Requirements for a CCID for Interactive Media
no draft (yet)

Tom Phelan
tphelan@sonusnet.com
http://www.phelan-4.com/dccp/IETF-64-media-CCID.ppt
History

• Group has interest in interactive media apps
  – draft-burness-dccp-interactive-apps
  – draft-phelan-mfrc
  – draft-phelan-dccp-media (evolved to draft-ietf-dccp-tfrc-media)

• Some overlap with iccrg and tmrg
  – But in Paris we decided dccp was good place to center work
  – Start with requirements

• Mailing list discussions perhaps as precursor to draft
Some Possible Requirements

but not all, and other issues
Delay

- Human-to-human voice apps can tolerate no more than about 150ms delay from lips to ears
  - The limit is one-way, not round trip
  - One-way 250ms, return 10ms doesn’t work
- Human-to-human video limited by voice
  - Channel surfing limit around 500ms round trip
- But how do you phrase a requirement for the CCID?
  - Must not unnecessarily delay transmissions
    - Wishy-washy
TCP-Friendliness

• Competing flows with similar circumstances get order of magnitude equal throughput measured over time periods lasting seconds

• I think this is a requirement
  – DCCP is general-purpose transport, deployable anywhere in Internet

• This has gotten most discussion on mailing list
  – Can’t we trade off our self-limiting discipline against TCP’s grab-what-you-can greed for a bigger than fair share?
  – It hurts us inelastic apps more than the elastic apps, so we should get more
  – How about measuring fairness over longer time periods?
TCP Courage

• Apps that self-limit to less than fair share shouldn’t be driven off
  – Note that TFRC has this characteristic (at least for large packet flows)
Slow Start

- New flows must gently enter network
- Like TCP, TFRC slow start
  - Although not necessarily those exact algorithms
Variable Rate

• Combining Stop/Start and variable rate in this slide
  – Silence suppression for voice
  – Motion compensation for video
• Basic premise – CCID should not force apps to use more bits now in order to protect capability to use more bits later
• Toughest problem, IMHO
What’s Fair?

• How do you judge fair share?
  – Peak rate must be <= fair share?
  – Average rate must be <= fair share?

• TCP-Fairness judged on scale of seconds
  – So average rate seems reasonable

• But peak rate must be limited too
  – Peak rate must be less than lowest link capacity
What’s Fair? (2)

• Requirements (IMHO):
  – Peak rate $\leq$ fair share allowed
  – Average rate $\leq$ fair share better
    • Average over seconds
    • But peak rate must be limited also
  – Peak-to-average ratio may be limited
    • E.g., if actual average less than peak/$X$, use peak/$X$
      $\leq$ fair share
Small Packets

• Fairness judged in bytes/second, not packets/sec
• Bytes/sec are app data plus necessary DCCP and IP headers
  – Could also include MAC header
  – Small packets have more headers, so less app throughput
Return to High Rate

• In addition to reducing the allowed rate during congestion, the CCID must allow increasing the rate when congestion dissipates

• Apps may chose not to return to high rate
  – User perception of variable quality as worse than constant low quality