A Survey of Lower-than-Best Effort Transport Protocols

draft-welzl-ledbat-survey-00

Michael Welzl

75th IETF Meeting
Stockholm, Sweden
29 July 2009

Introduction

- Intention: Avoid reinventing the wheel

- Classification
  - delay-based (react early to queue growth)
  - non-delay-based (different CA behavior)
  - application layer (may be delay-based or not)
  - orthogonal (other stuff worth mentioning)
    - But maybe not worth discussing in this presentation

Delay-based approaches

- TCP Vegas: was not designed to be LBE,
  - less aggressive then Reno when sharing a link
  - performs better than Reno in its absence
  - nice example case: LBE ≠ worse performance

- Several others exist
  - TCP Nice, TCP-LP

Non-delay-based approaches

- Different window updates with no “delay growth = queue growth” considerations
  - e.g. consider going cwnd = cwnd/4 instead of cwnd = cwnd/2, and growing with 1/(2*cwnd):
you’d be less aggressive than Reno

- Examples: 4CP and MulTFRC (our own, work in progress), which is as TCP-friendly as *n* TCPs, including the possibility of 0 < *n* < 1

Application layer approaches

- Covered so far: rwnd tuning
  - Quite sophisticated approaches exist, e.g. Key, Massoulie, Wang, "Emulating Low-Priority Transport at the Application Layer: a Background Service“, SIGMETRICS ’04.
  - Is this used in Microsoft’s Background Intelligent Transfer Service (BITS)?
- Probably a lot still missing here.

Questions derived from the survey

- Application layer approach could mean that we don’t need several per-transport-protocol specs
  - Quote from the SIGMETRICS paper: “encouraging simulation results suggest that such an application level mechanism can work almost as well as a transport layer scheme like TCP-LP.”
  - Is “almost as well” good enough for LEDBAT?

- For the mechanism itself, will it be good enough to immediately give way, or do we want to quantify “aggression”?

Thank you!