Sender RTT Estimate Option for DCCP

draft-renker-dccp-tfrc-rtt-option-00
Outline

- Motivation
- Presentation
- Conclusion
Motivation

• original TFRC specification required the sender to communicate its RTT to the receiver

• Errata 610/611 change RFC 4342 so that X_recv is now based on RTT
  – over-estimating RTT <=> under-estimating X_recv
  – leads to performance degradation

• CCID-3 RTT estimation uses RTT/4 counter
  – only usable for differences 2..4
  – difference of 5 has "special" semantics
Limitations of CCVal Algorithm

• requires at least 1 sample per RTT
  – problematic for slow senders (audio streaming)
  – CCID-4 (>= 10ms packet gap) in particular

• MP3 sender (sending less than 1 packet per RTT):
  – no suitable samples for over 1 hour!

• test run statistics (38,000 packets in 20 seconds):
  – about 394 usable samples (1 %) with delta = 4
  – about 1702 usable samples (5 %) with 1 <= delta < 4
  – too few samples (aliasing, sub-sampling)!
Presentation: Sender RTT Option

- **sender piggybacks RTT estimate on data**
  - sender measures its RTT as usual *(timestamps)*
  - as per original TFRC proposal
- **negotiated using “Send RTT estimate” feature**
  - Boolean feature
  - per default off *(like an extension)*
  - server priority
- **forward/backward compatible**
RTT Estimate option

- 4 byte value with microsecond resolution
- 0 means: “no suitable estimate yet”
- up to a RTT of 4295 seconds (ca. 1.2 hours)
- permitted on any packet
  - suggested to send this on all data packets
Conclusions

- *easy-to-implement* extension
- *compatible* with existing base
- *sender has greater accuracy available*
  - timestamp / elapsed time option
  - needs to sample anyway
- *affords better & more reliable performance*
Where to go from here

- can we please have an IANA type for this
- need to get started with an implementation
- current receiver estimation very unsatisfactory

Thanks.