CSC2209
Computer Networks

DCCP = Unreliable TCP

Stefan Saroiu
Computer Science
University of Toronto
Treat Your Advisor Like Your Grandparents!
Treat Your Advisor Like Your Grandparents!

- Advisors have very short attention spans
- They always forget what they said and to whom

- Every time you see them, spend time:
  - Introducing yourself
  - Telling them what you work on
  - Ignoring them telling the same story over and over again
    - Act surprised!

- Also, make sure you visit them at least once in awhile
Project Proposals

• Next deadline:
  – Friday, November 19th: project report
• Websites up by the end of the week
• Take this seriously!
• Analyzing and Enhancing Content Availability in BitTorrent-like P2P. Nadeem, Jin, Di.
• Large-Scale Measurement of Bluetooth Activity. Ivan, Alireza.
• Non-Interfering VMs. Tom.
• Proxy Support for Application Replay. Vladan.
• Cooperative Browser Caches. Robert.
• Something related to Spoofing & Phishing. Shvet.

• Waqas, Andrew, Fareha?!
Take-away Messages?

- Congestion Management
- DCCP
Take-away Messages?

• Congestion Management
  – Application-level API for congestion control

• DCCP
  – How to build an unreliable, congestion control protocol
What are the uses of DCCP?
What are the uses of DCCP?

• Need congestion control, but no reliability:
  – Streaming media, VoIP, games, video-conferencing

• What does YouTube use?
What are the uses of DCCP?

• Need congestion control, but no reliability:
  – Streaming media, VoIP, games, video-conferencing

• What does YouTube use?
  – Streaming media works just fine with TCP, unless it’s live

• How much live content is there on the Internet?
What are the uses of DCCP?

• Need congestion control, but no reliability:
  – Streaming media, VoIP, games, video-conferencing

• What does YouTube use?
  – Streaming media works just fine with TCP, unless it’s live

• How much live content is there on the Internet?
  – Not much: is video-conferencing the only app?
DCCP Data Transfer Example

- Pure Seq #, not bytes
- Each packet carries a Seq #
- Seq # increase per packet
- Pure Acks also consume Seq #

Slide from Xiafeng Han
DCCP Data Transfer Example

- No Retransmissions
- Acks the largest Seq # received

Slide from Xiafeng Han
DCCP Data Transfer Example

- GSR – Greatest Sequence Number Received
- GSS – Greatest Ack Number Received
- Window Size = 8

Slide from Xiafeng Han
DCCP Data Transfer Example

- **GSR** – Greatest Sequence Number Received
- **GSS** – Greatest Ack Number Received
- **Window Size** = 8

Slide from Xiafeng Han
Two congestion control schemes

- **CCID 2: TCP-like**
  - Slow-start, timeouts like TCP
  - Additionally, congestion control for ACKs
- **CCID 3: equation-based**
  - Receiver measures loss rate and feeds back to sender
  - Sender uses feedback messages to estimate RTT
  - Loss rate and RTT are fed into equation
  - Sender adjusts actual rate to match expected rate
- When should one use CCID 2 vs. CCID 3?
Two congestion control schemes

• CCID 2: TCP-like
  – Slow-start, timeouts like TCP
  – Additionally, congestion control for ACKs

• CCID 3: equation-based
  – Receiver measures loss rate and feeds back to sender
  – Sender uses feedback messages to estimate RTT
  – Loss rate and RTT are fed into equation
  – Sender adjusts actual rate to match expected rate

• When should one use CCID 2 vs. CCID 3?
  – Whenever reacting to congestion must be abrupt
    • How about CBR vs. online games?
  – Many unknown, poorly-understood issues are left
Adding Reliability Back

- How would you add reliability to DCCP?
Adding Reliability Back

• How would you add reliability to DCCP?
  – FEC
  – Bloom filters
  – Likely very different than TCP…

• How about in-order delivery?
Congestion Manager Original Motivation
Congestion Manager Original Motivation

1. Many short Web Xfers
2. Web accelerators
3. Application heterogeneity: many apps over UDP to avoid TCP’s reliability mechanism

- Are any of these motivations valid today?
Congestion Manager Current Motivation
**Congestion Manager Current Motivation**

- Congestion is located at the end-points today
  - CM manages flows to different destinations but sharing same bottleneck
- Should we expose API to applications?