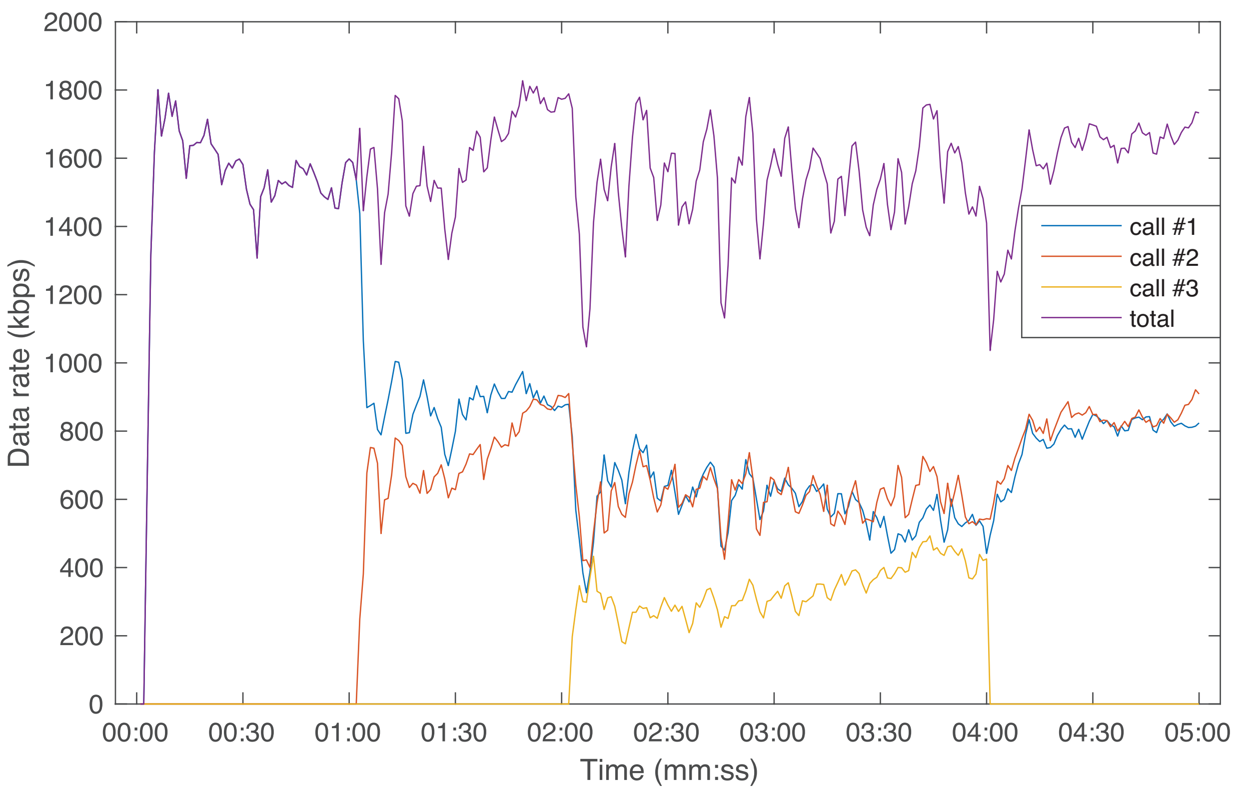
COLUMBIA UNIVERSITY

TU Delft

Performance Evaluation

Cross traffic – Intra-protocol fairness

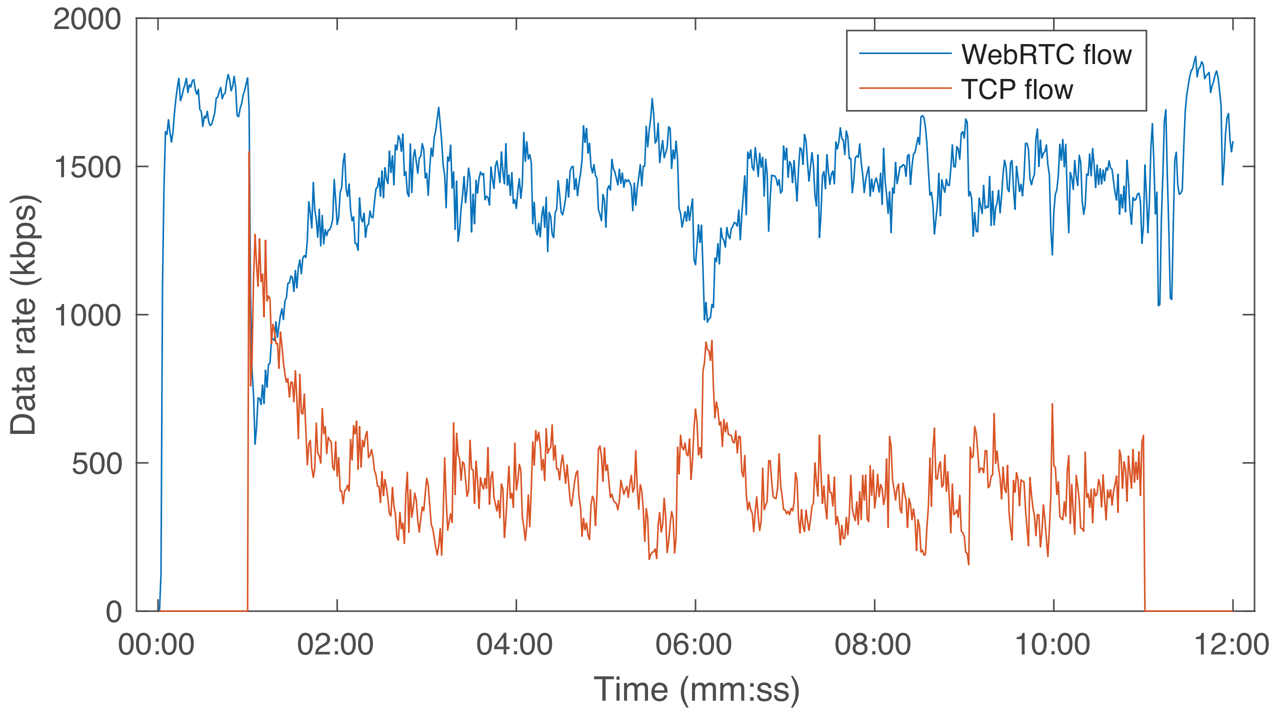
- Three separate calls with 2Mbps limit
- Fairness is reached with delay



Performance Evaluation

Cross traffic – Inter-protocol fairness

- 2Mbps limit
- Competing TCP flows
- Fairness can be improved



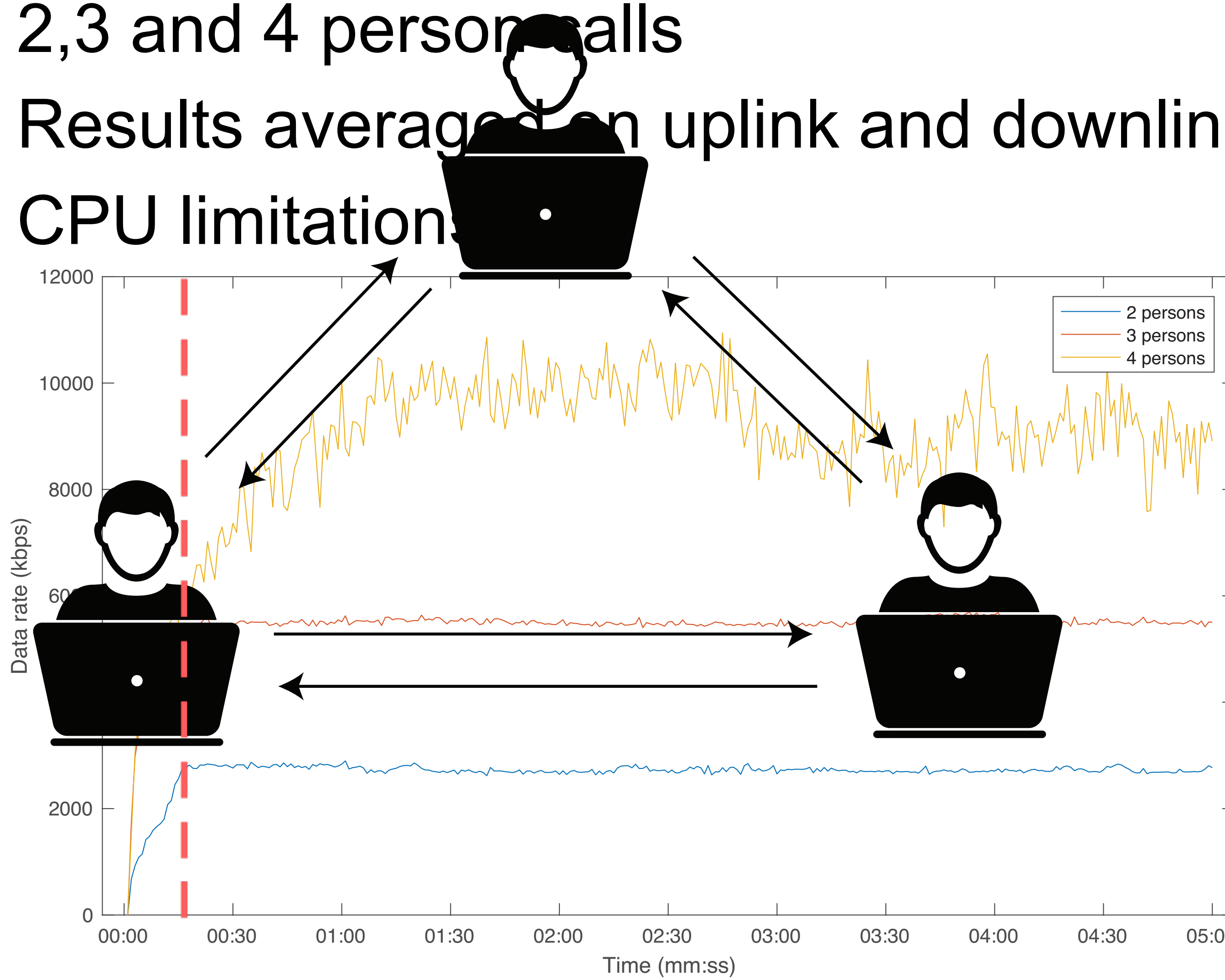
Columbia University

TU Delft

Performance Evaluation

Multi-party - Meshed

- 2,3 and 4 person calls
- Results averaged on uplink and downlink
- CPU limitations



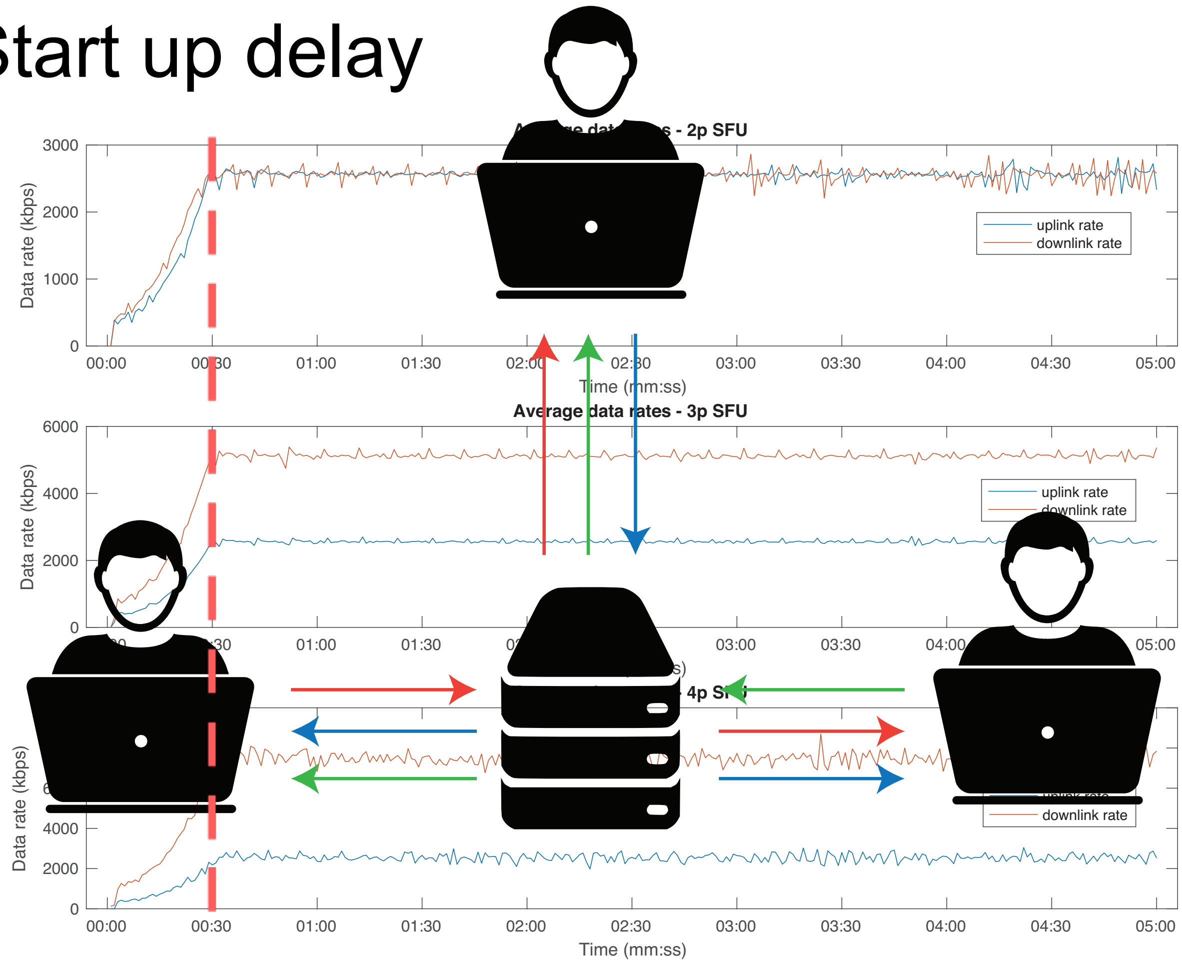
COLUMBIA UNIVERSITY

TU Delft

Performance Evaluation

Multi-party - SFU

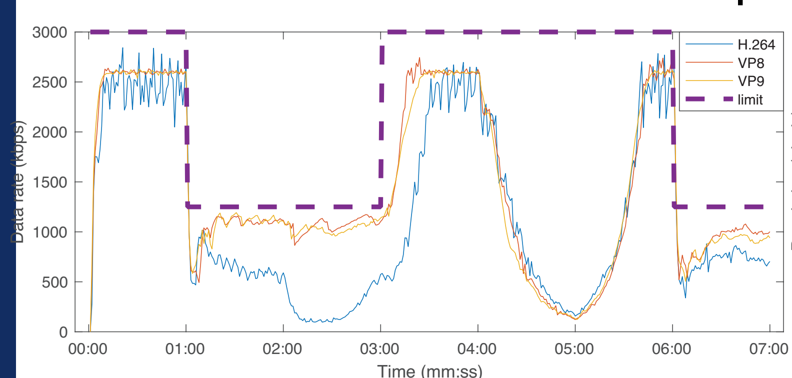
- Less uplink bandwidth required
- Start up delay

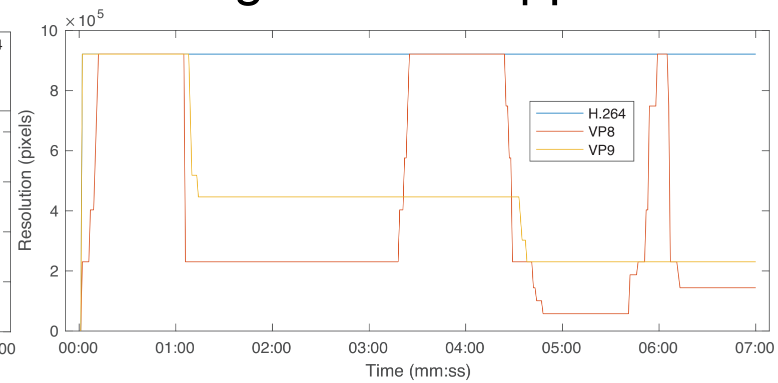


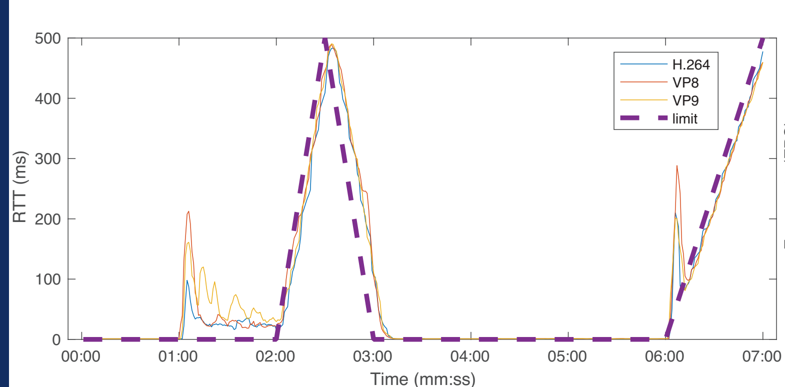
Performance Evaluation

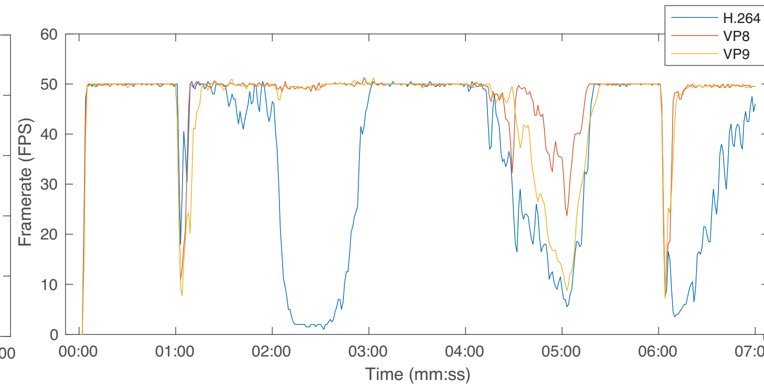
Video codec comparison

- VP8 (default), VP9 and H.264
- Room for improvement:
  - H.264 needs to balance between framerate and resolution
  - VP9 needs to scale up when congestion disappears









COLUMBIA UNIVERSITY

TU Delft

Experimental test setup
Wireless performance

- 3 nodes:
  - Local wireless node (NYC)
  - Local wired node (NYC)
  - Remote wired nodes (Oregon or Sydney)

- Varying:
  - AP transmission power (to 1mW)
  - Distance from AP (5ft – 25ft)
  - MAC retry limit

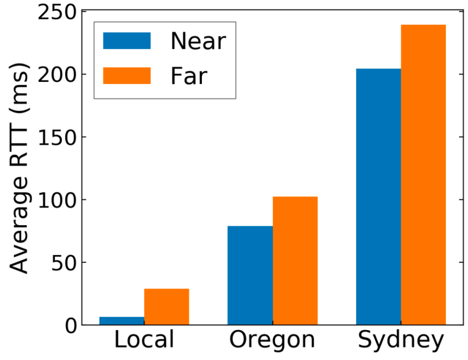
Columbia University

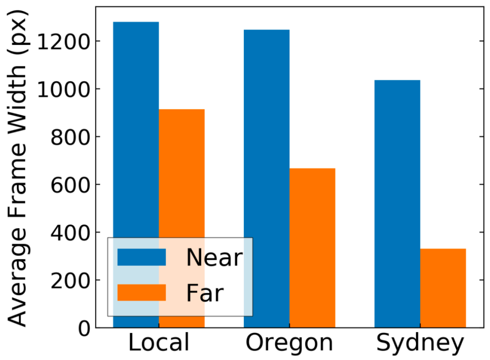
TU Delft

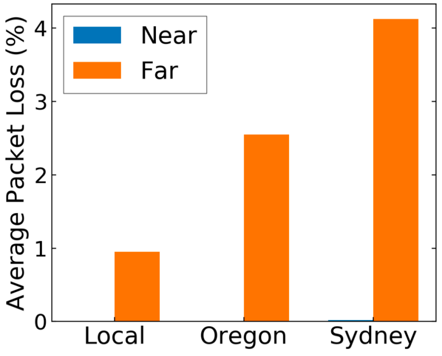
Performance Evaluation
Wireless network conditions

- 5ft vs 25ft AP distance

- Results
  – Higher RTTs results in more packet loss and lower quality







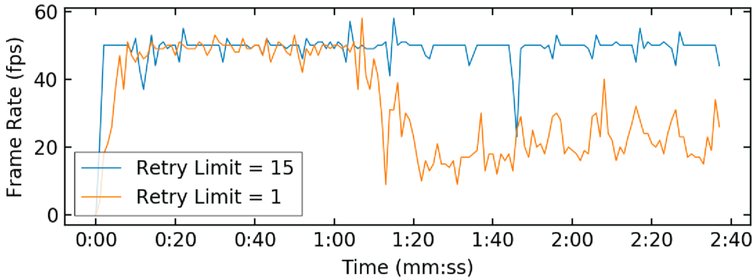
COLUMBIA UNIVERSITY

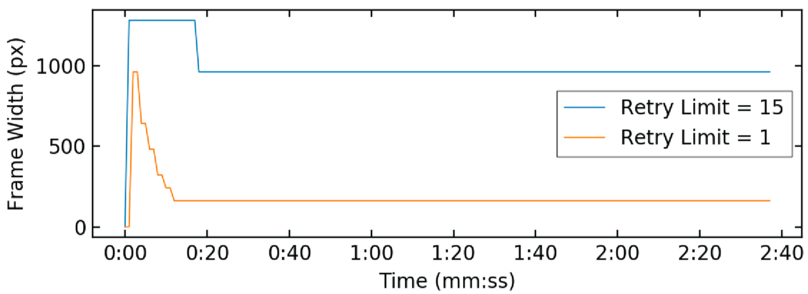
TUDelft

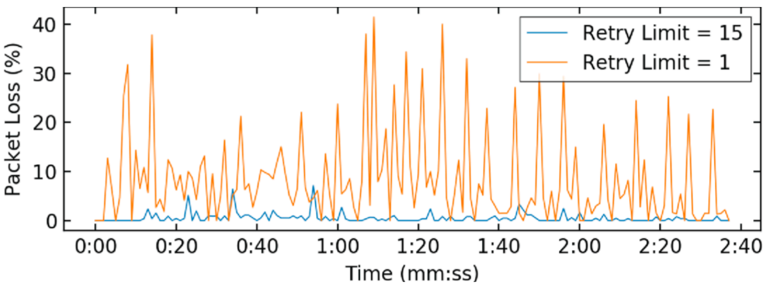
Performance Evaluation
Wireless network conditions

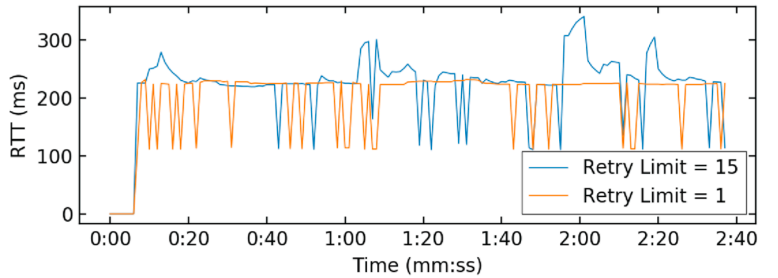
- Limit retransmissions in AP
  - From MAC retry limit 1 (min) and 15 (max)

- Results
  - Trade off between RTT and packet loss
  - Less video freezes
  - GCC relies too heavy on packet loss









Summary

- Improved browser interoperability
- Thorough evaluation of WebRTC
  - Custom video stream
  - Similar results
- Improved cross traffic fairness
- Poor performance over wireless due to:
  - Bursty losses
  - Packet retransmissions

Columbia University

TU Delft

Conclusions

- Heavily reliant on packet loss
- Multi-party
  - When more than 3 people: Add SFUs
- Room for improvement
  - Mobile performance
  - Video codecs

Columbia University
TU Delft

Future Work

- Take CPU limitations into account
  - Energy consumption (battery)
  - Call characteristics when CPU is exhausted
- Simulate Google Congestion Control
- Tests with cellular connections

COLUMBIA UNIVERSITY

TUDelft