DCCP Spec Updates

* * *

[Eddie Kohler, Mark Handley]
UCLA

IETF 59 DCCP Meeting
March 4, 2004

Overview

- Spec looks more different than it is
  - Organizational changes
  - Cleanups from reviewers
- Technical updates
  - Event processing
  - Simplifications discussed in Minneapolis
  - Most significant changes mentioned on mailing list

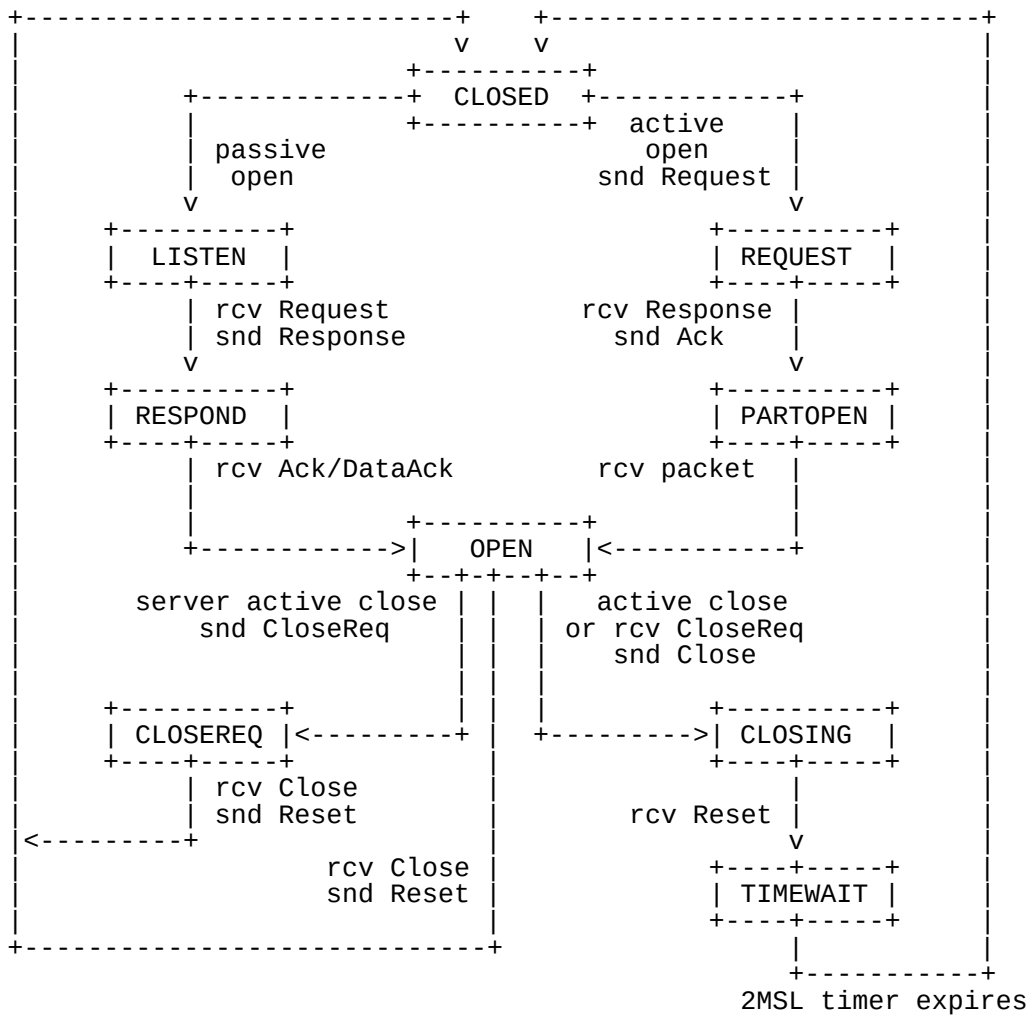
Organizational Changes

* *

- Rewrote initial material
- Reorganized text

  Moved specifics of packet processing, validation, etc, out of Header Processing into new sections
- Changed option names, and in some cases semantics, to improve understandability
- Clearer examples
- New (non-normative) state transition diagram

State Diagram



Event processing

- Added event processing pseudocode
- Specific processing steps for all events
- Improved state diagram

  Added PARTOPEN state: after receiving Response, client must send acknos on all packets until hearing from server

- Checked it out with a finite state model and an exhaustive state walk

Event processing pseudocode

* *

```
....
Eighth, check sequence numbers;
   If S.SWL <= P.seqno <= S.SWH
         && (P.ackno does not exist || S.AWL <= P.ackno <= S.AWH),
      Update S.GSR, S.GAR, S.SWL, S.SWH
   Otherwise,
       Send Sync packet acknowledging P.seqno
       Drop packet and return

Ninth, check packet type;
   If (S.is_server && P.type == CloseReq)
        || (S.is_server && P.type == Response)
        || (S.is_client && P.type == Request)
        || (S.state >= OPEN && P.type == Request && P.seqno >= S.OSR)
        || (S.state >= OPEN && P.type == Response && P.seqno >= S.OSR)
        || (S.state == RESPOND && P.type == Data),
      Send Sync packet acknowledging P.seqno
      Drop packet and return

Tenth, process options;
   /* may involve resetting connection, etc. */
   Mark packet as ‘‘received’’ for acknowledgement purposes
   On processing Confirm R(Mobility ID),
      Check that the confirmed Mobility ID is correct
       If a DCCP-Move was recently processed,
         Remove any old Mobility ID from table
...
```

Sequence number validity

* *

- Cleaner rules depend only on packet type (not connection state)
- Previously a DCCP-Sync elicited a DCCP-Sync

  Not convinced a Sync storm couldn't happen.

  Add DCCP-SyncAck packet type to avoid possible problems.
- Added section calculating probability of successful sequence number guessing attacks.

  Suggest using extended sequence numbers if window is greater than 100 packets.

Sequence number validity

Packet Type	Sequence Number Check	Acknowledgement Number Check
DCCP-Request	SWL <= seqno <= SWH (*)	N/A
DCCP-Response	SWL <= seqno <= SWH (*)	AWL <= ackno <= AWH
DCCP-Data	SWL <= seqno <= SWH	N/A
DCCP-Ack	SWL <= seqno <= SWH	AWL <= ackno <= AWH
DCCP-DataAck	SWL <= seqno <= SWH	AWL <= ackno <= AWH
DCCP-CloseReq	SWL <= seqno <= SWH	AWL <= ackno <= AWH
DCCP-Close	SWL <= seqno <= SWH	AWL <= ackno <= AWH
DCCP-Reset	seqno == 0 or seqno > GSR	GAR <= ackno <= AWH
DCCP-Move	seqno >= SWL	ISS <= ackno <= AWH
DCCP-Sync	seqno >= SWL	AWL <= ackno <= AWH
DCCP-SyncAck	seqno >= SWL	AWL <= ackno <= AWH

- In general, packets are sequence-valid if their Sequence and Acknowledgement Numbers lie within the corresponding valid windows, [SWL, SWH] and [AWL, AWH].

Forward compatibility

* *

- Added Forward Compatibility section

  Describes how features should be defined to facilitate forward and backward compatibility

  1: Use a feature to negotiate the use of an extension, default is “No”

  2: Don’t reset odd options or features

- `Ignored` option proved non-useful, so removed it
- Some existing features were rewritten so they act like extensions:

  Sequence number transition

  Check Data Checksum, . . .

- Also reserve some options and features for experimental use

Feature negotiation

* *

- Added empty Change option

  "What's your current value for this feature?"

- Add empty Confirm option

  "I didn't understand your Change option"

- Both make the protocol more explicit

- Simplified state diagram

  Remove FAILED state—no need to support it if features are implemented as suggested in "Forward compatibility"

Update on open issues from IETF 58

* *

- # NDP

  Removed in favor of NDP Count option

- Identification and Challenge

  Removed in favor of DCCP-Sync and DCCP-SyncAck

- Data Dropped requirements in CCID 3

  Problem is receiver (as opposed to network) congestion

  CCID 3 draft now suggests manipulating X_{recv} to indirectly limit the transmit rate.

Update on open issues 2

- Packet sizes

  "CCID x implementations MAY check for applications that appear to be manipulating the packet size inappropriately."

- Payload Checksum

  Use SCTP's CRC-32c

- Service Code Wildcarding

  Previously allowed DCCP-Request and/or listening socket to wildcard the service code.

  Potential security confusion.

  Dropped wildcarding, echo service code in DCCP-Response

CCID 2 and 3

* *

- No other significant changes

So where are we?

* *

- Rev documents, suggest real WG last call immediately after IETF
- Onward and upward

Future Work

* *

- Faster recovery after idle.
- CCID for TFRC-PS

  TFRC-PS needs doing in TSVWG

- Fixed rate apps.

Faster recovery after idle

- Open issue as to what the bad consequences are from not slow-starting when a session becomes active again after an idle period.

TFRC-PS

* *

- TFRC is designed for applications that change their sending rate by varying the number of packets sent per second.

  Audio applications generally want to send a constant rate of packets/second, and change the compression of each of those packets.

- Research is still needed as to how to modify TFRC to do this safely.

  Depending on this research, we need to create a new CCID for TFRC-PS.

Fixed rate applications

* *

- DCCP as currently written assumes data will be transmitted at the congestion-controlled rate.

  Some applications are inherently fixed rate.

  Some applications have a number of fixed rates they can switch between.

- It should be possible to use TFRC to provide a *reference rate*.

  DCCP would tell the application the reference rate, and police the application only if went outside a fairly wide band centered on the reference rate.

  Perhaps: $0.5X_{reference} < X_{app} < 2X_{reference}$

  May be issues when few flows stat-muxing - need research.