DCCP-NAT Encapsulation
Internet Draft   T. Phelan
Document: draft-phelan-dccp-natencap-01.txt   Sonus Networks
Expires: January 2009   July 10, 2008
Intended status: Proposed Standard

                 Datagram Congestion Control Protocol (DCCP)
                  Encapsulation for NAT Traversal (DCCP-NAT)

   Status of this Memo

   By submitting this Internet-Draft, each author represents that any
   applicable patent or other IPR claims of which he or she is aware
   have been or will be disclosed, and any of which he or she becomes
   aware will be disclosed, in accordance with Section 6 of BCP 79.

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF), its areas, and its working groups. Note that
   other groups may also distribute working documents as Internet-
   Drafts.

   Internet-Drafts are draft documents valid for a maximum of six months
   and may be updated, replaced, or obsoleted by other documents at any
   time. It is inappropriate to use Internet-Drafts as reference
   material or to cite them other than as "work in progress."

   The list of current Internet-Drafts can be accessed at
   http://www.ietf.org/ietf/1id-abstracts.txt.

   The list of Internet-Draft Shadow Directories can be accessed at
   http://www.ietf.org/shadow.html.

   This Internet-Draft will expire on August 14, 2008.

   Abstract

   This document specifies an alternative encapsulation of the Datagram
   Congestion Control Protocol (DCCP), referred to as DCCP-NAT. This
   encapsulation will allow DCCP to be carried through the current
   generation of Network Address Translation (NAT) middleboxes without
   modification of those middleboxes.

Table of Contents

1. Introduction

   The Datagram Congestion Control Protocol (DCCP), specified in
   [RFC4340], is a transport-layer protocol that provides upper layers
   with the capability of using unreliable but congestion controlled
   flows. According to [RFC4340], DCCP packets are directly
   encapsulated in IPv4 or IPv6 packets.

   In order for the [RFC4340] encapsulation to pass through Network
   Address Translation (NAT) devices, these devices must be updated to
   recognize and properly modify DCCP. This is the long-term objective
   for DCCP, and work is underway to specify the necessary operations.

   However, in the short term it would be useful to have an
   encapsulation for DCCP that would be compatible with NAT devices
   conforming to [RFC4787]. This document specifies that encapsulation,
   which is referred to as DCCP-NAT. For convenience, the [RFC4340]
   encapsulation is referred to as DCCP-RAW.

2. Terminology

   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].

3. DCCP-NAT

   The basic approach here is to insert a UDP ([RFC768]) "shim" layer
   between the IP header and a DCCP packet with a modified generic
   header (modified to eliminate redundancies between UDP and DCCP).
   Note that this is not strictly a tunneling approach. The IP
   addresses of the communicating end systems are carried in the IP
   header (which could be modified by NAT devices) and there are no
   other IP addresses embedded.

   Devices offering or using DCCP services via DCCP-NAT encapsulation
   listen on a UDP port (default port awaiting IANA action) for incoming
   packets and pass received packets along to the DCCP protocol. DCCP
   implementations MAY allow services to be simultaneously offered over
   all combinations of DCCP-RAW and DCCP-NAT encapsulations with IPv4
   and IPv6.

The basic format of a DCCP-NAT packet is:

```
+-----------------------------------+
|      IP Header (IPv4 or IPv6)      |  Variable length
+-----------------------------------+
|            UDP Header             |  8 bytes
+-----------------------------------+
|      DCCP-NAT Generic Header      |  12 bytes
+-----------------------------------+
| Additional (type-specific) Fields |  Variable length (could be 0)
+-----------------------------------+
|            DCCP Options           |  Variable length (could be 0)
+-----------------------------------+
|         Application Data Area     |  Variable length (could be 0)
+-----------------------------------+
```

3.1 UDP Header

   The format of the UDP header is taken from [RFC768]:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Source Port           |            Dest Port          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Length              |           Checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

   For DCCP-NAT, the fields are interpreted as follows:

   Source and Dest(ination) Ports: 16 bits each
      These fields identify the UDP ports on which the source and
      destination (respectively) of the packet are listening for incoming
      DCCP-NAT packets (normally both are the default port to be assigned
      by IANA). Note that they do not identify the DCCP source and
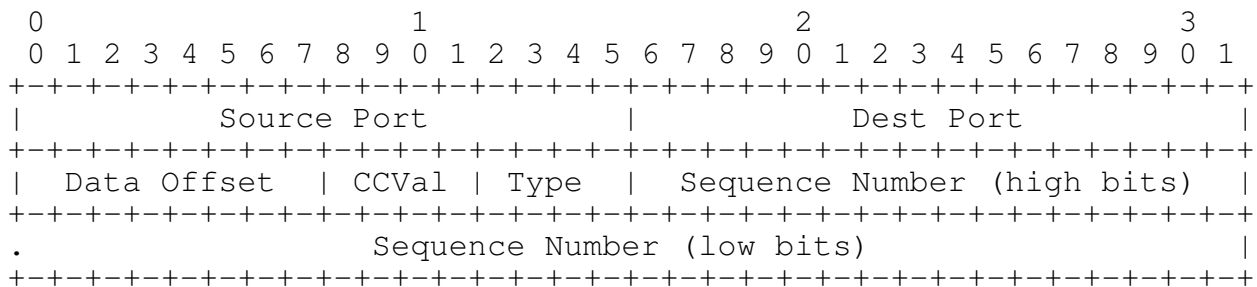   destination ports.

   Length: 16 bits
      This field is the length of the UDP datagram, including the UDP
      header and the payload (which for DCCP-NAT is the DCCP-NAT
      datagram). For DCCP-NAT, when the UDP Checksum is non-zero, Length
      MUST be at least the size of the UDP header (8 bytes) plus the
      minimum size of a DCCP-NAT header (12 bytes), for a total minimum
      value of 20 bytes. When the UDP Checksum is zero, the DCCP-NAT
      header MUST also contain a Partial Checksum Extension Header,
      therefore the minimum DCCP-NAT header is 16 bytes and the total
      minimum is 24 bytes. Received packets with a UDP Length of less
      than the applicable minimum length MUST be ignored.

Checksum: 16 bits
    This field is the Internet checksum of a network-layer pseudoheader
    and the entire UDP packet. For DCCP-NAT, a packet with a checksum
    field equal to 0 that does not contain a Partial Checksum Extension
    Header, or contains an invalid Partial Checksum Extension Header
    MUST be ignored as incorrect checksum.

3.2 DCCP-NAT Generic Header

Unlike the DCCP-RAW generic header, the DCCP-NAT generic header takes
only one form; it does not support short sequence numbers. Its
format is as follows:

```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |          Source Port          |           Dest Port           |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |  Data Offset  | CCVal | Type  |  Sequence Number (high bits)  |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 .                  Sequence Number (low bits)                   |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

All DCCP-NAT generic header fields function as specified in
[RFC4340].

3.2.1 DCCP-RAW Checksum Field

For DCCP-NAT, the function of the DCCP-RAW generic header field
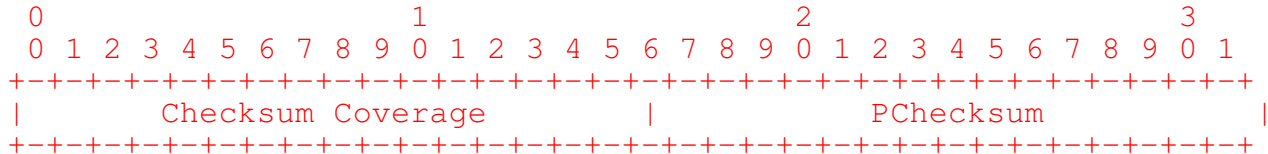Checksum is performed by the UDP Checksum field.

If the UDP Checksum field in a received packet is non-zero and is
invalid, that packet MUST be ignored as per the invalid checksum
procedures of DCCP-RAW (i.e., the options in the packet MUST NOT be
processed).

If the UDP Length field in a received packet is less than the length
of the UDP header plus the entire DCCP-NAT header (including the
generic header, Partial Checksum Extension Header if present, type-
specific fields and options), or the UDP Length field is greater than
the length of the packet from the beginning of the UDP header to the
end of the packet, that packet MUST also be ignored as per the
invalid checksum procedures.

If the UDP Checksum field is zero, then the Partial Checksum
Extension Header procedures apply. See section 3.3 for more
information.

3.3 Partial Checksum Extension Header

If the UDP Checksum field is zero, the DCCP-NAT generic header MUST
be immediately followed by a Partial Checksum Extension Header.
Additional type-specific header fields and DCCP Options would then
follow the Partial Checksum Extension Header. If the UDP Checksum
field is non-zero the Partial Checksum Extension Header MUST NOT be
included. The format of the header is as follows:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |          Checksum Coverage         |            PChecksum           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The fields are defined as follows:

Checksum Coverage: 16 bits
  This is the number of bytes of user data that are covered by the
  partial checksum. It MUST NOT be greater than the entire length of
  the user data (from the end of the DCCP-NAT header, including
  options, to the end of the packet). Packets whose Checksum
  Coverage fields are greater than the length of the user data MUST
  be ignored as incorrect checksum.

PChecksum: 16 bits
  This is the Internet checksum of the DCCP-NAT header plus Checksum
  Coverage bytes of the user data. Using the TCP/IP checksum
  algorithm, the PChecksum field is first set to zero. If the
  Checksum Coverage field is odd, the data to be summed is extended
  by one byte set to zero. This byte does not overwrite the
  corresponding byte in the DCCP-NAT packet, and is not transmitted.
  The PChecksum field is then set to the one's complement of the
  one's complement sum of the sixteen-bit words covered (DCCP header
  plus Checksum Coverage bytes of user data plus one zero byte if
  Checksum Coverage is odd). Note that PChecksum does not include an
  IP pseudoheader. Packets with invalid PChecksum fields MUST be
  ignored as incorrect checksum.

3.4 Minimum Checksum Coverage Feature

The Minimum Checksum Coverage Feature lets a DCCP endpoint determine
whether its peer is willing to accept packets with partial checksum
coverage. It takes values from 0 to 15. For DCCP-NAT the feature
values are interpreted as follows:

  o Minimum Checksum Coverage = 0, the peer will not accept packets
     with partial checksum. All UDP Checksum fields should be non-
     zero and the Partial Checksum Extension Header is never included.

o Minimum Checksum Coverage > 0, the peer will accept packets with
   partial checksum as long as the Checksum Coverage field is at
   least (Minimum Checksum Coverage - 1)*4.

As in DCCP-RAW, peers may refuse to process packets with unacceptable
Checksum Coverage. Such packets SHOULD be reported using Data
Dropped options with Drop Code 0, Protocol Constraints.

3.5 Other DCCP Headers and Options

All type-specific DCCP headers are as in DCCP-RAW, except that the
short sequence number version of the acknowledgement header is not
supported. All option and feature encodings are as in DCCP-RAW.

3.6 Service Codes and the DCCP Port Registry

There is one Service Code registry and one DCCP port registry and
they apply to all combinations of encapsulation and IP version. A
DCCP Service Code specifies an application using DCCP regardless of
the combination of DCCP encapsulation and IP version. An application
MAY choose not to support some combinations of encapsulation and IP
version, but its Service Code will remain registered for those
combinations and MUST NOT be used by other applications. An
application SHOULD NOT register different Service Codes for different
combinations of encapsulation and IP version.

Similarly, a port registration is applicable to all combinations of
encapsulation and IP version. Again, an application MAY choose not
to support some combinations of encapsulation and IP version on its
registered port, although the port will remain registered for those
combinations. Applications SHOULD NOT register different ports just
for the purpose of using different encapsulation combinations. Since
the port registry supports multiple applications registering the same
port (as long as the Service Codes are different), other applications
MAY register on the same port, but those registrations are also
applicable to all combinations of encapsulation and IP version.

4. Security Considerations

DCCP-NAT provides all of the security risk-mitigation measures
present in DCCP-RAW, and also all of the security risks, except those
associated with short sequence numbers (since DCCP-NAT does not
support that feature).

The purpose of DCCP-NAT is to allow DCCP to pass through NAT devices,
and therefore it exposes DCCP to the risks associated with passing
through NAT devices. It does not create any new risks with regard to
NAT devices.

DCCP-NAT may also allow DCCP applications to pass through existing
firewall devices, if the administrators of the devices so choose.

The option is a binary one however; either allow all DCCP
applications or allow none. Proper control of DCCP application-by-
application will require enhancements to firewalls.

5. IANA Considerations

A port allocation request has been placed with IANA for the dccp-nat
service port in UDP. No other IANA actions are necessary.

6. References

6.1 Normative References

   [RFC4340]   Kohler, E., Handley, M., Floyd, S., "Datagram Congestion
                Control Protocol (DCCP)", RFC 4340, March 2006.

   [RFC768]   Postel, J., "User Datagram Protocol", RFC 768, August
                1980.

   [RFC4787]   Audet, F., Jennings, C., "Network Address Translation
                (NAT) Behavioral Requirements for Unicast UDP", RFC 4787,
                January 2007.

   [RFC2119]   Bradner, S., "Key words for use in RFCs to Indicate
                Requirement Levels", RFC 2119, March 1997.

7. Author's Address

   Tom Phelan
   Sonus Networks
   7 Technology Park Dr.
   Westford, MA USA 01886
   Phone: 978-614-8456
   Email: tphelan@sonusnet.com

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any
Intellectual Property Rights or other rights that might be claimed to
pertain to the implementation or use of the technology described in
this document or the extent to which any license under such rights
might or might not be available; nor does it represent that it has
made any independent effort to identify any such rights. Information
on the procedures with respect to rights in RFC documents can be
found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any
assurances of licenses to be made available, or the result of an
attempt made to obtain a general license or permission for the use of
such proprietary rights by implementers or users of this
specification can be obtained from the IETF on-line IPR repository at
http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any
copyrights, patents or patent applications, or other proprietary
rights that may cover technology that may be required to implement
this standard. Please address the information to the IETF at
ietf-ipr@ietf.org.

Disclaimer of Validity

This document and the information contained herein are provided on an
"AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS
OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND
THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS
OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF
THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED
WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Copyright (C) The IETF Trust (2008).

This document is subject to the rights, licenses and restrictions
contained in BCP 78, and except as set forth therein, the authors
retain all their rights.

Acknowledgment

Funding for the RFC Editor function is currently provided by the
Internet Society.