Implementing DCCP

Experience so far…

Tom Phelan
tphelan@sonusnet.com

DCCP-TP

- Fresh start implementation optimized for portability
  - Thinking embedded systems
  - No code from Linux implementation – all original
- Current release R0.00 – early stage
  - Supports core DCCP (RFC 4340, but with DCCP-NAT encap) and CCID 2 (RFC 4341).
  - Doesn't support every feature of DCCP, but a lot
    - Good effort made to ensure supported features are responsibly implemented
    - Corners cut mostly in API at this stage
  - Includes Linux user-space port
    - DCCP service and apps run as single Linux process (with Pthreads)

DCCP-TP (more)

- Near-term next steps:
  - I'm working on CCID 3
  - Jawad Shafi working on Windows user-space port
- Documentation (wiki), source code and discussion forum at:
  - http://www.phelan-4.com/dccp-tp/
- Anyone who wants to contribute, see http://www.phelan-4.com/dccp-tp/tiki-index.php?page=Todo+List and jump in ☺

Observations

- The specs present a number of difficulties for implementers
  - Gerrit Renker's recent post on rfc3448bis hits the nail on the head
  - Almost all features have relevant text scattered about
    - Often need to read multiple sections of two or three different RFCs to get full sense of a feature
  - Many features interact with other features
    - To implement a single feature you need to first understand several other features
  - Difficult to tell normative from informative
  - Conflicting text about optionality/desirability of features
    - Reducing oscillations in rfc3448bis good example

Observations (2)

- Much of the suggested implementation text didn't work for me
  - Issues with fuzziness, memory use, complexity, efficiency often led me to pursue other solutions
  - Coming to an understanding of the suggested implementation helped
- Pseudo-code was great
  - One, shows you how to do it, all in one place
  - Two, getting to understand it teaches that you need to look all over for relevant text
  - But does it really work?
    - I haven't found any problems yet

Observations (3)

- Ack Vectors are difficult
  - Biggest problem is unbounded memory needs
  - Second is complexity
    - One of the features where relevant text is most scattered about
  - And what does all this cost buy you?
    - Knowledge of which packets were lost
    - What good is that?

Conclusions

- DCCP is feature-rich
- Some features are highly complex
  - Easy to go down wrong path if you aren't careful
- Interop between independent implementations will be work