Linux's DCCP—much code, many bugs

Andrea Bittau, UCL

July 12, 2006

Implementation Status

Linux 2.6.17 (current) supports:

- IPv6.
- Feature negotiation.
- Ack vectors. [Most of the bugs lurk here. . .]
- CCID2.
- CCID3.
- Options supported: Padding, Mandatory, NDP count, Timestamp, Elapsed time and CCID specific options.

Implementation Status

Deficiencies in implementation

General:

- Only long sequence numbers are supported.
- ECN is supported but is not integrated with IP yet.

Feature negotiation:

- May not query status of features from user-land. Kernel API exists, but it is not yet hooked up.

CCID2:

- Timer resolution is HZ. On low RTT paths, `srtt` becomes zero and RTO is forced to 1 second.
- Ack ratios are supported but code to change them is commented out.
- Does not detect quiescent flows.

API

DCCP connection overview:

1. `socket(PF_INET, SOCK_DCCP, IPPROTO_DCCP)`
2. `setsockopt(s, SOL_DCCP, DCCP_SOCKOPT_SERVICE ...`
3. `send`, `recv` and `close` as with UDP.

Remarks:

- `SOCK_DCCP` → `SOCK_STREAM`?
- Use a default service code if none supplied (instead of error).

Details:

- Currently, DCCP has no queue—everything is passed on to IP.
- Application frames packets as with UDP.
- Features are changed via setsockopt (non-blocking).
  - User should poll status via getsockopt.

Future Work

Stability:

- Avoid the blue screen when possible.
- Many CCID2 patches queued for review.

Completeness:

- DCCP buffering patch submitted for review.
- Short sequence numbers.
- Any missing options.

CCID2:

- Make it do Gbit speeds on networks with high delay.

Expect Linux 2.4.20 to have a decent DCCP implementation.