rLEDBAT

Marcelo Bagnulo & Alberto Garcia Martinez
U. Carlos III de Madrid

Anna Maria Mandalari
Imperial College London

Praveen Balasubramanian, Daniel Havey, Gabriel Montenegro
Microsoft

About LEDBAT

- Ledbat (RFC6817) is a congestion controller that provides less than best effort traffic
- LEDBAT algorithm. Defines a queuing delay target T
  - Increase/decrease CWND based on the queuing delay
  - Queuing delay calculated as the difference between the OW Base Delay and the OW Current Delay

LEDBAT shortcomings

- Poor interledbat fairness
  - Late comer advantage
- Difficulties measuring the OWD in TCP, even using TCP TS.

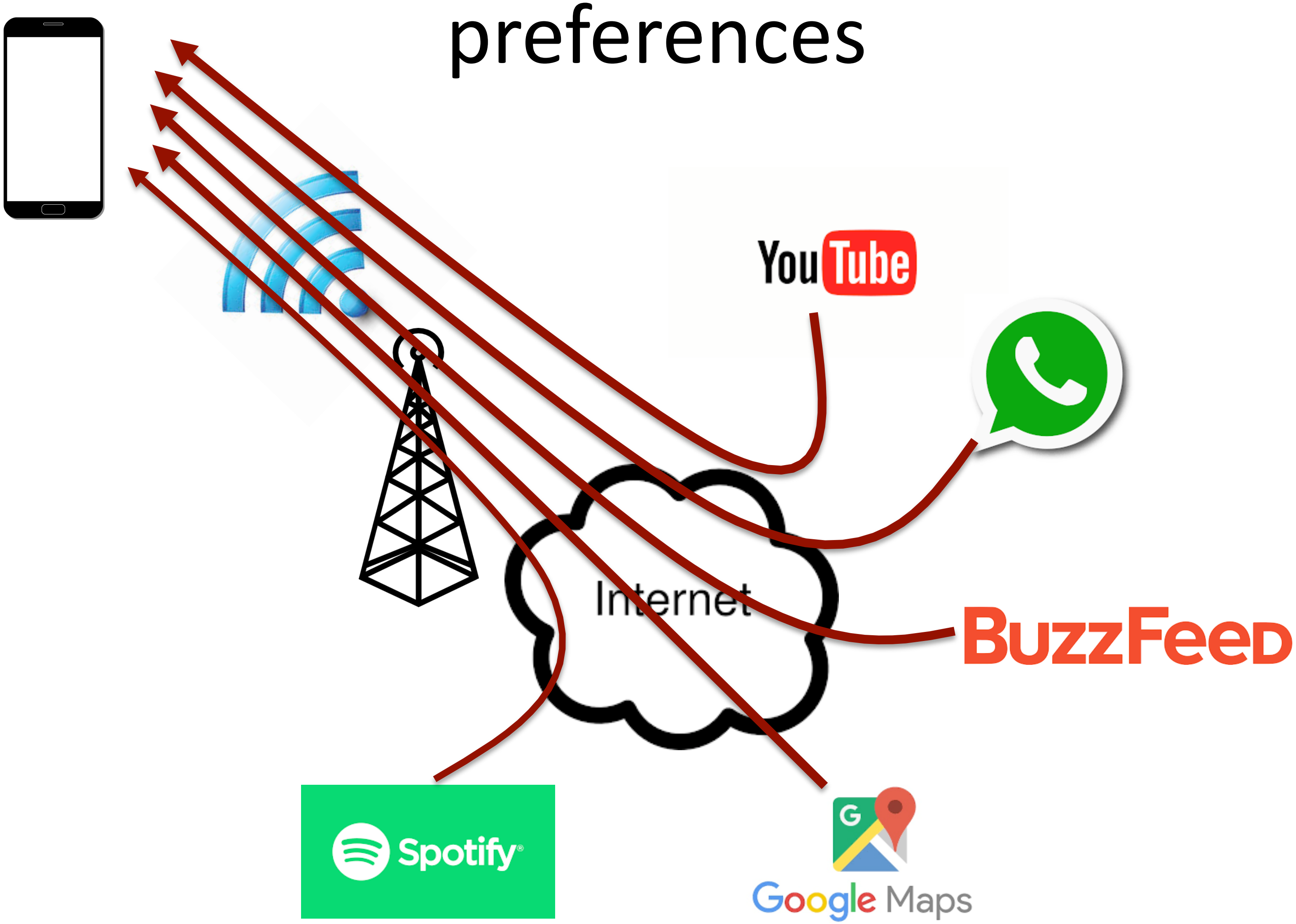
rLEDBAT

- Receiver based less than best effort congestion controller for TCP
  - Based on LEDBAT (LEDBAT++ actually)
- Congestion control algorithm runs in the receiver
- The receiver controls de sender's rate through the RCVWND
- Measure RTT and reacts accordingly

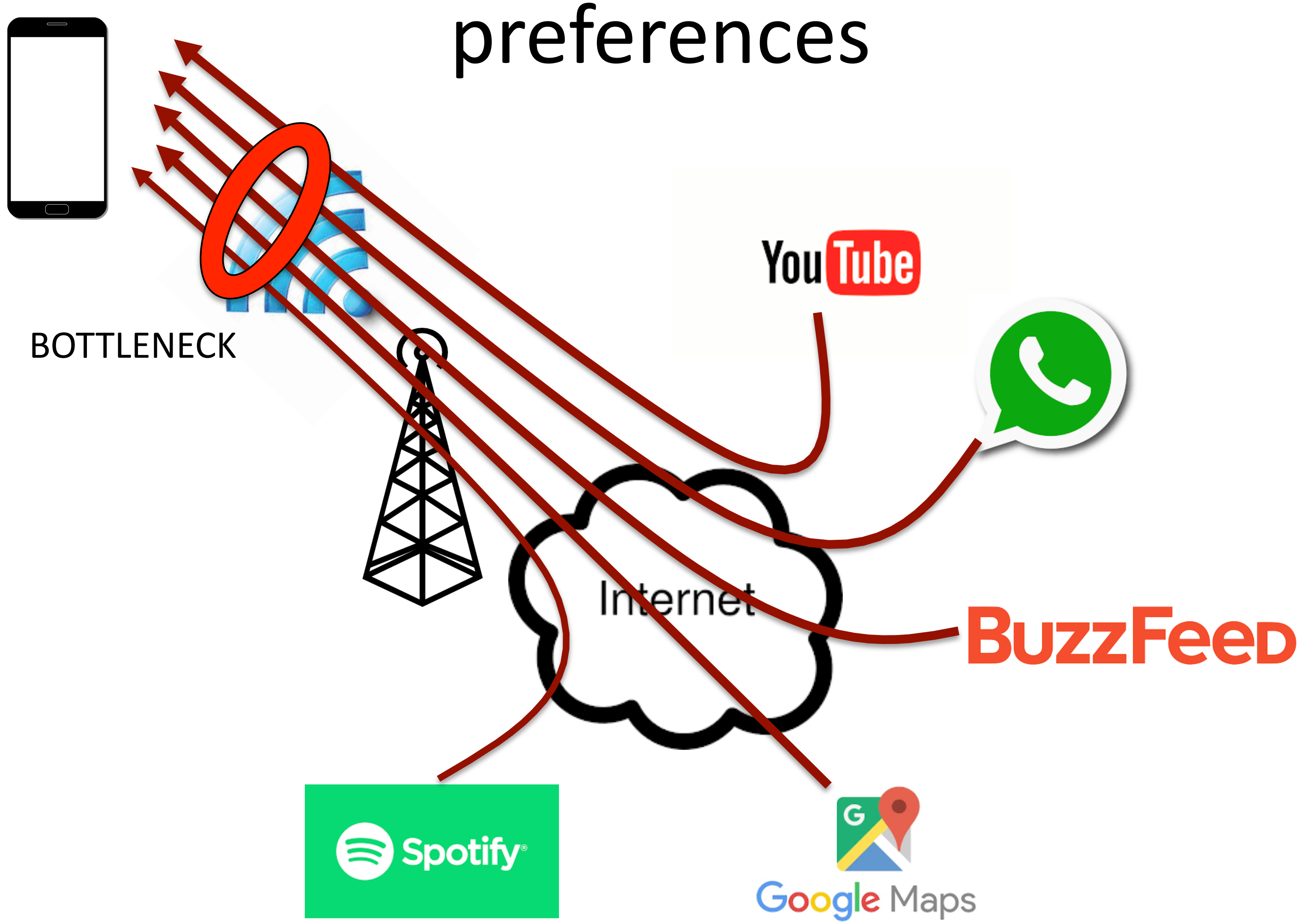
rLEDBAT motivation (I)

- New deployment models
  - File distribution through CDNs (e.g. software updates) do not benefit from LEDBAT
    - CDN surrogates rarely implement LEDBAT
    - No signalling available to convey which content is ledbat
  - Transport layer proxies are a commonplace.
    - Security, performance, etc
    - Proxies do not implement and do not know when to use LEDBAT
    - The segment between the proxy and the receiver does not benefits from LEDBAT

rLEDBAT motivations (II): User defined preferences



rLEDBAT motivations (II): User defined preferences



rLEDBAT motivations (II): User defined preferences

- The user is aware of her own preferences which should rule the bottleneck resource allocation
  - The servers are not aware of them
  - The base station is not aware neither
- The sender cannot properly select which traffic is LEDBAT is which one is best effort.

rLEDBAT algorithm goals

- When rLEDBAT is sharing a bottleneck with latency sensitive traffic (e.g., VOIP traffic), the queueing delay introduced by rLEDBAT should be acceptable for the other traffic in the link.
- When there are several rLEDBAT flows, the available capacity should be equally split.
- When there is only rLEDBAT traffic in a bottleneck link, it should be able to seize all available capacity in steady state.
- When rLEDBAT is competing with best effort flows (in particular with standard-TCP), rLEDBAT should minimize its interference by giving available capacity for best effort flows.

rLEDBAT algorithm

- Measuring the RTT
- qd is equal to the current RTT minus the base RTT
- Define a target RTT T
- if qd < T , then rlWND[t1] = rlWND[t0] + α*MSS/rlWND[t0]
- if qd > T , then rlWND[t1] = rlWND[t0] * βd (once per RTT)

Controlling the RCVWND

- Compatibility with flow control
  - RWND carries the min of the flow ctrl and rLEDBAT
- Interaction with sender's congestion control
  - CWND is the min of sender's cong ctrl and RWND
  - rLEDBAT is less aggressive than TCP cong ctrl, so likely RWND< cong ctrl wnd
- Avoiding shrinking the window
  - rLEDBAT multiplicative decrease may result in reductions larger than the received bytes
  - rLEDBAT multiplicative decreases at most once per RTT
  - rLEDBAT drains packets in flight until reaches the desired RWND without shrinking it
- WS option
  - WS values between 0 and 11 result in units of less than 1 MSS
  - WS values larger than 12 result in more coarse control, more experiments are needed buit values are rarely used and WS is set by the client.

Using the RTT

- Includes the queuing delay in the reverse path
- Pure receivers: use the TS to match packets
- Increased RTT due to inability to send packets in the sender
  - No data, rare
  - No RCVWND, avoid measuring if the RCVWND is being reduced
- Granularity of TS values may result in multiple pkts carrying the same TS
  - Only use the first pkt with a TS value

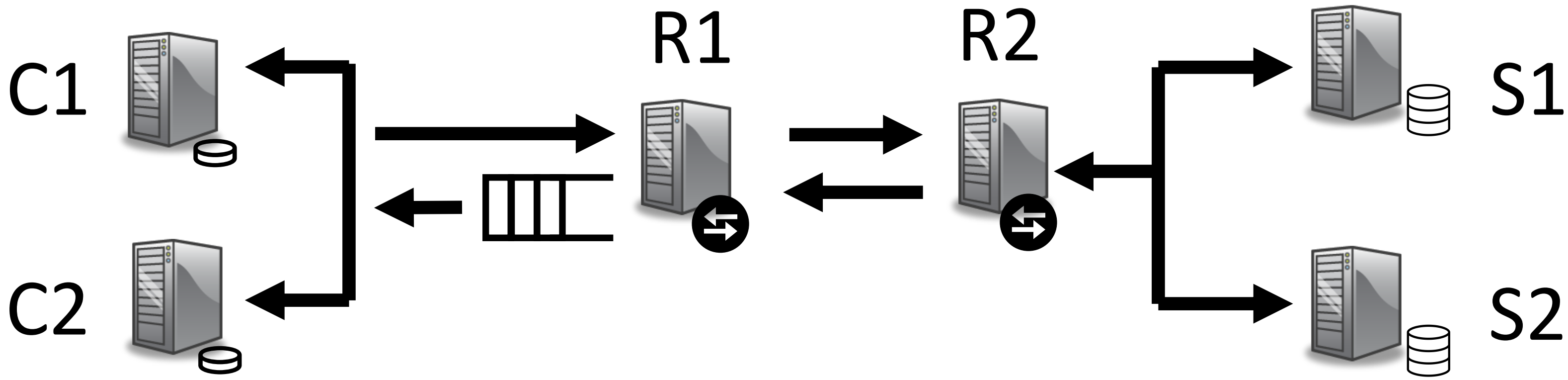
Other design choices

- Interledbat fairness: AIMD
- Reacting to packet loss: MD
- Bootstrapping: let flow control to take over
- Path changes: similar to LEDBAT

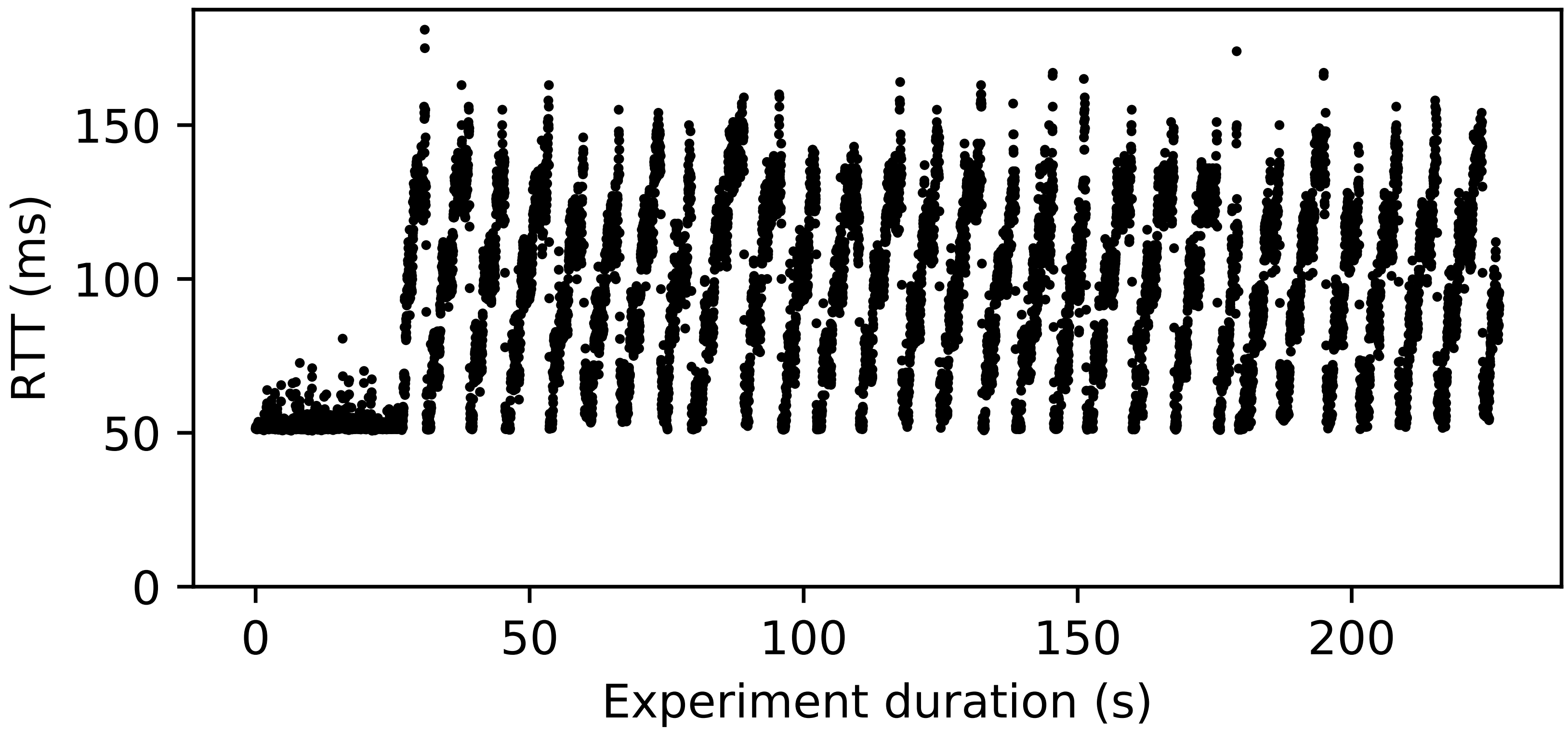
Experimental setup

rLEDBAT implementation available

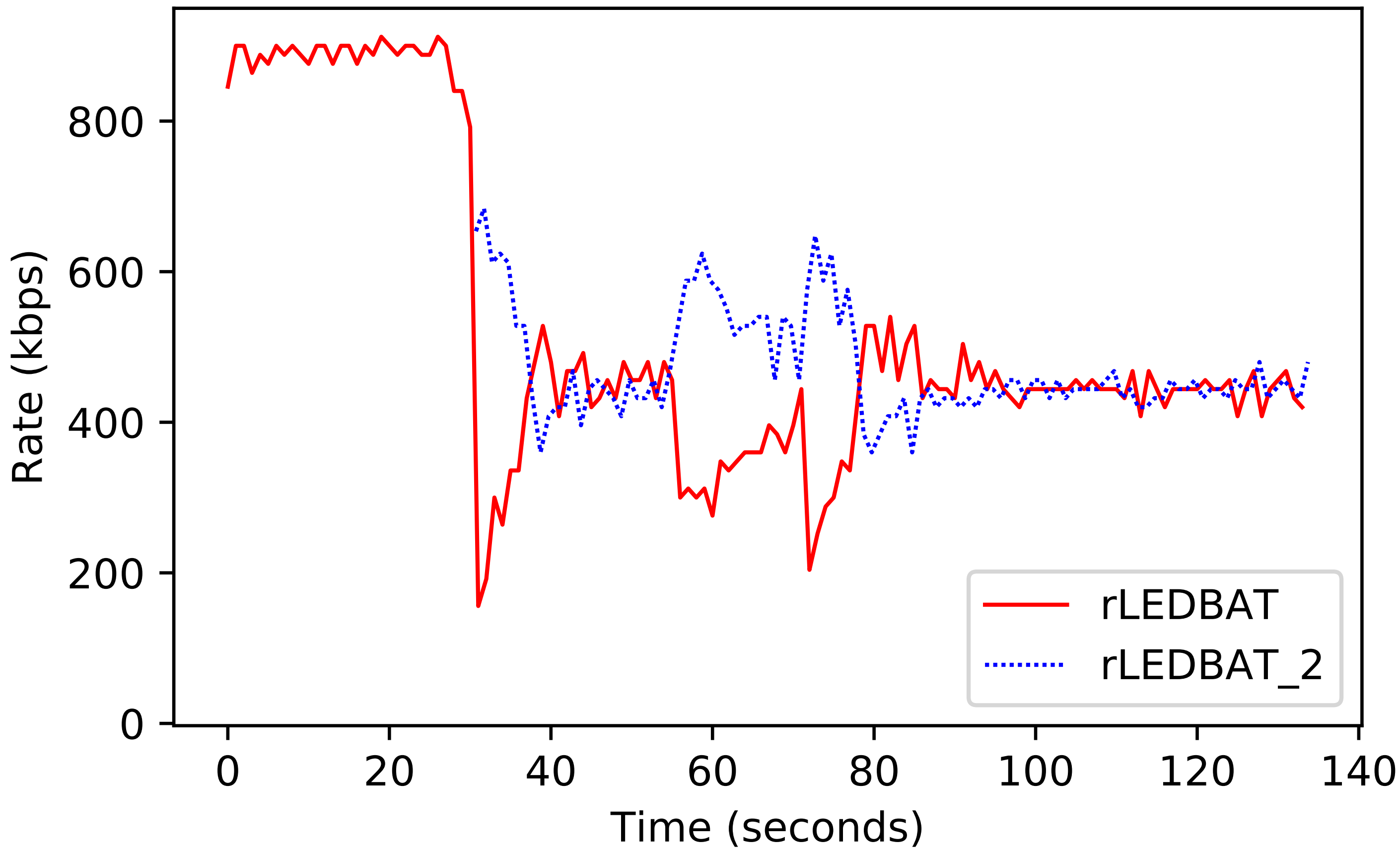
rledbat.netcom.it.uc3m.es



rLEDBAT and VOIP



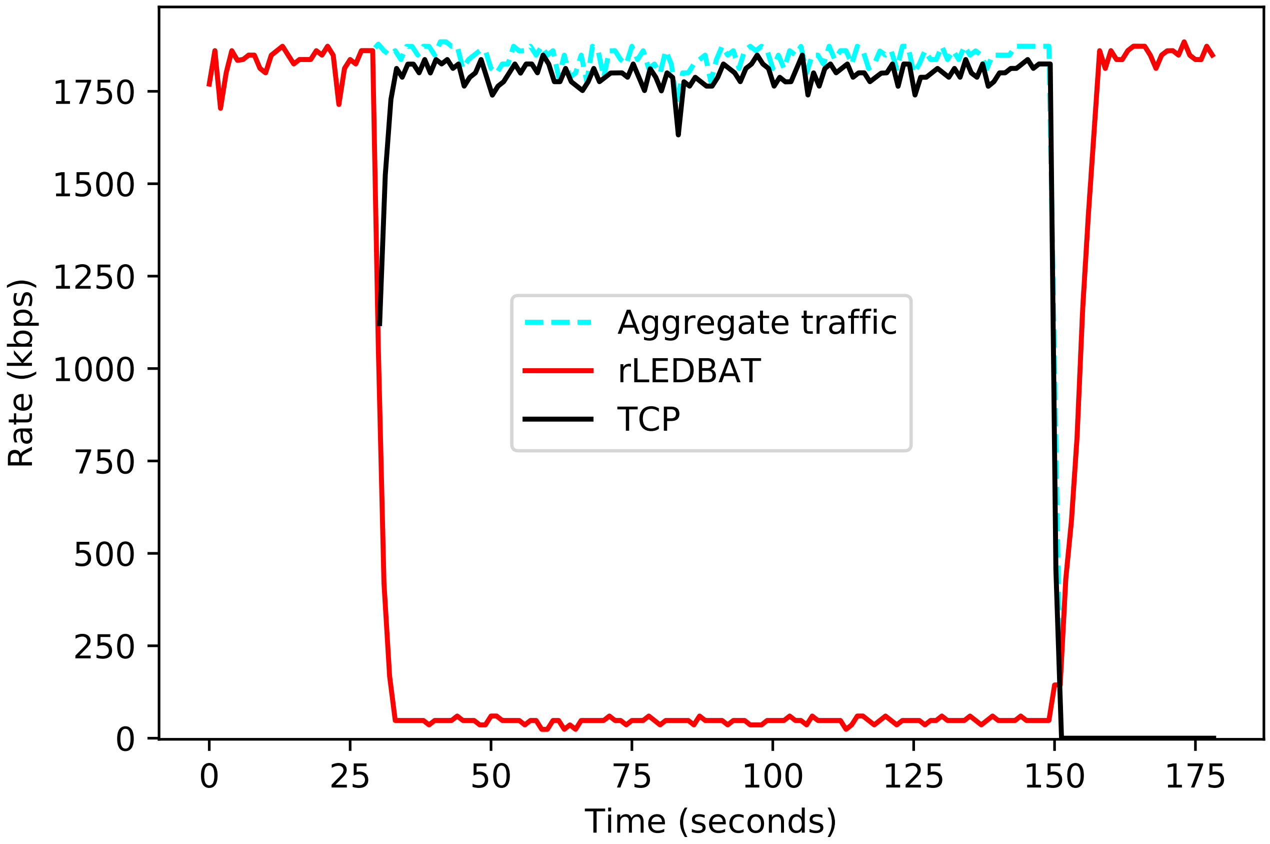
Inter-rLEDBAT fairness



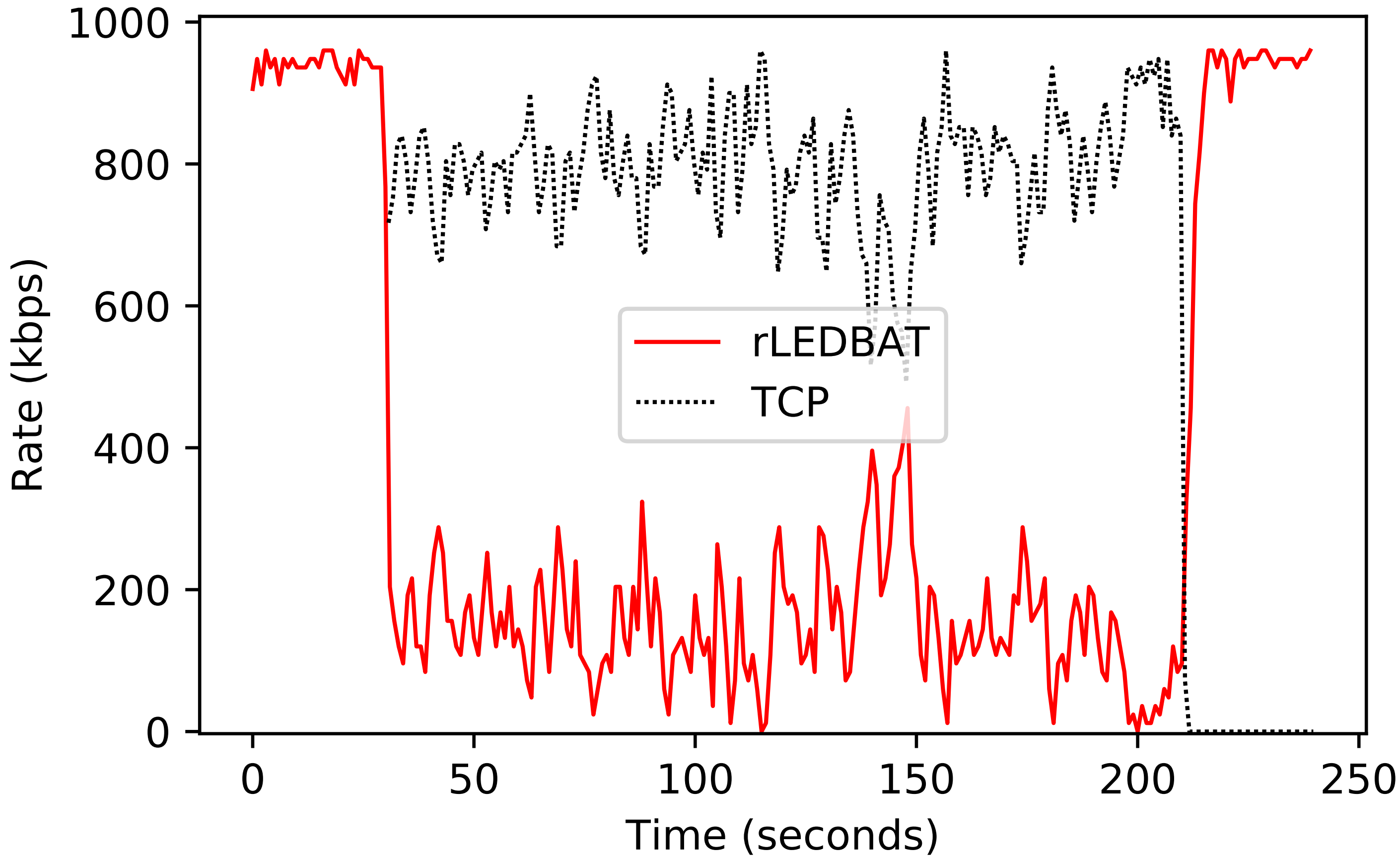
rLEDBAT solo performance

RTT	Capacity used	Theory Approx.
74 ms	944 kbps (99%)	100 %
124ms	922 kbps 96.6%)	99 %
174 ms	886 kbps (92.8%)	97 %
224 ms	857 kbps (89.8%)	95 %
274 ms	838 kbps (87%)	93 %
324 ms	819 kbps (85%)	91 %
374 ms	815 kbps (85%)	90%
424 ms	809 kbps (84%)	88%
474 ms	788 kbps (82%)	87%

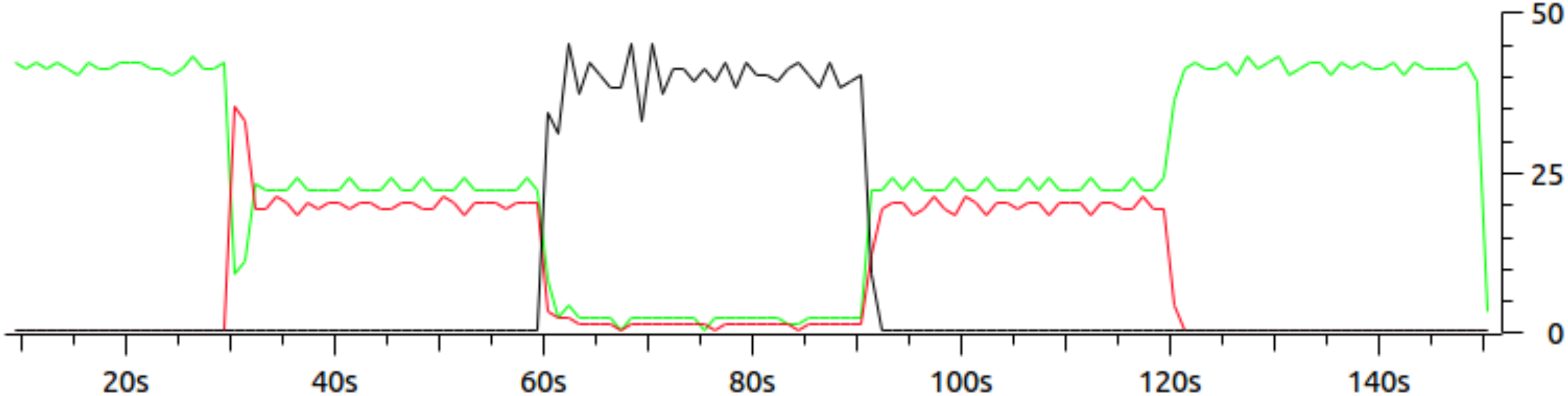
Interference with TCP – delay based



Interference with TCP – loss based



Interaction with LEDBAT++



Wrap up

- Is ICCRG interested in working on this?