DCCP Specification Update

Mark Handley

International Computer Science Institute

Berkeley, California.

Changes since Yokohama

- Statement on TCP friendliness
- Rewrite of Challenge mechanism.
- Change of Reset packet format.
- Buffer closed signalling.
- Clarification of meta-data processing at receiver.

TCP Compatibility

Old version used to say in several places that congestion control schemes needed to be TCP friendly.

New Wording:
All CCIDs standardized for use with DCCP will correspond to congestion control mechanisms previously standardized by the IETF. We expect that for quite some time, all such mechanisms will be TCP-friendly, but TCP-friendliness is not an explicit DCCP requirement.

DCCP Challenge Mechanism.

The DCCP Challenge mechanism is used in DCCP-level mobility, and to resynchronize after a prolonged outage where the sender and receivers sequence number expectations become desynchronized.

DCCP Challenge Mechanism.

The DCCP Challenge mechanism is used in DCCP-level mobility, and to resynchronize after a prolonged outage where the sender and receivers sequence number expectations become desynchronized.

Problem: the old challenge mechanism didn’t work.

DCCP Challenge Mechanism.

We now allow multiple Challenge Regimes - previously there was no way to use alternatives to the default.

The default is a non-cryptographic authentication scheme based on nonces. Before any move or resynchronization attempt, each endpoint uses the option mechanism to send a randomly chosen nonce to the other endpoint.

DCCP Challenge Mechanism.

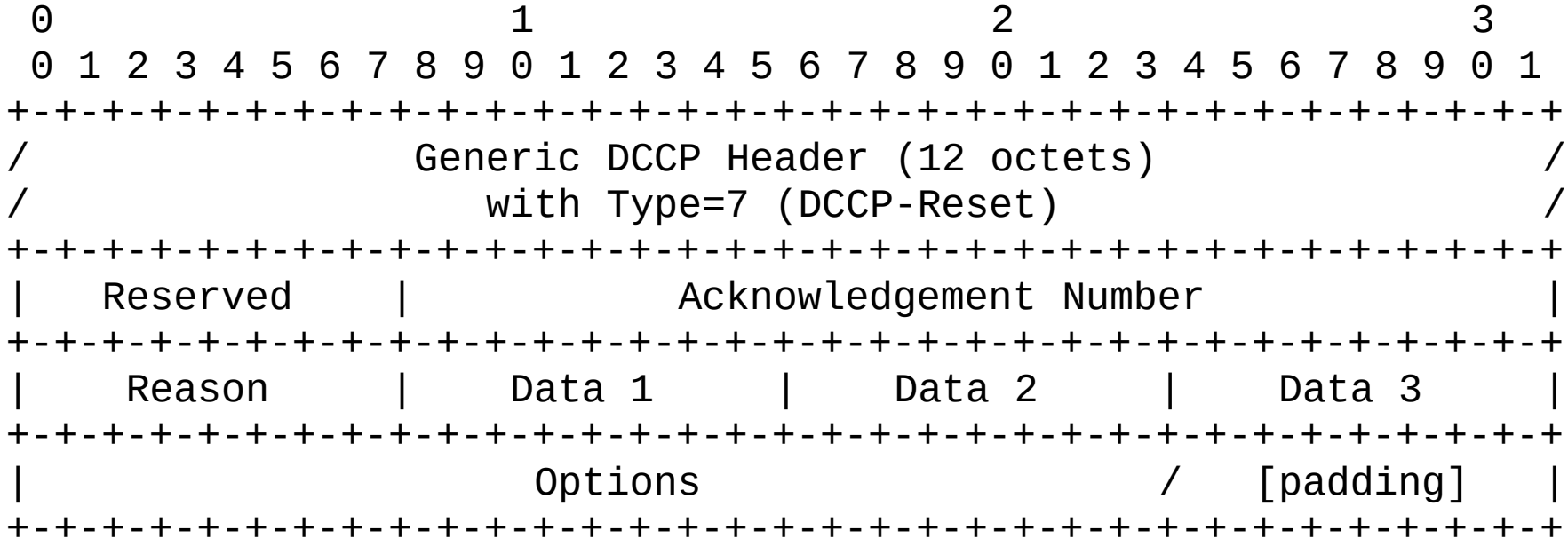
Simple challenge between host A and host B

A's nonce is N_A, B's nonce is N_B.

- A sends (N_A XOR N_B) to B in a Challenge Option.
- B sends N_B in a Challenge Response Option.

Note: B must not send a Challenge Response unless the Challenge is good and the source address of the Challenge is that of A.

DCCP Reset Packet



DCCP Reset Packet Reasons

Reason	Name	Data 1	Data 2	Data 3
0	Unspecifi ed	N/A	N/A	N/A
1	Closed	N/A	N/A	N/A
2	Invalid Packet	packet type	N/A	N/A
3	Option Error	option number	option data	
4	Feature Error	feature number	feature data	
5	Connection Refused	N/A	N/A	N/A
6	Bad Service Name	N/A	N/A	N/A
7	Too Busy	N/A	N/A	N/A
8	Bad Init Cookie	N/A	N/A	N/A
9	Invalid Move	N/A	N/A	N/A
10	Unanswered Challenge	N/A	N/A	N/A
11	Fruitless Negotiation	feature number	N/A	N/A

Buffer Closed Option

Previous version had a Buffer Closed Drops option. This has been removed.

Current version has a simple Buffer Closed option which indicates that the application is no longer accepting any more data on this half-connection.

Receive Buffer Behaviour

With respect to the ACK mechanisms:

- Packets ACKed as being “received” MUST be delivered to the application unless explicitly dropped due to application intervention.
- Packets ACKed as “not yet received” MUST NOT have been processed by DCCP (especially feature negotiations, options and ACK number).
- Packets dropped due to receive buffer overflow SHOULD be acknowledged as “not yet received” - they MOST NOT have previously been acked as being received. The Receive Buffer Drops option distinguishes this from network congestion loss.

Summary

- Good progress made.
- Mechanisms are stabilizing, and corner cases specified.
- More implementation experience still needed.
- Open Issues:
  - RTP over DCCP