UiO : Faculty of Mathematics and Natural Sciences
University of Oslo

ifi

# Considerations for Controlling TCP's Fairness on End Hosts

*Michael Welzl*

*Capacity Sharing Workshop*
*Stuttgart*
*13. 10. 2011*

# The problem

# TCP's fairness

- ...has been criticized a lot.
- Hi Bob! ☺
- Many good reasons
  - e.g., depending on RTT = technical artifact
- Here: a very pragmatic, practical view of the problem, and what to do about it

# How we use the Internet today: 2 stories

1. I clean our flat while listening to Spotify via my wife's laptop
   - in parallel, downloading files via my own
   - Suddenly I begin to think:
     "please, dear downloads, don't make the music stop!"

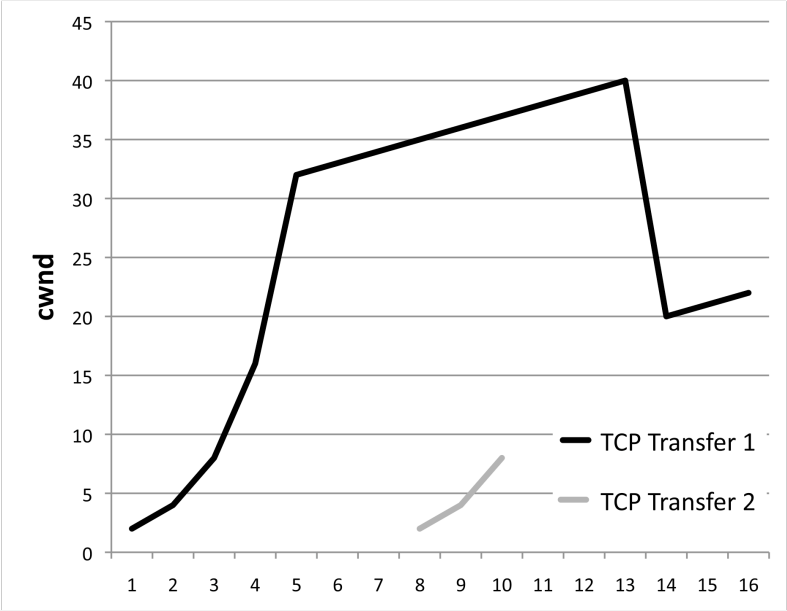
2. I am in a hotel room, using Skype with video to see my daughter
   - Quality barely good enough
   - I avoid clicking on anything
   - Note: that's different when I talk to my mother...

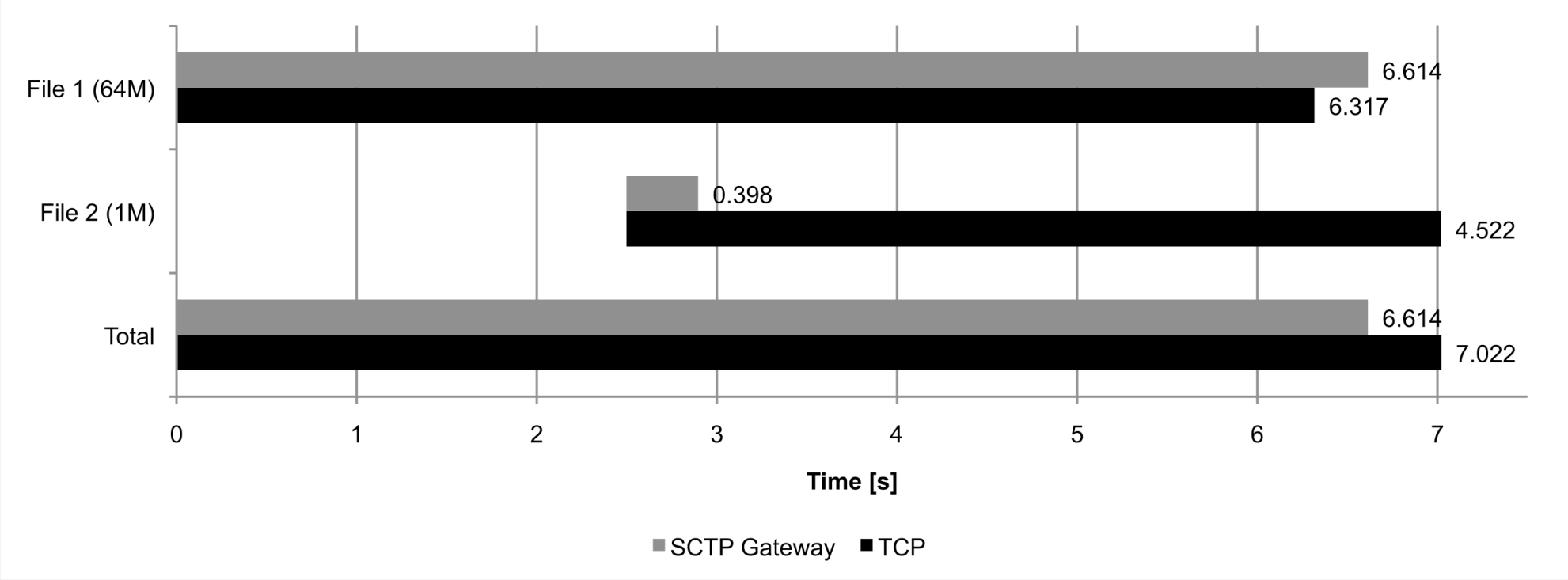
# A major problem

- We may have become used to this, but that doesn't mean it's good?!
  - Would like to specify: do not interrupt Spotify / Skype (or know: do downloads disturb Spotify / Skype or not?)
- These were just two examples
  - Downloads can also have different priorities
  - When I download two files, I try to guess whether the downloads slow each other down

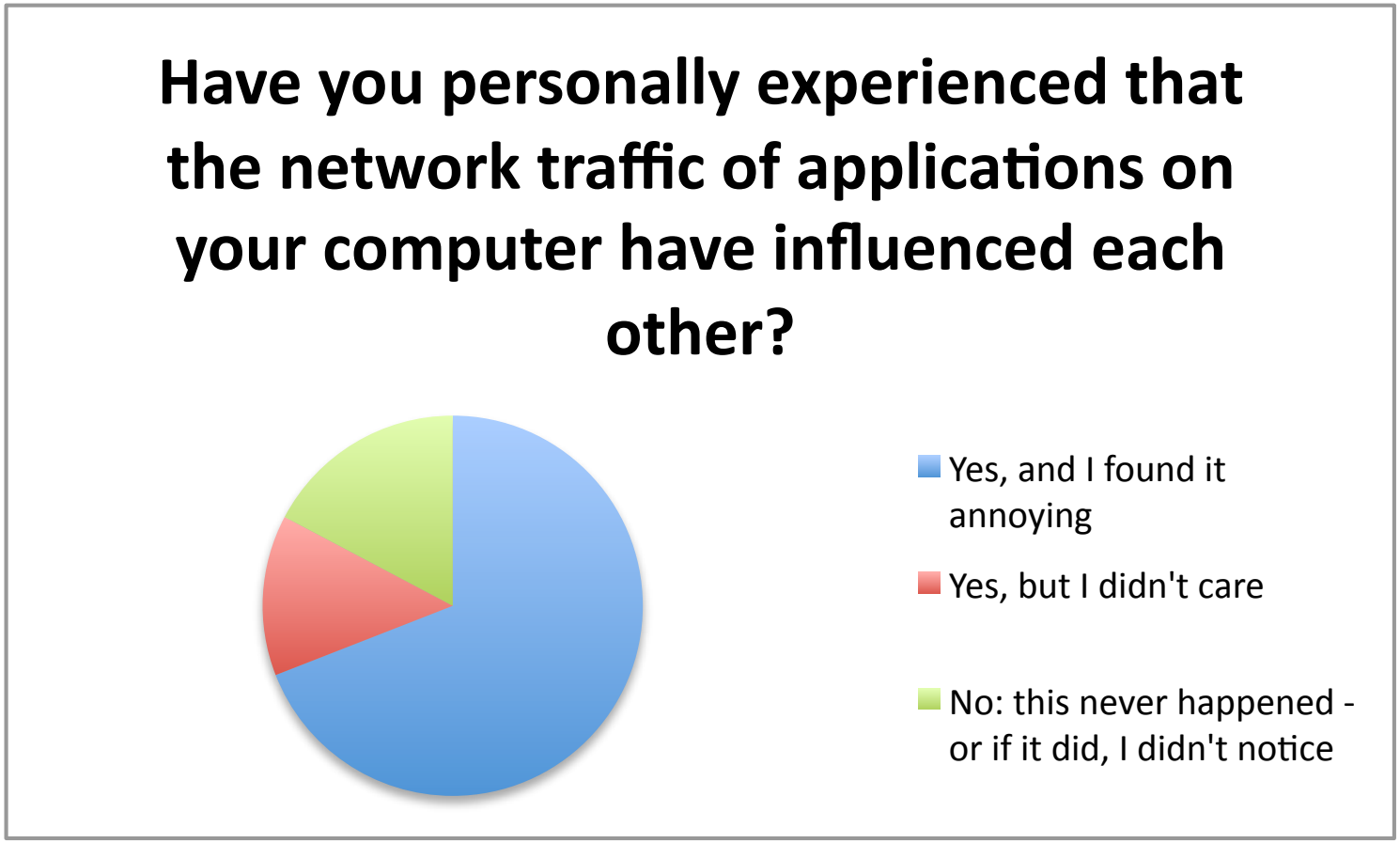
# So you care more about "performance"?

- What is it to you?

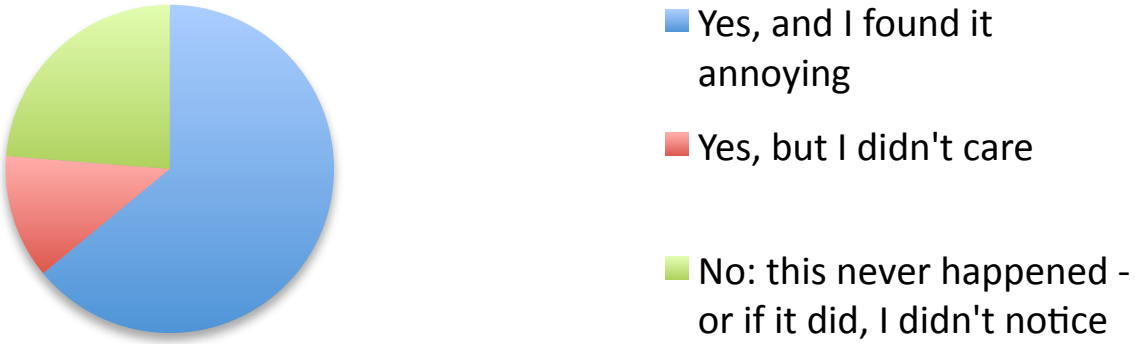


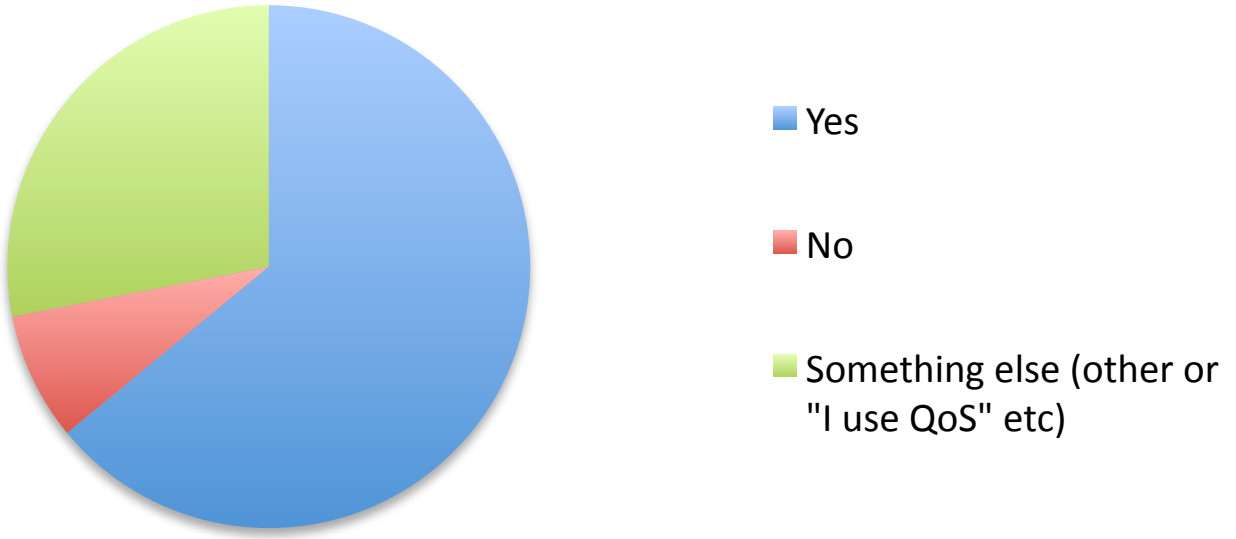
# Opinions:
# 139 of my work colleagues, students, and Facebook "friends"



## Have you personally experienced that the network traffic of applications in a small local network (e.g.: within your home) have influenced each other?



## If there was an easy-to-use tool that would let me prioritize how my applications access the network, I'd use it



# The solution

# NOT queue management!

(e.g. Linux gateway with tc or GUI tools like NetLimiter)

- Your access link may not be the bottleneck
  - Even if the access is likely, it can also be the other side (e.g. P2P, Skype, ..)
- We want TCP to maintain priorities at all times
- Two cases, both relevant to end users, and separate but interoperating solutions needed:
  1. Uploads
  2. Downloads

# Uploads

- Exact control over fairness between N flows across one bottleneck requires cwnd sharing
  - but need “aggression” of N to avoid being disadvantageous => a good MulTCP-like mechanism
  - We have PA-MulTCP, CP, MulTFRC, and some more
- Share cwnd if flows use different paths: very inappropriate behavior
  - Do this only when traversing the same bottleneck
  - Need shared bottleneck detection

# Downloads

- Need to control the sender
  - Need signaling extension to TCP
- Do this only for flows that share bottlenecks
  - Need shared bottleneck detection

# Conclusion: Ingredients of the fairness soup

- Shared bottleneck detection
  - for the user: know about mutual influence of transfers
  - for upload and download: control fairness only among flows that share a bottleneck
  - Solutions exist; have been critized for not being 100% reliable – not a problem for this application?!
- cwnd sharing
  - Solutions exist (CM, TCB interdependence (RFC 2140))
- Tunable-aggression-TCP
  - Solutions exist
- E2E-signaling of fairness requirements
  - Doesn't exist?!

(... and a GUI that shows transfers by application; existing tools can do that)

# Thank you!

# Questions?