Politecnico di Bari
Dipartimento di Ingegneria Elettrica e dell'Informazione

C3LAB
Control of Computing and Communication Systems Lab

Making Google Congestion Control Robust over Wi-Fi Networks Using Packet Grouping

Applied Networking Research Workshop 2016
Berlin, Germany, July 2016 (PDF)

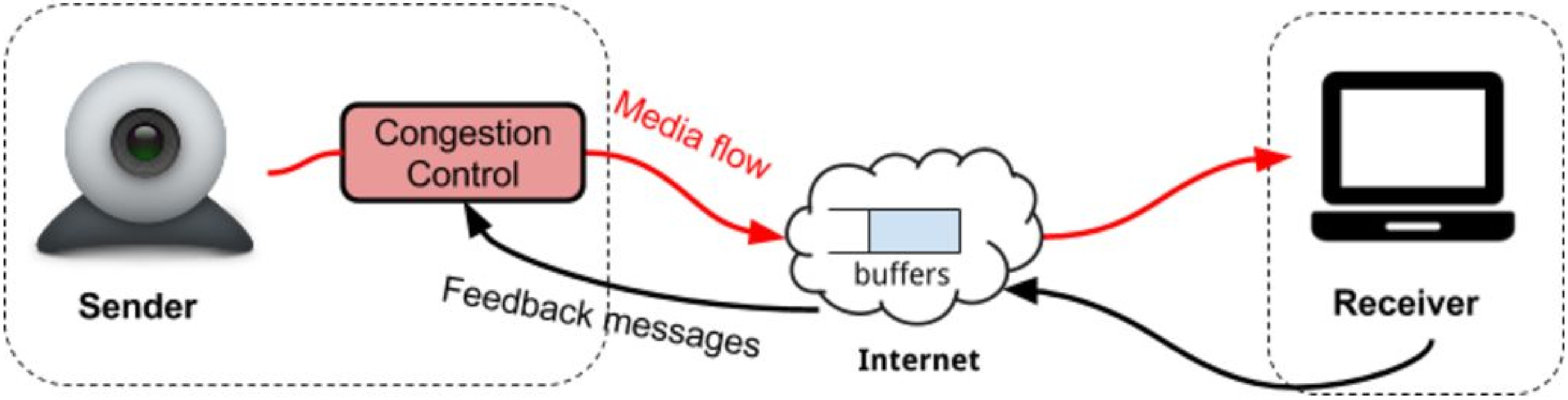
G. Carlucci, L. De Cicco, S. Holmer*, S. Mascolo

Politecnico di Bari, Italy, *Google Inc., Sweden

Context

- Real-time media communication requires not only congestion control, but also minimization of queuing delays to provide interactivity

- The design of such an algorithm is still an open issue



Context

RMCAT WG aims at defining a congestion control algorithm for real-time media flows sent using RTP over UDP (i.e. WEBRTC media flows).

Google Congestion Control (**GCC**) has been proposed to RMCAT WG as candidate algorithm draft-ietf-rmcat-gcc-01.

Other candidates:

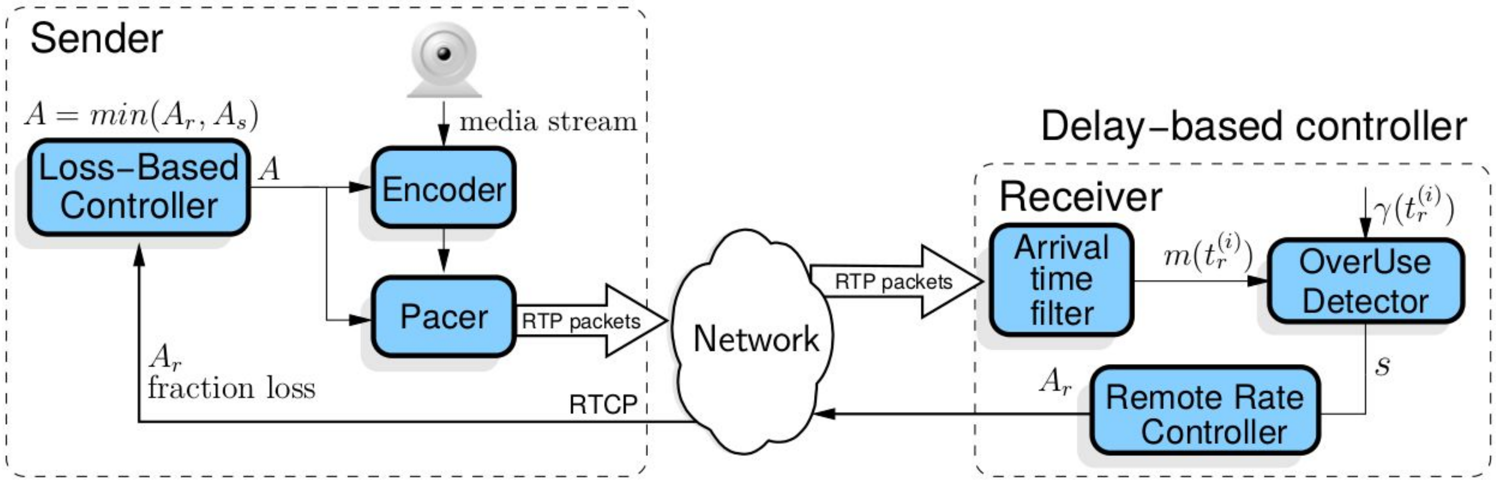
- **NADA:** “A unified congestion control scheme for real-time media”. *X. Zhu, R. Pan, S. Mena, P. Jones, J. Fu, S. D’Aronco, and C. Ganzhorn.* draft-ietf-rmcat-nada-02
- **SCREAM**: “Self-Clocked Rate Adaptation for Multimedia”. *I. Johansson and Z. Sarker.* draft-ietf-rmcat-scream-cc-05
- **SBD**: “Shared Bottleneck Detection for Coupled Congestion Control for RTP Media.” *D. Hayes, S. Ferlin, M. Welzl and K. Hiorth.* draft-ietf-rmcat-sbd-04

Congestion Control Requirements for RTC

- Contain queuing delays to improve interactivity
- Contain packet losses to avoid media quality degradation
- Reasonable fair sharing of the bandwidth with concurrent flows (both intra-protocol and inter-protocol)
- Prevent starvation when competing with loss-based TCP flows

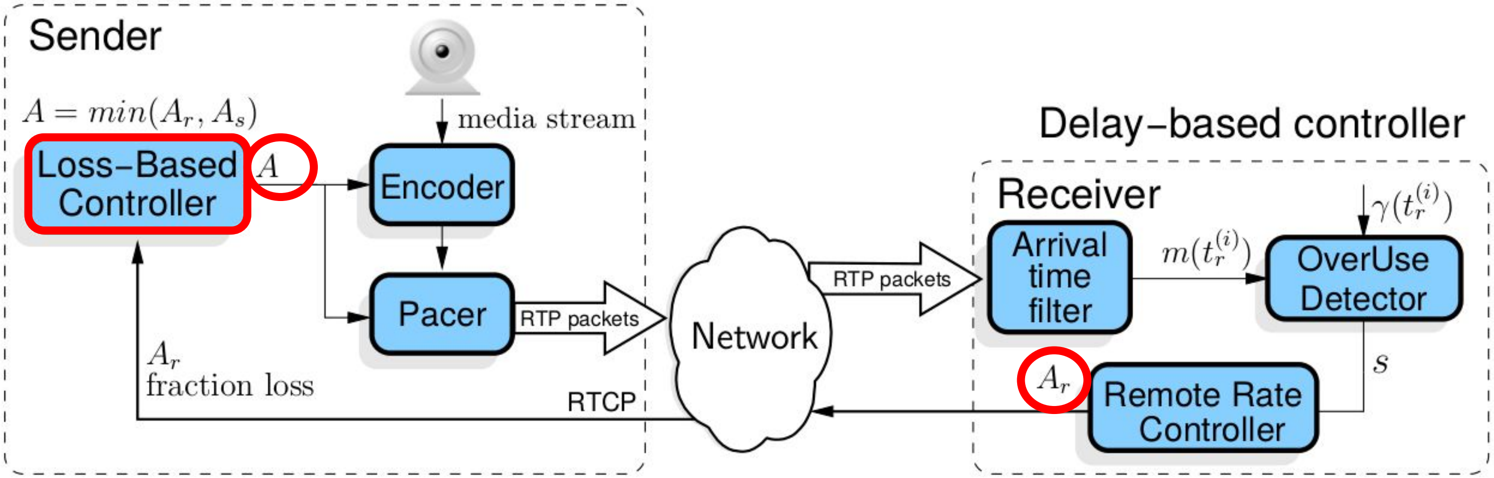
R. Jesup and Z. Sarker "Congestion Control Requirements for Interactive Real-Time Media" draft-ietf-rmcat-cc-requirements-09

Google Congestion Control



- Audio/video flows sent using RTP over UDP (feedback over RTCP)
- The delay-based controller aims at containing queuing delays
- Loss-based controller is used as a fallback

Sending Rate Computation

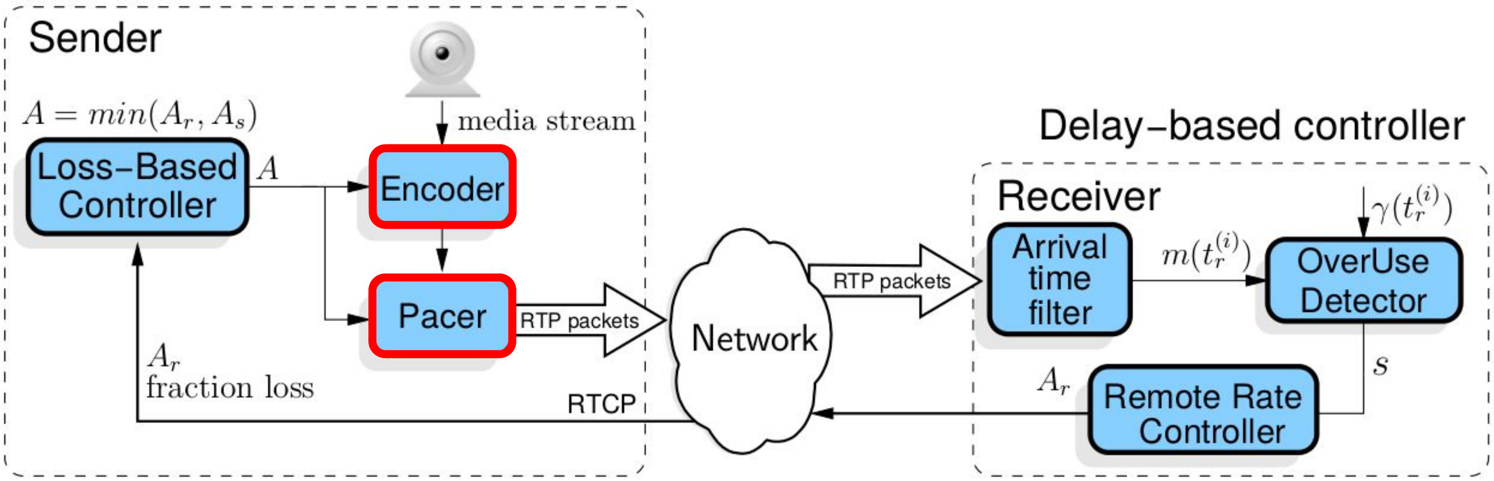


- The **Delay-Based** controller computes the rate A_r

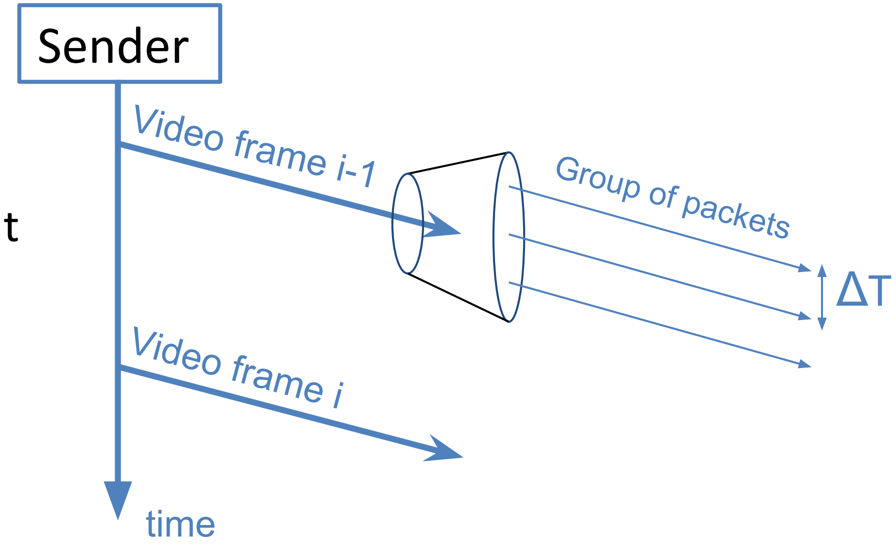
- The **Loss-Based** controller computes the rate A_s according to the fraction loss (**fl**) reported in RTCP:
  - **fl** > 0.1, A_s is decreased;
  - **fl** < 0.02, A_s is increased;
  - 0.02 ≤ **fl** ≤ 0.1, A_s is kept constant.

- The target bitrate ***A*** is set equal to min(A_s, A_r)

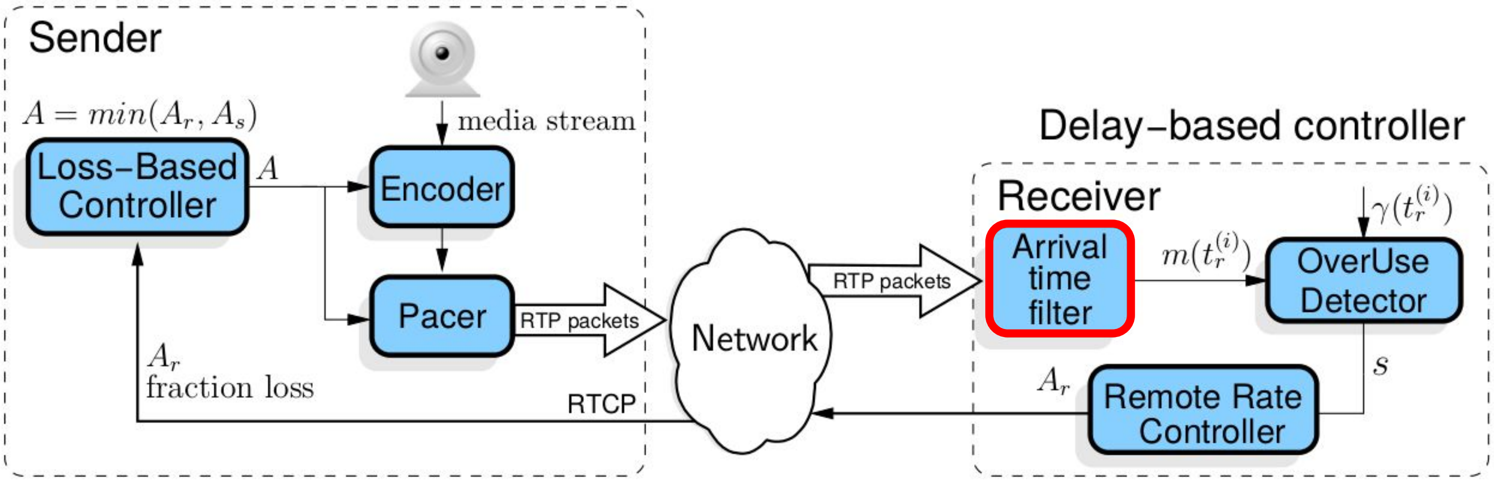
Sending Engine

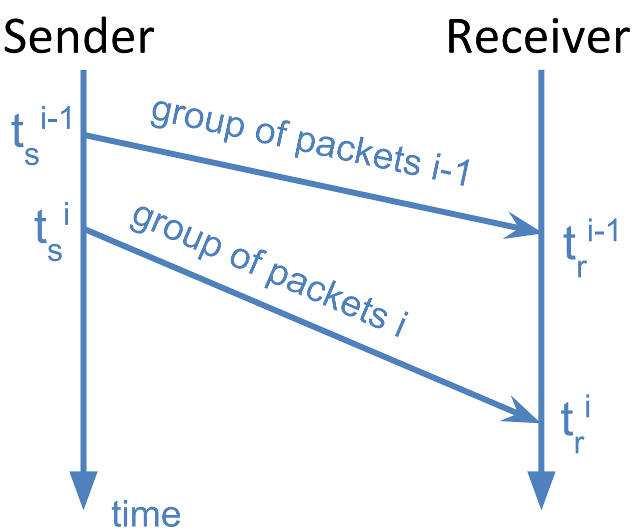


- Encoded media is fed into a Pacer queue.
- Pacer divides media into groups of packets that are sent to the network every ΔT=5ms.
- The size of a group of packets is equal to *A*·ΔT



Arrival Time Filter





ATF measures the one-way delay variation $d_m(t_r^{(i)})$:

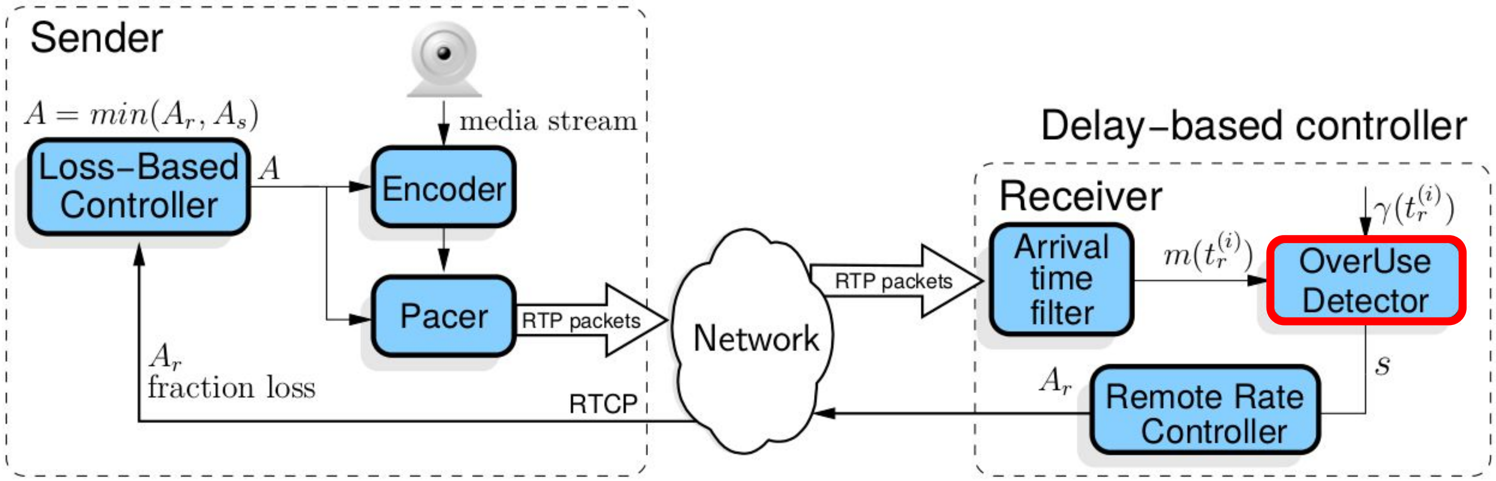
$$d_m(t_r^{(i)}) = (t_r^{(i)} - t_r^{(i-1)}) - (t_s^{(i)} - t_s^{(i-1)})$$

Estimates the queuing delay variation $m(t_r^{(i)})$ by means of a Kalman filter:

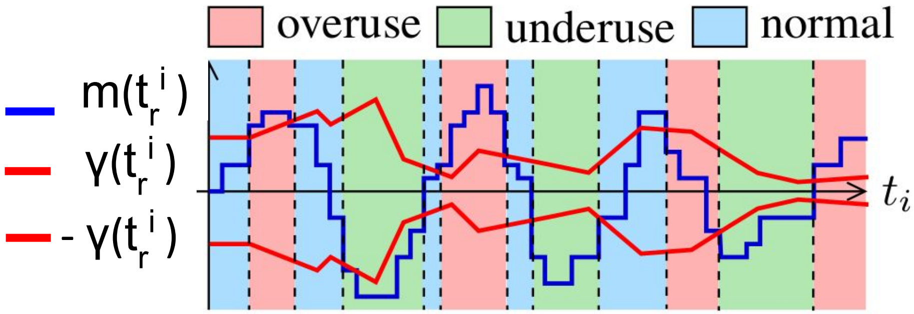
$$m(t_r^{(i+1)}) = (1-K(t_r^{(i)})) \cdot m(t_r^{(i)}) + K(t_r^{(i)}) \cdot d_m(t_r^{(i)})$$

$K(t_r^{(i)})$ is the Kalman Gain

OverUse Detector



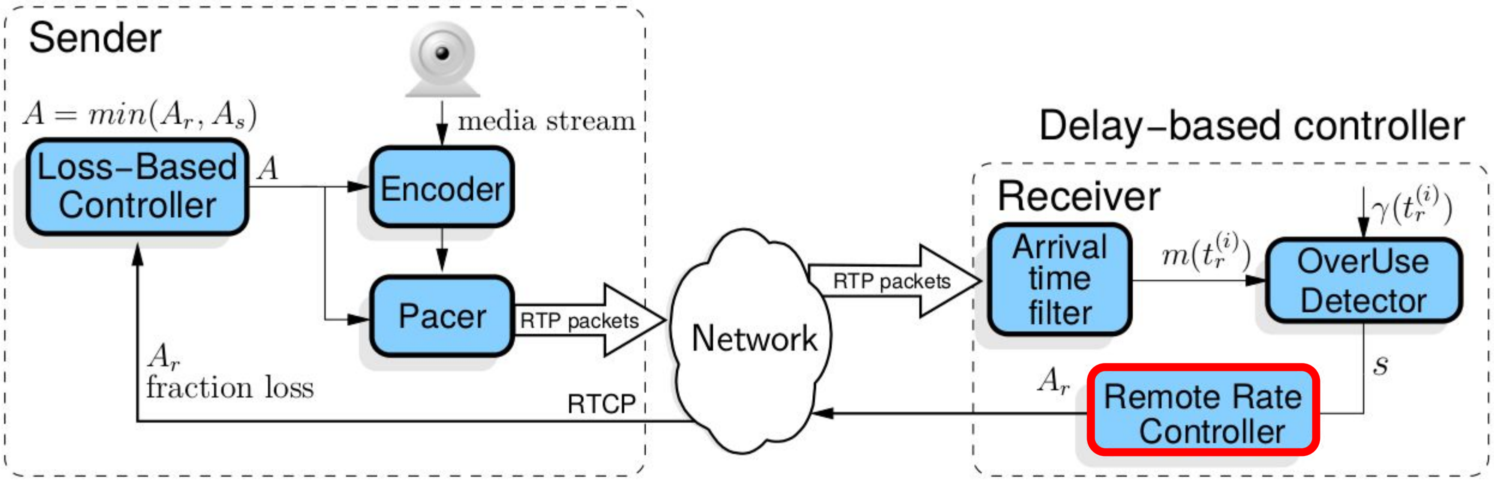
- Compares the estimated queuing delay variation $m(t_r^{(i)})$ with an adaptive threshold $\gamma(t_r^{(i)})$
- Based on the comparison, a signal s (overuse, underuse, normal) is generated



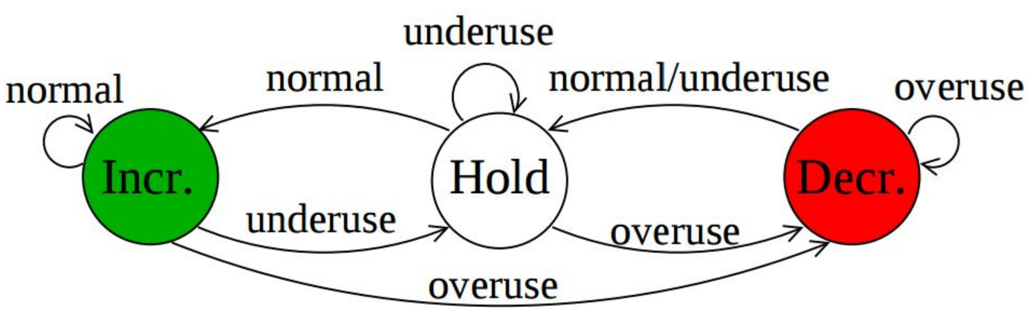
- The threshold γ is increased when m is outside the range $[-\gamma,\gamma]$ otherwise is decreased:

$$\gamma(t_r^{(i+1)}) = \gamma(t_r^{(i)}) + k_\gamma \cdot (|m(t_r^{(i)})| - \gamma(t_r^{(i)}))$$

Rate Controller



- The signal s is used to drive a FSM
- The FSM is used to compute the rate A_r based on the FSM state



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

- The goal of the FSM is to keep the bottleneck queue empty.

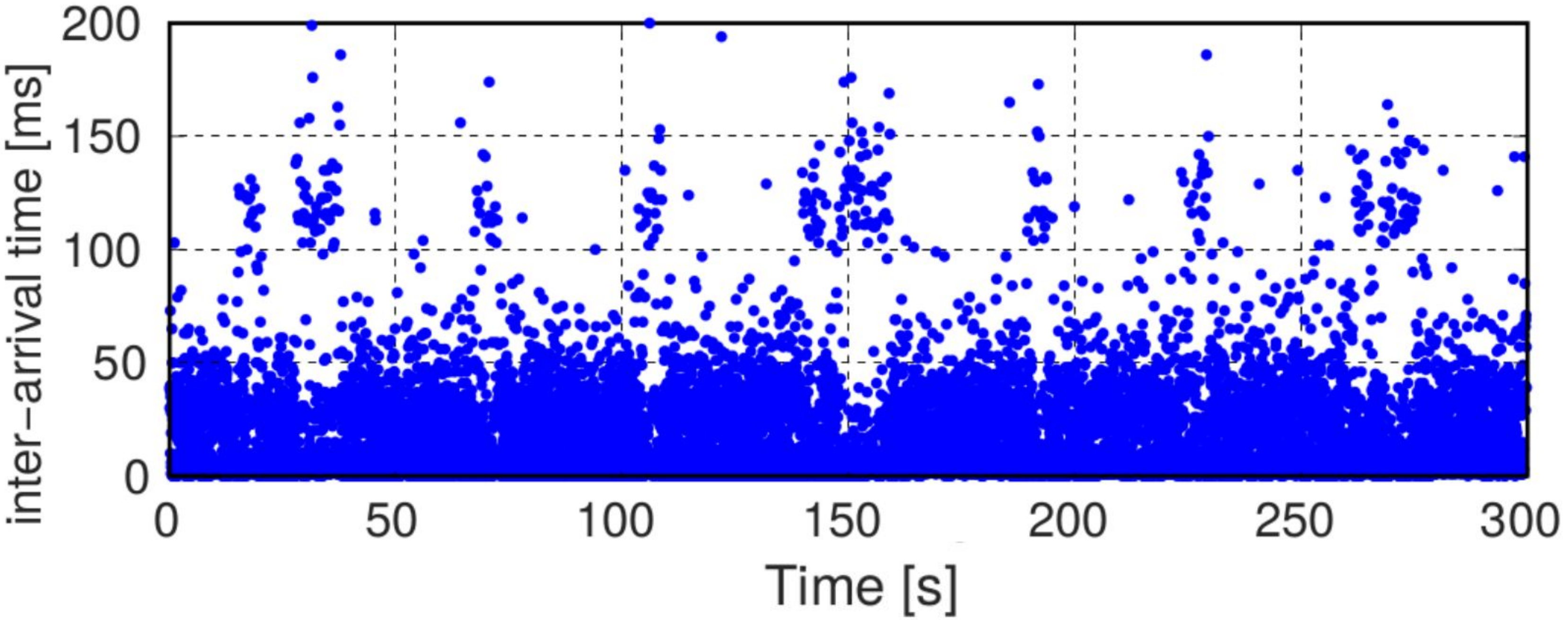
Focus of the Talk

Investigating the effect of wireless channel outages on GCC

Measurements over Wi-Fi network

Inter-arrival time between groups of packets over loaded 802.11n Wi-Fi network:

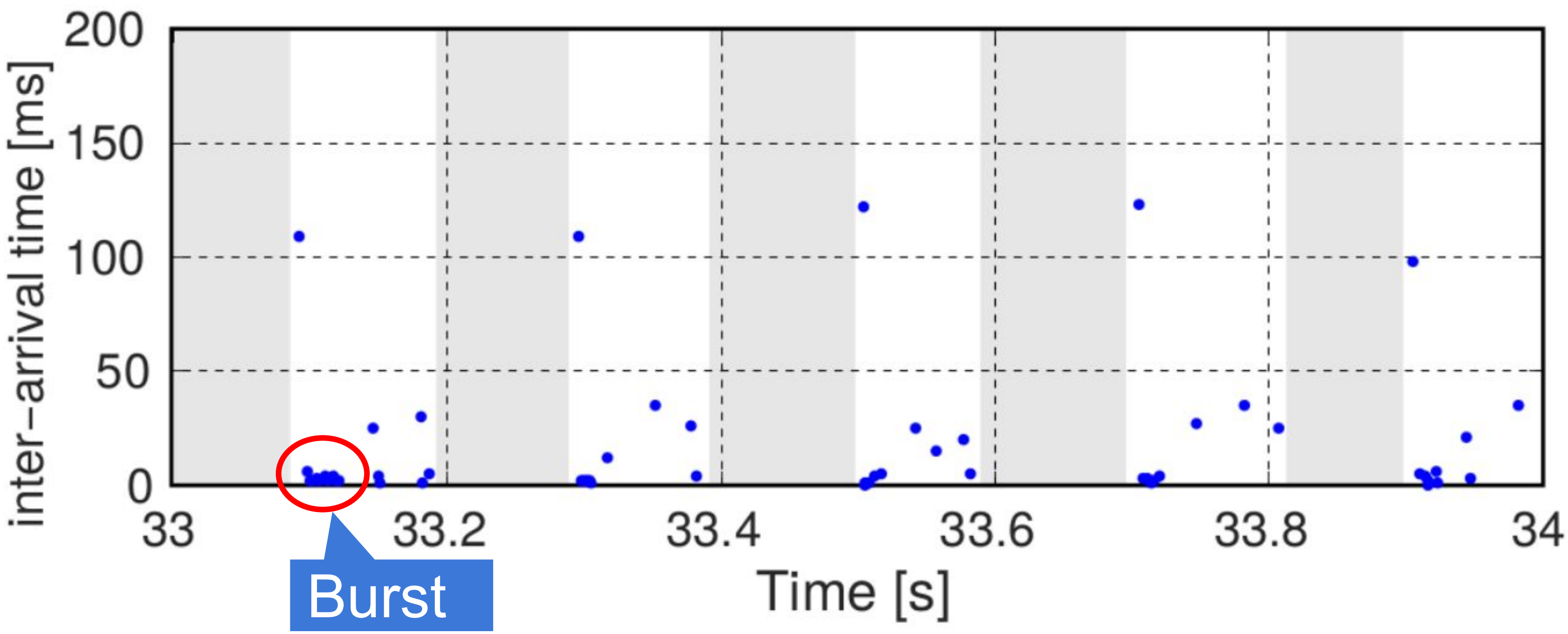
$$\text{inter-arrival time} = t_r^{(i)} - t_r^{(i-1)}$$



The inter-arrival time experiences a high variance.

Measurements over Wi-Fi network

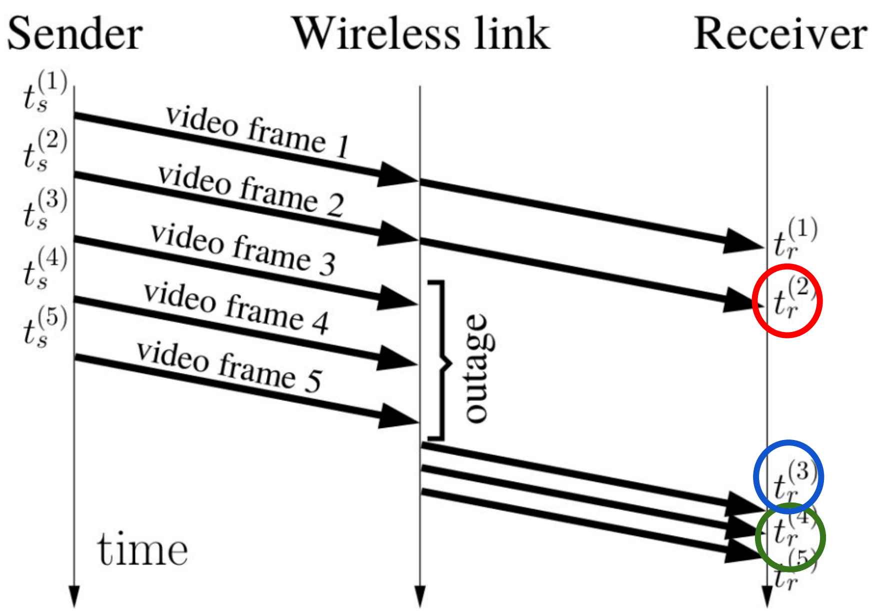
Inter-arrival time temporal zoom [33,34]:



- High variance is due to the effect of outages (grey)

- Channel outages are time-varying: packets being queued in network buffers, for reasons unrelated to congestion, are delivered in a burst when the outage ends.

Effects of channel outages on the one-way delay variation



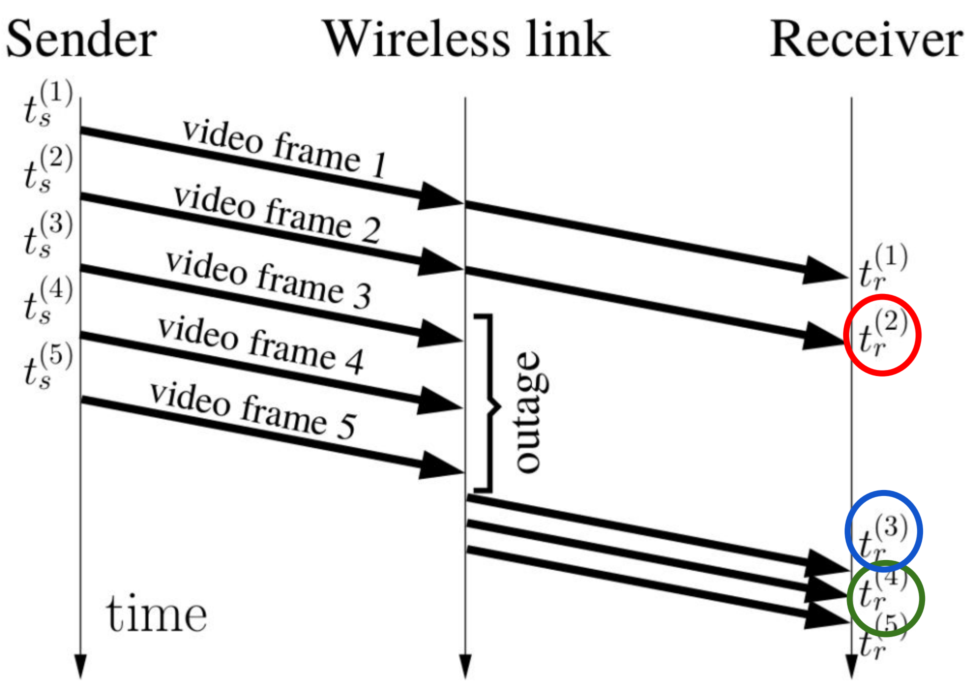
Three patterns of the one way delay variation $d_m(t_r^{(i)})$ can be distinguished:

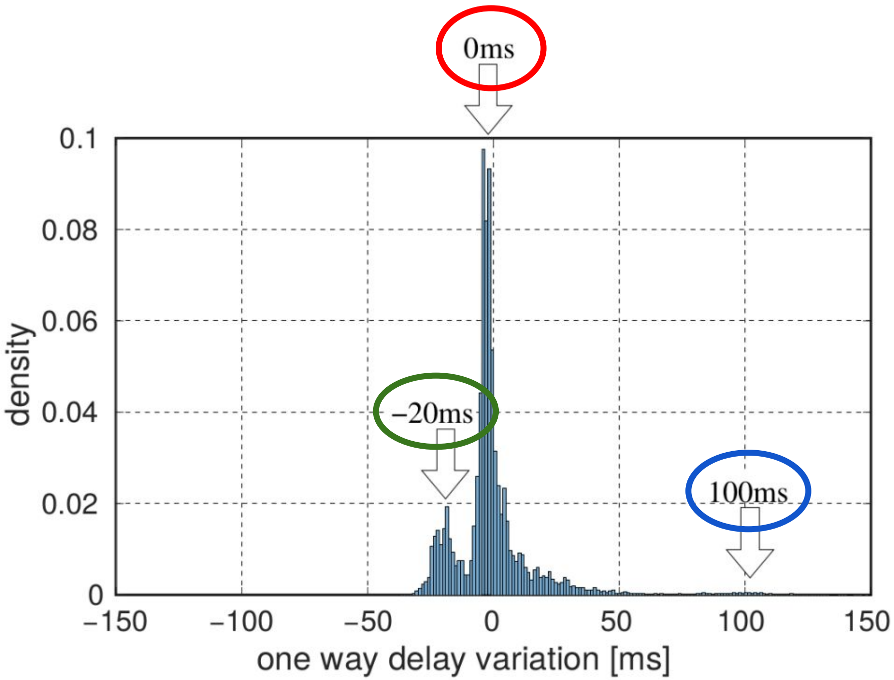
1) $d_m(t_r^{(2)}) = (t_r^{(2)} - t_r^{(1)}) - (t_s^{(2)} - t_s^{(1)}) \simeq 0ms$

2) $d_m(t_r^{(3)}) = (t_r^{(3)} - t_r^{(2)}) - (t_s^{(3)} - t_s^{(2)})$ → large and positive

3) $d_m(t_r^{(4)}) = (t_r^{(4)} - t_r^{(3)}) - (t_s^{(4)} - t_s^{(3)})$ → small and negative

Effects of channel outages on the one-way delay variation

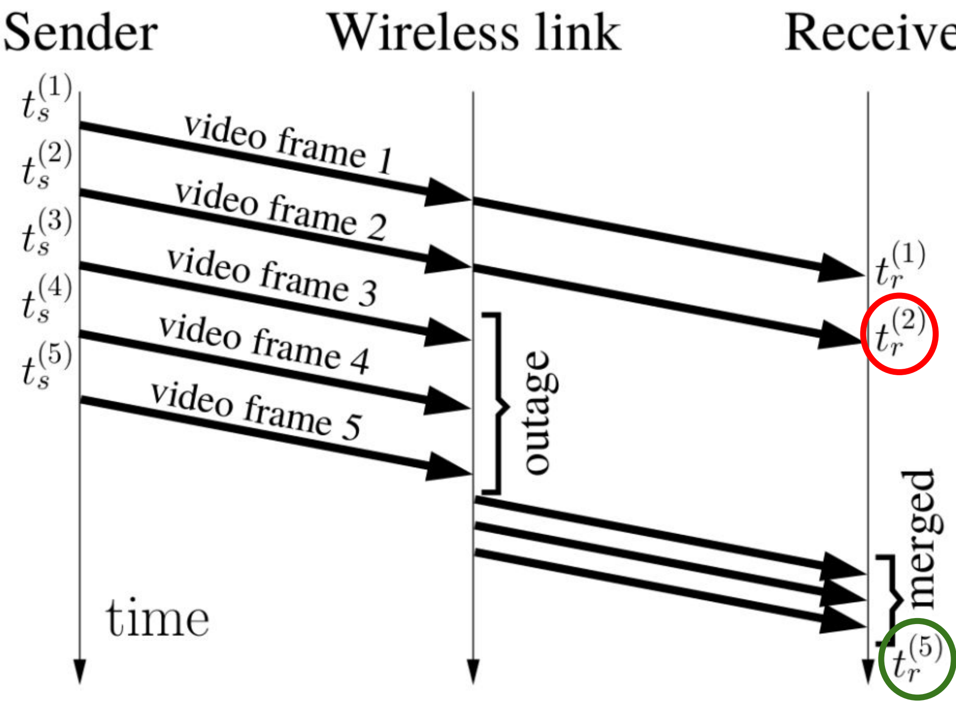




The probability density function is the superposition of three Gaussian-like distributions.

Solution

We add a **pre-filtering block** before the Arrival Time Filter (ATF) which merges packets that arrive in a burst in one group



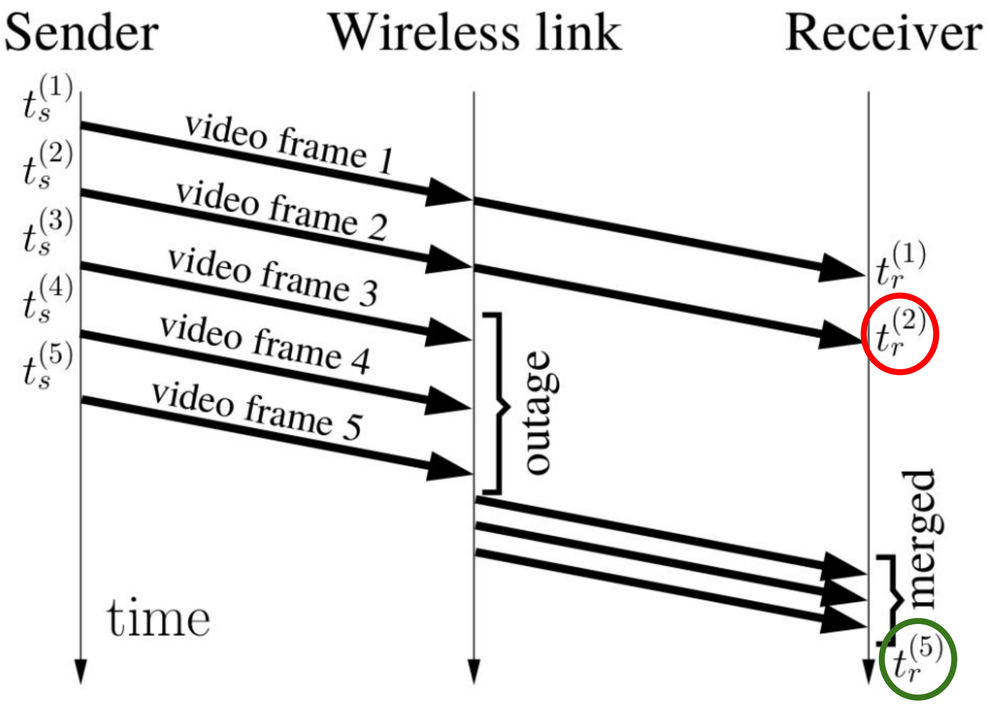
1) $d_m(t_r^{(2)}) = (t_r^{(2)} - t_r^{(1)}) - (t_s^{(2)} - t_s^{(1)}) \simeq 0\text{ms}$

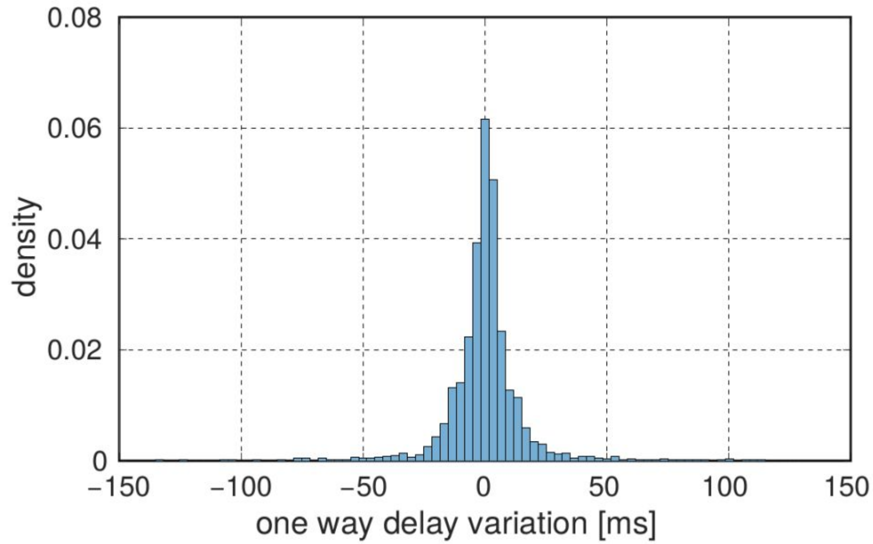
2) $d_m(t_r^{(5)}) = (t_r^{(5)} - t_r^{(2)}) - (t_s^{(5)} - t_s^{(2)}) \simeq 0\text{ms}$

By merging burst arrivals fewer samples will be fed to the ATF

Solution

We add a **pre-filtering block** before the Arrival Time Filter (ATF) which merges packets that arrive in a burst in one group

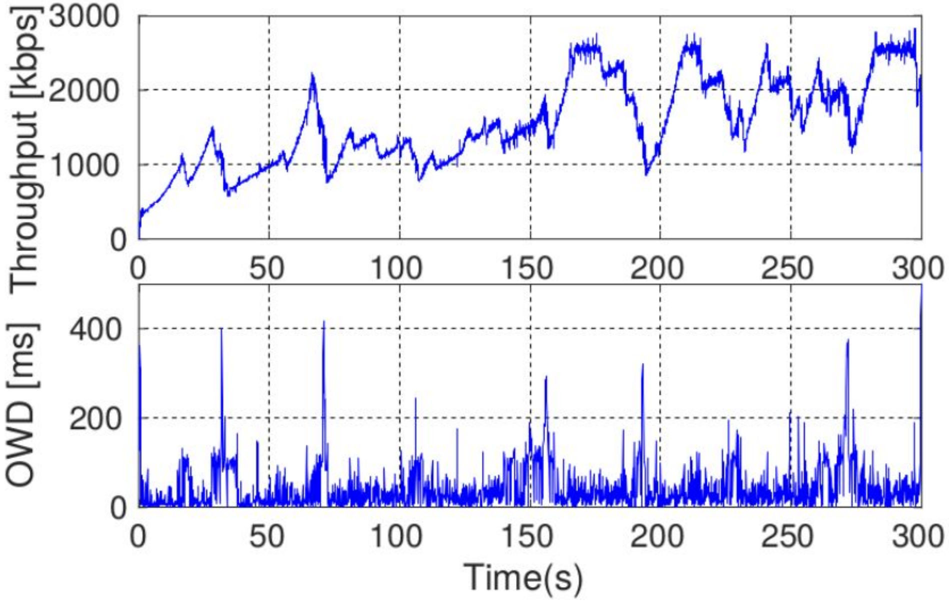




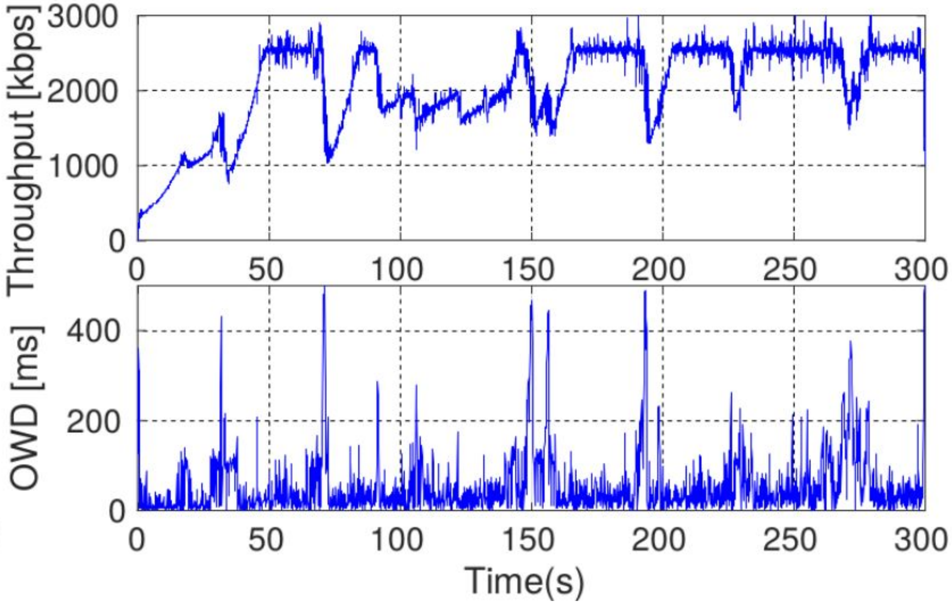
Pre-filtering makes the distribution of one way delay mono-modal

Experimental comparison

w/o pre-filtering

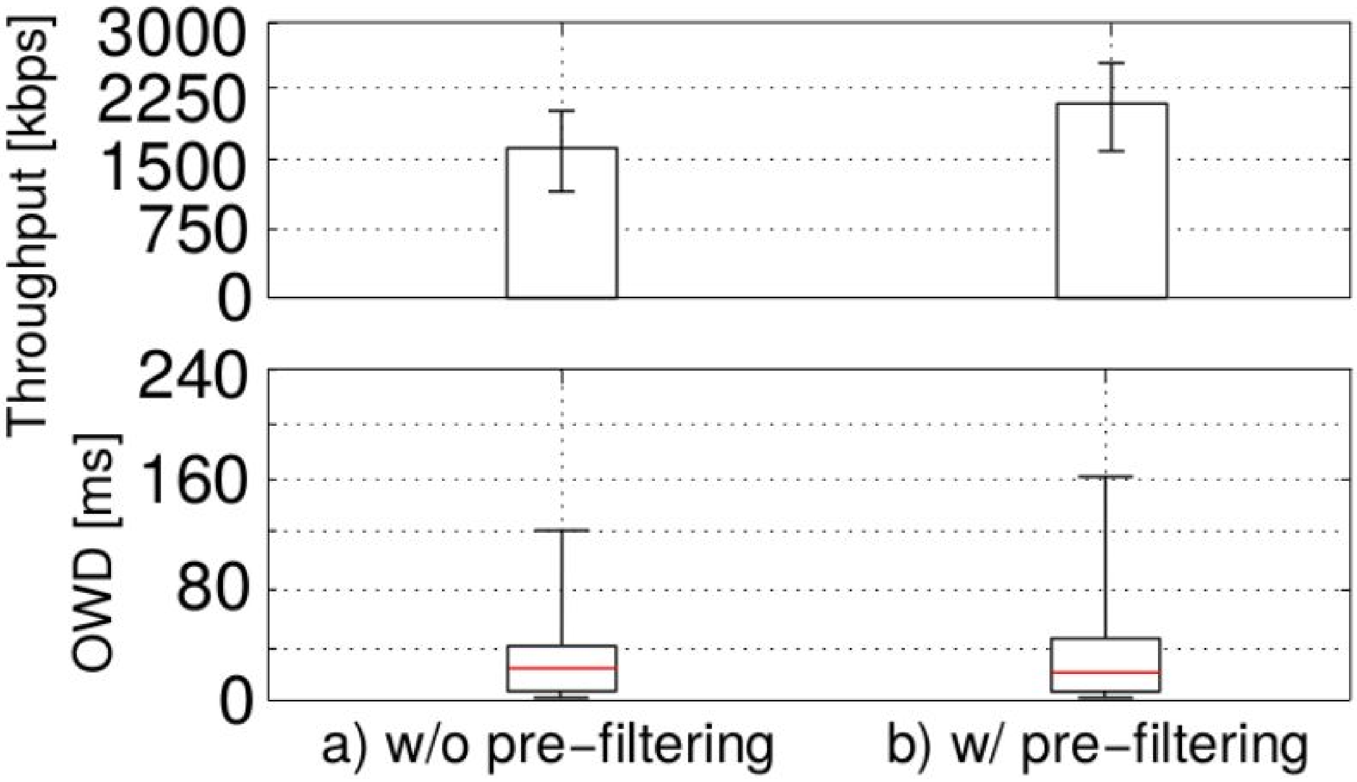


w/ pre-filtering



Trace-based evaluations shows that average throughput is higher without worsening the one way delay (GCC video encoder does not send more than 2500 kbps)

Experimental comparison



- Average thoughput is improved by ~20%
- Median one way delay is not affected
- 95th percentile of the one way delay is slightly higher with pre-filtering

CONCLUSION

- GCC is being used in Google Hangout and in the WebRTC implementation of Google Chrome since more than 3 years
- GCC shown a pretty stable behaviour over wired networks
- In this work we have made GCC more robust over Wi-Fi networks
- GCC is implemented in the webrtc.org repository and results can be easily reproduced

CONCLUSION

QUESTIONS?

THANK YOU

BACK UP

BACK UP

Pre-Filtering Algorithm

$(t_r, t_s) \leftarrow$ on RTP packet arrival;

if $((t_r - t_{r\_old}) < \Delta T)$ *and* $(d_m(t_r) < 0)$ **then**

  Merge packet in the current Group;

else

  if $(t_s - t_{s\_old}) \geq \Delta T$ **then**

    New Group is arriving;

    Update ATF with (t_{r_old}, t_{s_old});

  else

    Merge packet in the current Group;

$t_{r\_old} \leftarrow t_r$;

$t_{s\_old} \leftarrow t_s$;