Deleted: DTLS over DCCP

Deleted: April 14, 2008

1. Introduction

The Datagram Congestion Control Protocol (DCCP) [RFC4340] is both datagram-based and connection-oriented. To establish connections, DCCP servers passively listen for incoming connection requests that are actively transmitted by DCCP clients. These asymmetric roles can cause problems when the server is 'inside' a middlebox, such as a Network Address and Port Translator (NAPT) [ID-BEHAVE-DCCP] that only allows connection requests to be initiated from inside.

UDP can handle this situation through the use of various techniques that leverage the connectionless nature of UDP and are therefore not suitable for DCCP. TCP functions in this case through the use of its simultaneous open procedure. The concepts of the TCP solution are applicable to DCCP.

Deleted: As such, it faces the same problems as TCP in sending packets through devices that require all connections to be initiated by a 'trusted' host. This is true of host-based firewalls, the default policy on many firewalls, and (due to port overloading) Network Address Port Translators, NAPTs [ID-BEHAVE-DCCP]. DCCP can not simply reuse traversal solutions that work for UDP. In addition, the original specification of DCCP did not allow a server to perform a simultaneous-open, an inherent characteristic of TCP that greatly simplifies TCP Network Address Translator (NAT) traversal.

After discussing the problem space for DCCP, this document specifies an update to the DCCP state machine to offer native DCCP support for connections to DCCP servers that are inside middleboxes. This reduces dependence on external aids such as data relay servers [ID-BEHAVE-TURN] by explicitly supporting a widely used principle known as 'hole punching'.

The method requires only a minor change to the standard DCCP operational procedure. The use of a dedicated DCCP packet type ties usage to a specific condition, ensuring the method is inter-operable with hosts that do not implement this update, or choose to disable it (see Section 4).

1.1. Scope of this Document

This method is useful in scenarios when a DCCP server is located inside a middlebox. It is relevant to both client/server and peer-to-peer applications, such as VoIP, file sharing, or online gaming and assists connections that utilise prior out-of-band signaling (e.g. via a well-known rendezvous server ([RFC3261], [H.323])) to notify both endpoints of the connection parameters ([RFC3235], [NAT-APP]).

Deleted: middlebox traversal

Deleted: behind

1.2. DCCP NAT Traversal

The behavioural requirements for NAT devices supporting DCCP are described in [ID-BEHAVE-DCCP]. A "traditional NAT" [RFC3022], that directly maps an IP address to a different IP address does not require the simultaneous open method described in this document.

The method is required when the DCCP server is positioned behind one or more NAPT devices in the path (hierarchies of nested NAPT devices are possible). This document refers to DCCP hosts located inside one or more NAPT devices as having "private" addresses, and to DCCP hosts

Deleted: behind

Deleted: Phelan Expires - October 2008

located in the global address realm (outside of any middleboxes) as having "public" addresses.

DCCP NAT traversal is considered for the following scenarios:

1. Private client connects to public server.

2. Public client connects to private server.

3. Private client connects to private server.

A defining characteristic of NAPT devices [RFC3022] is that private hosts can connect to external hosts, but not vice versa. Hence the case (1) is possible using the protocol defined in [RFC4340]. A pre-configured, static NAT address map would allow connections in cases (2) and (3).

A DCCP implementation conforming to [RFC4340] and a NAT device conforming to [ID-BEHAVE-DCCP] would require a DCCP relay server to perform NAT traversal for cases (2) and (3). This document describes a method to support cases (2) and (3) without a relay server. This method requires the DCCP server to discover the IP address and the DCCP port that correspond to the DCCP client. Such signalling may be performed out-of-band (e.g. using SDP [RFC4566]).

1.3. Structure of this Document

For background information on existing NAT traversal techniques, please consult Appendix A.

The normative specification of the update is presented in Section 2. An informative discussion of underlying design decisions then follows, in Section 3. Security considerations are provided in Section 4 and IANA considerations in the concluding Section 5.

2. Procedure for Near-Simultaneous Open

This section is normative and specifies the simultaneous-open technique for DCCP. It updates the connection-establishment procedures of [RFC4340].

2.1. Conventions and Terminology

The document uses the terms and definitions provided in [RFC4340]. Familiarity with this specification is assumed. In particular, the following convention from ([RFC4340], 3.2) is used:

"Each DCCP connection runs between two hosts, which we often name DCCP A and DCCP B. Each connection is actively initiated by one of the hosts, which we call the client; the other, initially passive host is called the server."

Deleted: DTLS over DCCP

Deleted: April 14, 2008

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2.2. Protocol Method

The term "session" is used as defined in ([RFC2663], 2.3): DCCP sessions are uniquely identified by the tuple of <source IP-address, source port, target IP-address, target port>.

DCCP, in addition, introduces service codes, which can be used to identify different services available via the same port [ID-DCCP-SC].

2.2.1. DCCP-Listen Packet Format

This document adds a new DCCP packet type, DCCP-Listen, whose format is shown below.

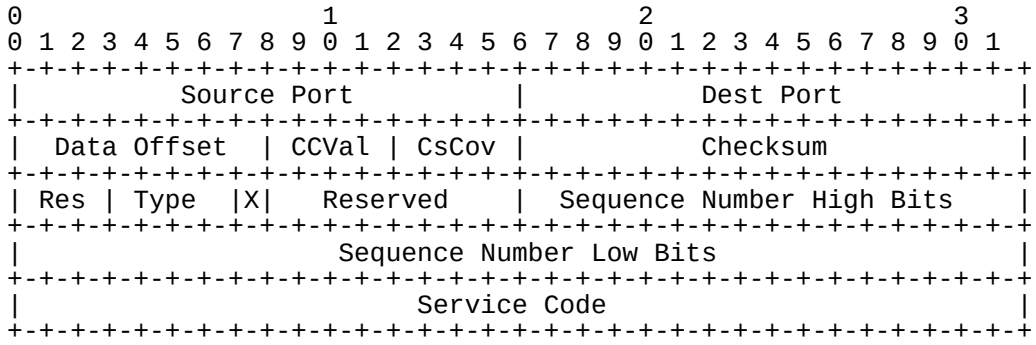

Figure 1: Format of a DCCP-Listen Packet

Value for the Dest Port field seems to be missing.

o The Source Port is the port on which the DCCP server is listening for a connection from the IP address that appears as the destination IP address in the packet.

o The value of X MUST be set to 1. A DCCP-Listen packet is sent before a connection is established, therefore there is no way to negotiate use of short sequence numbers ([RFC4340], 5.1).

o The sequence number of a DCCP-Listen packet is not related to the DCCP sequence number for normal DCCP messages Section 3. Thus, for DCCP-Listen packets:

  * A DCCP server should set the high and low bits of the Sequence Number field to 0.

Deleted: Phelan Expires - October 2008

Deleted: DTLS over DCCP

Deleted: April 14, 2008

* A DCCP client MUST ignore the value of the Sequence Number field.

* Middleboxes MUST NOT interpret sequence numbers on DCCP-Listen packets.

o The Service Code field contains the service code value for which the server is listening for connections([RFC4340], 8.1.2). This value MUST correspond to a service code that the server is actually offering for a connection identified by the same source IP address and the same Source Port as that of the DCCP-Listen packet. Since the server may use multiple service codes, the specific value of the Service Code field needs to be communicated out-of-band, from client to server, prior to sending the DCCP-Listen packet, e.g. described using the Session Description Protocol, SDP [RFC4566].

Deleted: a

o At the time of writing, there are no known uses of header options ([RFC4340] , sec. 5.8) with a DCCP-Listen packet. Clients MUST ignore all options in received DCCP-Listen packets. Therefore, feature values can not be negotiated using a DCCP-Listen packet.

Deleted: the

o At the time of writing, there are no known uses of payload data with a DCCP-Listen packet. Since DCCP-Listen packets are issued before an actual connection is established, endpoints MUST ignore any payload data encountered in DCCP-Listen packets.

o The following protocol fields are required to have specific values:

* Data Offset MUST have a value of five or more (i.e. at least 20 bytes).

* CCVal MUST be zero (a connection has not been established).

* CsCov MUST be zero (if there is no payload). Why only if there is no payload? Since there's been no negotiation of the Checksum Coverage feature, always zero sounds like the right thing to me.

* Type has the value 10, assigned by IANA to denote a DCCP-Listen packet.

* X MUST be 1 (the Generic header must be used).

The remaining fields, including the "Res" and "Reserved" fields are specified by [RFC4340] and its successors. The interpretation of these fields is not modified by this document.

Note to the RFC Editor:

Deleted: Phelan Expires - October 2008

This value assigned to the DCCP-Listen packet needs to be confirmed by IANA when this document is published. Please then remove this note.

==> End of note to the RFC Editor. <==

2.2.2. Protocol Method for DCCP Server Endpoints

This document updates section 8.1 of [RFC4340] for the case of a fully specified DCCP server endpoint. The update modifies the way the server performs a passive-open.

Prior to connection setup, it is common for DCCP server endpoints to not be fully specified: before the connection is established, a server usually only specifies the destination port and Service Code, leaving the destination address and source IP-address:port unspecified (sometimes the destination address is specified); the endpoint only becomes fully specified after performing the handshake with an incoming connection. For such cases, this document does not update [RFC4340], i.e. the server adheres to the existing state transitions in the left half of Figure 2 (CLOSED => LISTEN => RESPOND).

Deleted: sets the target IP-address:port to wildcard values (i.e. leaves these unspecified)

A fully specified DCCP server endpoint permits exactly one client, identified by source IP-address:port, destination IP-address:port plus a single service code, to set up the connection. Such a server SHOULD perform the actions and state transitions shown in the right half of Figure 2 in section 8.4, and specified below.

Deleted: target

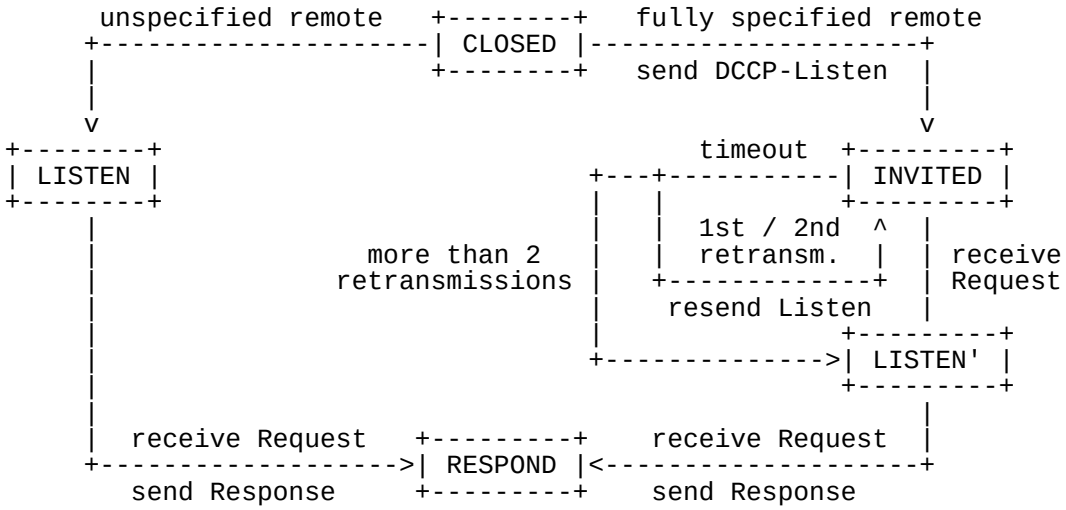

Figure 2: Updated state transition diagram for DCCP-Listen

Should there be a 'v' on the line from INVITED to LISTEN'?

This diagram introduces two additional DCCP server states in addition to those listed in section 4.3 of [RFC4340]:

o INVITED The INVITED state is associated with a specific DCCP connection and represents a fully-specified server socket in the listening state that is generating DCCP-Listen packets towards the client.

o LISTEN' The LISTEN' state is associated with a specific DCCP connection and represents a fully-specified server socket in the passive listening state that will not generate further DCCP-Listen packets towards the client.

A fully specified server endpoint performs a passive-open from the CLOSED state by inviting the remote client to connect. This is performed by sending a single DCCP-Listen packet to the specified remote IP-address:port, using the format specified in Section 2.2.1. The figure below provides pseudocode describing the packet processing in the INVITED state. This processing step follows Step 2 in section 8.5 of [RFC4340]).

The INVITED state is, like LISTEN, a passive state, characterised by waiting in the absence of an established connection. If the server endpoint in state INVITED receives a DCCP-Request that matches its specification, it transitions to the LISTEN' state and then immediately transitions to the RESPOND state, where further processing resumes as specified in [RFC4340].

Deleted: DTLS over DCCP

Deleted: April 14, 2008

Deleted: ,

Deleted: Phelan Expires - October 2008

Deleted: DTLS over DCCP

Deleted: April 14, 2008

The server SHOULD repeat sending a DCCP-Listen packet while in the INVITED state, at a 200 millisecond interval with up to at most 2 repetitions (Section 3 discusses this choice of time interval). If the server is still in the INVITED state after a further period of 200ms following transmission of the third DCCP-Listen packet, it SHOULD progress to the LISTEN' state.

Fully specified server endpoints SHOULD treat ICMP error messages received in response to a DCCP-Listen packet as "soft errors" that do not cause a state transition. Reception of an ICMP error message as a result of sending a DCCP-Listen packet does not necessarily indicate a failure of the following connection request, and therefore should not result in a server state change. This reaction to soft errors exploits the valuable feature of the Internet that for many network failures, the network can be dynamically reconstructed without any disruption of the endpoints.

Server endpoints SHOULD ignore any incoming DCCP-Listen packets. A DCCP server in the LISTEN, INVITED, or LISTEN' states MAY generate a DCCP-Reset packet (Code 7, "Connection Refused") in response to a received DCCP-Listen packet. This DCCP-Reset packet is an indication that two servers are simultaneously awaiting connections on the same port.

Further details on the design rationale are discussed in Section 3.

The figure below provides pseudocode describing the packet processing in the INVITED state. This processing step follows Step 2 in section 8.5 of) [RFC4340]

Deleted: ,

Deleted: Phelan . Expires - October 2008

Deleted: DTLS over DCCP

Deleted: April 14, 2008

```
   Step 2a: Process INVITED state
       If S.state == INVITED,
          /* State only entered for fully-specified server endpoints */
          /* on entry S will have been set to a socket */
          If P.type == Request or S.DCCPListencount=3
```

Is the 'or' condition needed? And what is DCCPListencount anyway? When it's 3, does that mean that you've sent 3 Listens? Don't you got to LISTEN1 after sending the third Listen? This seems to require that you receive *something* to exit INVITED. I think eliminate this.

```
             /* Exit INVITED state and continue to  process the packet*/
             S.state = LISTEN1
             Continue with S.state := LISTEN1
          Otherwise,
             If P.type == Listen
                /* The following line is optional */
                Generate Reset(Connection Refused)
                /* otherwise Drop packet and return */
             Otherwise,
                Generate Reset(No Connection) unless P.type == Reset

   Step 2b: Process LISTEN1 state
      If S.state == LISTEN1,
          /* State only entered for fully-specified server endpoints */
          /* Follows receipt of a Response packet */
          /* or sending third Listen packet */
          If P.type == Request,
              S.state = RESPOND
              Choose S.ISS (initial seqno) or set from Init Cookies
              Initialize S.GAR := S.ISS
              Set S.ISR, S.GSR, S.SWL, S.SWH from packet or Init Cookies
              Continue with S.state == RESPOND
              /* A Response packet will be generated in Step 11 */
          Otherwise,
           If P.type == Listen
                 /* The following line is optional */
                 Generate Reset(Connection Refused)
                 /* otherwise Drop packet and return */
           Otherwise,
                 Generate Reset(No Connection) unless P.type == Reset
```

Deleted: INVITED

Figure 3: Updated DCCP pseudocode for INVITED and LISTEN' states

2.2.3.

This document updates section 8.1.1 of [RFC4340], by adding the following rule for the reception of DCCP-Listen packets by clients:

A client in any state MUST silently discard any received DCCP-Listen packet, unless it implements the optional procedure defined in the following section.

Deleted: Phelan Expires - October 2008

Deleted: DTLS over DCCP

Deleted: April 14, 2008

2.2.3.1. Optional generation of Triggered Requests

This section describes an optional optimisation at the client that can avoid clients having to wait for a timeout following a dropped DCCP-Request. The operation requires clients to respond to reception of DCCP-Listen packets when received in the REQUEST state. DCCP-Listen packets MUST be silently discarded in all other states.

A client implementing this optmisation MAY immediately perform a retransmission of a DCCP-Request following the reception of the first DCCP-Listen packet. The retransmission is performed in the same manner as a timeout in the REQUEST state [RFC4340]. A triggered retransmission SHOULD result in the client increasing the exponential-backoff timer interval.

Note that a path delay greater than 200ms will result in multiple DCCP-Listen packets arriving at the client before a DCCP-Response is received. Clients MUST therefore perform only one such retransmission for each DCCP connection. This requires maintaining local state (e.g. one flag per connection)

Implementors and users of this optional method need to be aware that host timing or path reordering can result in a server receiving two DCCP-Requests (i.e., the server sending one unnecessary packet). This would, in turn, trigger a client to send a second corresponding DCCP-Response (also unnecessary). These additional packets are not expected to modify or delay the DCCP open procedure [RFC4340].

Section 2.3 provides examples of the use of triggered retransmission.

Deleted: 2.

2.2.4. Processing by Routers and Middleboxes

DCCP-Listen packets do not require special treatment and should thus be forwarded end-to-end across Internet paths, by routers and middleboxes alike.

Middleboxes may utilise the connection information (address, port, service code) to establish local forwarding state. The DCCP-Listen packet carries the necessary information to uniquely identify a DCCP session in combination with the source and destination addresses (found in the enclosing IP-header), including the DCCP service code value [ID-DCCP-SC]. The processing of the DCCP-Listen packet by NAT devices is specified in [ID-BEHAVE-DCCP].

2.3. Examples of Use

In the examples below, DCCP A is the client and DCCP B is the server. A middlebox device (NAT/Firewall), NA is placed before DCCP A, and another middlebox, NB, is placed before DCCP B. Both NA and NB use a policy that permits DCCP packets to traverse the device for outgoing

Deleted: Phelan Expires - October 2008

links, but only permit incoming DCCP packets when a previous packet has been sent out for the same connection.

In the figure below, DCCP A and DCCP B decide to communicate using an out-of-band mechanism (in this case labelled SDP), whereupon the client and server are started. DCCP B actively indicates its listening state by sending a DCCP-Listen message. This fulfils the requirement of punching a hole in NB (also creating the binding to the external address and port) although it is dropped by NA since no hole exists there yet. DCCP A initiates a connection by entering the REQUEST state and sending a DCCP-Request. (It is assumed that if NA were a NAT device, then this would also result in a binding that maps the pinhole to the external address and port.) The DCCP-Request is received by DCCP B, via the binding at NB. DCCP B transmits the DCCP-Response and connects through the bindings now in place at NA and NB.

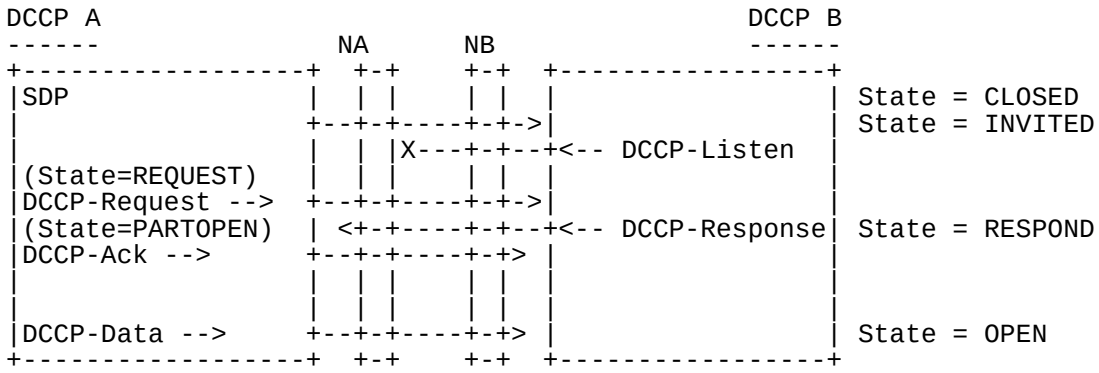

Figure 4: Event sequence when the server is started before the client

2.3.1. Repetition of DCCP-Listen

This section examines the effect of not receiving the DCCP-Request.

The figure below shows the sequence of packets where the DCCP server enters the INVITED state after reception of out-of-band signaling (e.g. SDP). The key timer operations at the client and server are respectively shown on the left and right of the diagram. It considers the case when the server does not receive a DCCP-Request within the first 600 ms (often the request would be received within this interval).

The repetition of DCCP-Listen packets may be implemented using a timer. The timer is restarted with an interval of 200ms when sending each DCCP-Listen packet. It is cancelled when the server leaves the INVITED state. If the timer expires after the first and second transmission, it triggers a transmission of another DCCP-Listen

Deleted: DTLS over DCCP
Deleted: April 14, 2008
Deleted: Listen
Deleted: A
Deleted: A
Deleted: A can
Deleted: re
Deleted: Request
Deleted: Phelan Expires - October 2008

Packet. If it expires after sending the third DCCP-Listen packet, the server leaves the INVITED state, to enter the LISTEN' state (where it passively waits for a DCCP-Request).

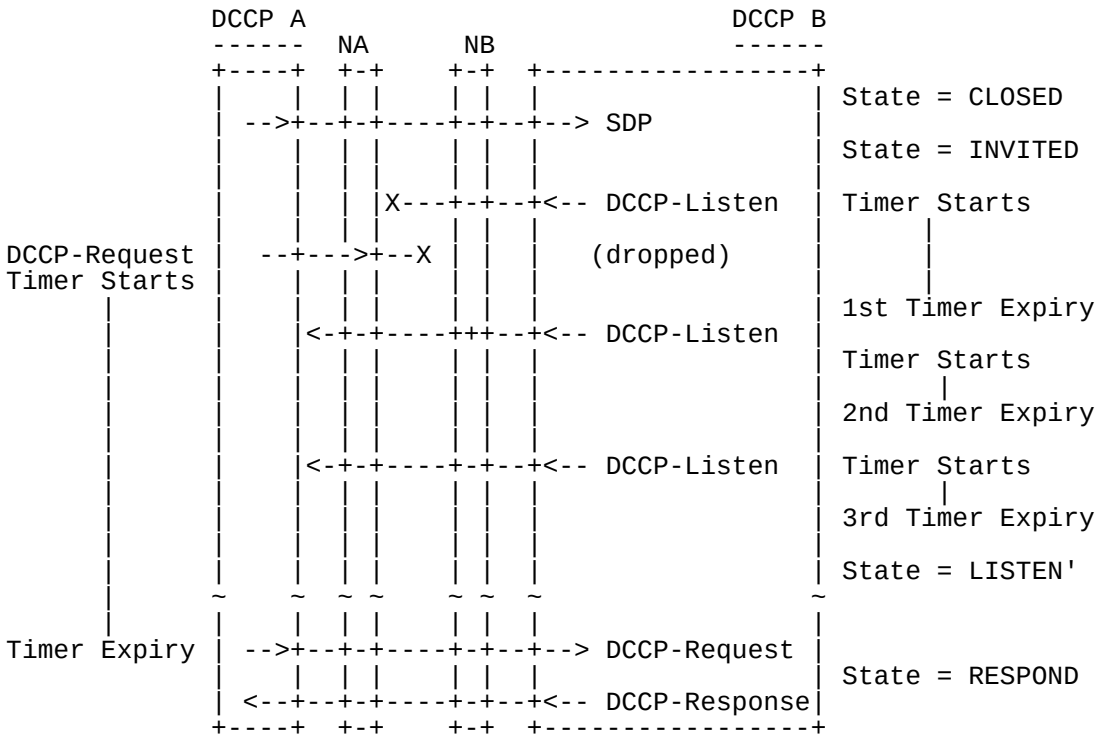

Figure 5: Repetition of the DCCP-Listen Packet

2.3.2. Optional Triggered Retransmission of DCCP-Request

The following figure illustrates a triggered retransmission. In this figure, the first DCCP-Listen is assumed to be lost in the network (e.g. does not open a pin-hole at NB). A later DCCP-Request is also not received (perhaps as a side-effect of the first loss). After 200ms, the DCCP-Listen packet is retransmitted and correctly received. This triggers the retransmission of the DCCP-Request, which, when received, results in a corresponding DCCP-Response.

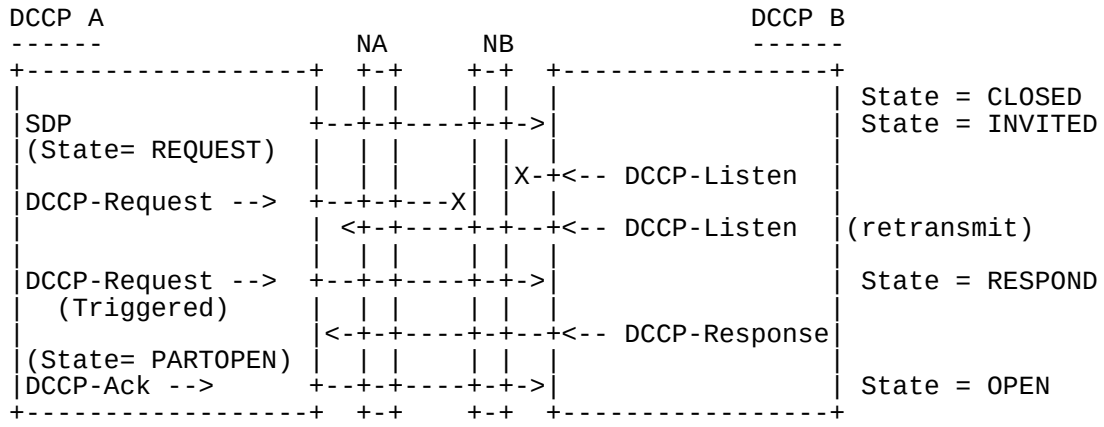

Figure 6: Example showing a triggered DCCP-Request

The figure below illustrates the sequence of packets exchanged when a DCCP-Listen and DCCP-Request are processed out of order. Reception of the DCCP-Listen packet by the client triggers retransmission of the DCCP-Request. The server responds to the first DCCP-Request, and enters the RESPOND state. The server subsequently responds to the second DCCP-Request with another DCCP-Response, which is ignored by the client (already in the PARTOPEN state).

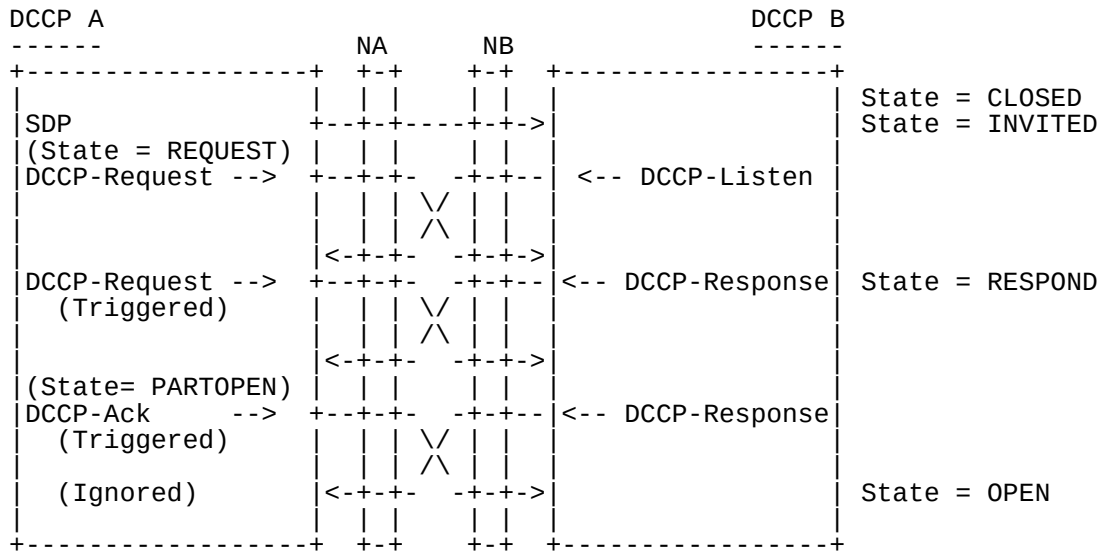

Figure 7: Example showing an unnecessary triggered DCCP-Request

Deleted: DTLS over DCCP
Deleted: April 14, 2008
Deleted: unecessary
Deleted: Phelan Expires - October 2008

Deleted: DTLS over DCCP

Deleted: April 14, 2008

2.4. Backwards Compatibility with RFC 4340

No changes are required if a DCCP client conforming to this document communicates with a DCCP server conforming to [RFC4340].

If a client implements only [RFC4340], an incoming DCCP-Listen packet would be ignored due to step 1 in [RFC4340], 8.1, which at the same time also conforms to the behaviour specified by this document.

This document further does not modify communication for any DCCP server that implements a passive-open without fully binding the addresses, ports and service codes to be used. The authors therefore do not expect practical deployment problems with existing conformant DCCP implementations.

3. Discussion of Design Decisions

This is an informative section that reviews the rationale for the design of this method.

3.1. Rationale for a New Packet Type

The DCCP-Listen packet specified in Section 2.2.1 has the same format as the DCCP-Request packet ([RFC4340], 5.1), the only difference is in the value of the Type field. The usage, however, differs. The DCCP-Listen packet serves as an advisory message, not as part of the actual connection setup: sequence numbers have no meaning, and no payload can be communicated.

A DCCP-Request packet could in theory also have been used for the same purpose. The following arguments were against this:

The first problem was that of semantic overloading: the DCCP-Request defined in [RFC4340] serves a well-defined purpose, being the initial packet of the 3-way handshake. Additional use in the manner of a DCCP-Listen packet would have required DCCP processors to have had two different processing paths: one where a DCCP-Request was interpreted as part of the initial handshake, and another where the same packet was interpreted as an indicator message. This would complicate packet processing in hosts and in particular stateful middleboxes (which may have restricted computational resources).

The second problem is that a client receiving a DCCP-Request from a server could generate a DCCP-Reset packet if it had not yet entered the REQUEST state (step 7 in [RFC4340], 8.5). The method specified in this document lets client endpoints ignore DCCP-Listen packets. Adding a similar rule for the DCCP-Request packet would have been cumbersome: clients would not have been able to distinguish between a Request meant to be an indicator message and a genuinely erratic connection initiation.

Deleted: Phelan Expires - October 2008

The third problem is similar, and refers to a client receiving the indication after having itself sent a (connection-initiation) DCCP-Request. Step 7 in section 8.5 of [RFC4340] requires the client to reply to an "indicator message" Request from the server with a DCCP-Sync. Since sequence numbers are ignored for this type of message, additional and complex processing would become necessary: either to ask the client not to respond to a DCCP-Request when the request is of type "indicator message"; or ask middleboxes and servers to ignore Sync packets generated in response to "indicator message" DCCP-Requests. Furthermore, since no initial sequence numbers have been negotiated at this stage, sending a DCCP-SyncAck would not be meaningful.

The use of a separate packet type therefore allows simpler and clearer processing.

3.1.1. Use of sequence numbers

Although the DCCP-Listen sequence number fields are ignored, they have been retained in the DCCP-Listen packet header to reuse the generic header format from section 5.1 of [RFC4340].

DCCP assigns a random initial value to the sequence number when a DCCP connection is established [RFC4340]. However, a sender is required to set this value to zero for a DCCP-Listen packet. Both clients and middleboxes are also required to ignore this value.

The rationale for ignoring the sequence number fields of DCCP-Listen packets is that at the time the DCCP-Listen is exchanged, the endpoints have not yet entered connection setup: the DCCP-Listen packet is sent while the server is still in the passive-open (INVITED) state, i.e. it has not yet allocated state, other than binding to the client's IP-address:port and service code.

3.2. Generation of Listen Packets

Since DCCP-Listen packets solve a particular problem with NAT and/or firewall traversal, the generation of DCCP-Listen packets on passive sockets is tied to a condition (binding to an a priori known remote address and service code) to ensure this does not interfere with the general case of "normal" DCCP connections (where client addresses are generally not known in advance).

Deleted: (
Deleted:)

In the TCP world, the analogue is a transition from LISTEN to SYN_SENT by virtue of sending data: "A fully specified passive call can be made active by the subsequent execution of a SEND" ([RFC0793], 3.8). Unlike TCP, this update does not perform a role-change from passive to active. Like TCP, DCCP-Listen packets are only sent by a DCCP-server when the endpoint is fully specified (Section 2.2).

3.3. Repetition of DCCP-Listen Packets

Repetition is a necessary requirement, to increase robustness and the chance of successful connection establishment when a DCCP-Listen packet is lost due to congestion, link loss, or some other reason.

The decision to recommend a maximum number of 3 timeouts (2 repeated copies of the original DCCP-Listen packet) results from the following considerations: The repeated copies need to be spaced sufficiently far apart in time to avoid suffering from correlated loss. The interval of 200 ms was chosen to accommodate a wide range of wireless and wired network paths.

Another constraint is given by the retransmission interval for the DCCP-Request ([RFC4340], 8.1.1). To establish state, intermediate systems need to receive a (retransmitted) DCCP-Listen packet before the DCCP-Request times out (1 second). With three timeouts, each spaced 200 milliseconds apart, the overall time is still below one second. On the other hand, the sum of 600 milliseconds is sufficiently large to provide for longer one-way delays, such as e.g. found on some wireless links.

The rationale behind transitioning to the LISTEN' state after two repetitions is that other problems, independent of establishing middlebox state, may occur (such as delay or loss of the initial DCCP-Request). Any late or retransmitted DCCP-Request packets will then still reach the server allowing connection establishment to successfully complete.

4. Security Considerations

The method specified in this document generates a DCCP-Listen packet addressed to a specific DCCP client. This exposes the state of a DCCP server that is in a passive listening state (i.e. waiting to accept a connection from a known client).

The exposed information is not encrypted and therefore could be seen on the network path to the DCCP client. An attacker on this return path could observe a DCCP-Listen packet and then exploit this by spoofing a packet (e.g. DCCP-Request, DCCP-Reset) with the IP addresses, DCCP ports, and service code that correspond to the values to be used for the connection. As in other on-path attacks, this could be used to inject data into a connection or to deny a connection request. A similar on-path attack is also possible for any DCCP connection, once the session is initiated by the client ([RFC4340], Section 18).

The DCCP-Listen packet is only sent in response to explicit prior out-of-band signaling from a DCCP client to the DCCP server (e.g. SDP [RFC4566]) information communicated via the Session Initiation Protocol [RFC3261]), and will normally directly precede a DCCP-Request sent by the client (which carries the same information).

This update does not significantly increase the complexity or vulnerability of a DCCP implementation that conforms to [RFC4340]. A DCCP server SHOULD therefore by default permit generation of DCCP-Listen packets. A server that wishes to prevent disclosing this information MAY refrain from generating DCCP-Listen packets, without impacting subsequent DCCP state transitions, but possibly inhibiting middlebox traversal.

5. IANA Considerations

The IANA should register a new Packet Type, "DCCP-Listen", in the IANA DCCP Packet Types Registry. The decimal value 10 has been assigned to this types. This registry entry must reference this document.

Note to the RFC Editor:

This value must be confirmed by IANA in the registry when this document is published, please then remove this note.