DCCP, TFRC & Open Problems in Congestion Control for Media Applications

Tom Phelan
tphelan@sonunsnet.com
13-Feb-2007
ICCRG

What is DCCP?

- UDP with congestion control
  - Maybe more like TCP without reliability
- “Pluggable” congestion control algorithms
  - CCID2, AIMD, TCP-like
  - CCID3, TCP-Friendly Rate Control (TFRC)
- RFC 4340 (main protocol), RFC 4341 (CCID2) RFC 4342 (CCID3)
- Other CCID work in draft:
  - TFRC-SP – TFRC for small packets, draft-floyd-ccid4
  - TFRC Faster Restart, draft-ietf-dccp-faster-restart
  - RFC3448bis, TFRC update, draft-ietf-dccp-rfc3448bis

DCCP and Media Applications

- Well, the real issue is TFRC (or any congestion control) and Media Applications
- Many issues, possible solutions described in draft-ietf-dccp-tfrc-media,
  - "Strategies for Streaming Media Applications Using TCP-Friendly Rate Control"
- Divides streaming media into three classes:
  - One-way, prerecorded
  - One-way, live
  - Two-way, interactive
- One-way apps relatively easily adapted to TFRC
  - But TCP works at least nearly as well
- Two-way apps have problems

What's Different about Two-Way?

- Delay intolerance
  - Conversational – well studied, max 150ms from lips to ears
  - "Remote Control" (program switching, fast-forward, etc.) – not as well studied, something less than one second, more like 500ms
- TFRC model mismatches media encoding practices
  - With several seconds of delay (as in one-way apps), the mismatches can be smoothed over
  - With 100ms delay, no way

What are the Mismatches?

- Some background first…

How is Media Encoded? – Voice

- Analog signal sampled periodically
  - Usually 8K, 8-bit samples per second
    - Derived from PSTN practices
  - Some codecs targeted for IP use have higher rate
    - Usually 16K samples, sometimes more bits/sample
    - Referred to as “wideband” codecs
- Multiple samples gathered into frames
  - Commonly 20ms or 40ms, sometimes 10ms
  - Many other frame periods in use
  - Often biggest contributor to end-to-end delay
- Frame then compressed (or not)
- Add IP, UDP, RTP headers
  - Payload often smaller than headers

Voice Encoding – Silence Suppression

- Voice codes usually generate fixed rate streams
- But a conversation usually is half-duplex
  - Half the time one side is not talking
- Why send silence?
- Silence suppression (also known as Voice Activity Detection, VAD) removes silence
- Gives stream an on-off characteristic
  - Silence block could last seconds
  - Some codecs send "comfort noise" at regular intervals (but less than the frame rate)

Video Encoding

- Video encoded one frame at a time
  - Many frame rates in use:
    - 30 or 25 frames/sec (~33ms or 40ms) for TV quality
    - 60 frames/sec for some HDTV (~17ms)
    - Lower frame rates sometimes used for low-bitrate video, practical minimum about 10 frames/sec (100ms)
- “Index” frames encoded more-or-less like still images
- Other frames (“predictive” frames) encoded as differences from index frames
  - Process called motion compensation
  - Index frames can come before or after predictive frame
    - Using later index frame not suitable for two-way apps
  - There can be very long periods (minutes) between index frames
- Ratio of bits per index frame and bits per predictive frame commonly 10 to 1, can be greater

So, What are the Mismatches?

- Media apps operate at a frame rate that has nothing to do with Round Trip Time
  - Can only make rate adaptations at frame boundaries
- Media apps sometimes make abrupt rate changes at frame boundaries
  - Voice goes from zero to max
    - Faster restart aimed at this problem, but still issues
  - Video never zero, but can vary 10 to 1 from frame to frame

More Mismatches

- Step adaptation vs. smooth adaptation
  - Media apps make rate adjustments in steps, not smoothly
  - TFRC uses smooth rate adjustments
  - Downward, step adaptation can coexist with smooth adaptation, but how do you go up?
- Small packets
  - Voice uses small packets, is perhaps unfairly penalized by packet-rate algorithms
    - TFRC-SP (draft-floyd-dccp-ccid4) address this

More Mismatches

- Greedy vs. self-limiting apps
  - Media apps self-limit, file transfer apps grab everything the network will give them
  - How do you handle the greedy bullies?
    - TFRC reasonably good at this
- Greedy apps don’t work without congestion control
  - Web server on 1G Ethernet would always overwhelm client on dial-up link without it
- Self-limiting apps (usually) work without congestion control
  - User chooses an encoding rate that fits her situation
  - Sometimes users make mistakes…

Conclusions

- Congestion control community has implied application model
- That app model fits many applications, but has mismatches with media application practices
- Those mismatches can be accommodated with the addition of enough delay (several seconds)
- The delay requirements of two-way media apps can’t accommodate the mismatches
- Two-way media apps need CC tailored to the app needs
  - One-way apps would benefit also