One control to rule them all

Michael Welzl
How we use the Internet today: 3 stories

1. I clean our flat while listening to Spotify
   – in parallel, downloading files via my own
   – Suddenly I begin to think:
     “please, dear downloads, don’t make the music stop!”

2. I am in a hotel room, using Skype to see my daughter
   – Quality barely good enough; I avoid clicking on anything
   – Note: that’s different when I talk to my mother...

3. Downloads can have different priorities, too
   – When I download two files, I try to guess whether the downloads slow each other down
So you care more about “performance”? 

- What is it to you?
How to fix this

• The problem can be solved with a single Congestion Control instance (as in RFC3124)
  – But solving it in general is hard – RFC3124 leaves some key issues unresolved + benefits weren’t shown
    • shared bottleneck or not?
    • overally less aggressive CC – bad e.g. for short flows?
      ... all at the cost of a complex implementation!

• But we could do this right for rtcweb
  – Common bottleneck is assumed (all-over-one-5-tuple)
  – long connections are somewhat likely
Lots of benefits

• **Really** able to control fairness
  – outcome is result of a sender-side scheduler, not of “fighting it out” at the bottleneck

• Less queuing delay: only one flow

• Better performance for short or application-limited flows
  – skip slow start; again less queuing delay from slow start overshoot

• Less feedback needed
  – avoid that e.g. data channel feedback (SCTP SACK chunks) is ignored by RTP’s CC.