RTP and the Datagram Congestion Control Protocol (DCCP)
draft-perkins-dccp-rtp-01.txt

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 September 7, 2006.

Copyright Notice

Copyright (C) The Internet Society (2006).

Abstract

The Real-time Transport Protocol (RTP) is a widely used transport for real-time media on IP networks. The Datagram Congestion Control Protocol (DCCP) is a newly defined transport protocol that provides desirable services for real-time applications. This memo specifies a mapping of RTP onto DCCP, along with associated signalling, such that real-time applications can make use of the services provided by DCCP.
Table of Contents

1. Introduction .............................................. 3
2. Rationale .................................................. 3
3. Conventions Used in this Memo ............................. 4
4. RTP over DCCP: Framing .................................... 4
   4.1. RTP Data Packets .................................. 4
   4.2. RTP Control Packets ................................. 5
   4.3. Multiplexing Data and Control ....................... 6
   4.4. RTP Sessions and DCCP Connections .................. 7
   4.5. RTP Profiles .......................................... 7
5. RTP over DCCP: Signalling using SDP ....................... 8
   5.1. Protocol Identification ............................. 8
   5.2. Service Codes ...................................... 9
   5.3. Connection Management ............................ 10
   5.4. Example ........................................... 10
6. Security Considerations .................................. 11
7. IANA Considerations ....................................... 11
8. Acknowledgements ......................................... 12
9. References ................................................. 12
   9.1. Normative References .............................. 12
   9.2. Informative References ............................ 13
Author’s Address .............................................. 14
Intellectual Property and Copyright Statements ............. 15
1. Introduction

The Real-time Transport Protocol (RTP) [1] is widely used in video streaming, telephony, and other real-time networked applications. RTP can run over a range of lower-layer transport protocols, and the performance of an application using RTP is heavily influenced by the choice of lower-layer transport. The Datagram Congestion Control Protocol (DCCP) [2] is a newly specified transport protocol that provides desirable properties for real-time applications running on unmanaged best-effort IP networks. This memo describes how RTP can be framed for transport using DCCP, and discusses some of the implications of such a framing. It also describes how the Session Description Protocol (SDP) [3] can be used to signal such sessions.

The remainder of this memo is structured as follows: we begin with a rationale for the work in Section 2, describing why a mapping of RTP onto DCCP is needed. Following a description of the conventions used in this memo in Section 3, the specification begins in Section 4 with the definition of how RTP packets are framed within DCCP; associated signalling is described in Section 5. We conclude with a discussion of security considerations in Section 6, and IANA considerations in Section 7.

2. Rationale

With the widespread adoption of RTP have come concerns that many real time applications do not implement congestion control, leading to the potential for congestion collapse of the network [14]. The designers of RTP recognised this issue, stating that [4]:

If best-effort service is being used, RTP receivers SHOULD monitor packet loss to ensure that the packet loss rate is within acceptable parameters. Packet loss is considered acceptable if a TCP flow across the same network path and experiencing the same network conditions would achieve an average throughput, measured on a reasonable timescale, that is not less than the RTP flow is achieving. This condition can be satisfied by implementing congestion control mechanisms to adapt the transmission rate (or the number of layers subscribed for a layered multicast session), or by arranging for a receiver to leave the session if the loss rate is unacceptably high.

While the goals are clear, the development of TCP friendly congestion control that can be used with RTP and real-time media applications is an open research question with many proposals for new algorithms, but little deployment experience.
Two approaches have been used to provide congestion control for RTP:
1) develop new RTP profiles that incorporate congestion control; and
2) provide mechanisms for running RTP over congestion controlled
transport protocols. The RTP Profile for TCP Friendly Rate Control
[15] is an example of the first approach, extending the RTP packet
formats to incorporate feedback information such that TFRC congestion
control [16] can be implemented at the application level. This
approach has the advantage that congestion control can be added to
existing applications, without needing operating system or network
support, and offers flexibility to experiment with new congestion
control algorithms as they are developed. Unfortunately, there is
also the consequent disadvantage that the complexity of implementing
congestion control is passed onto the application author, a burden
which many would prefer to avoid.

The other approach is to run RTP on a lower-layer transport protocol
that provides congestion control. One possibility is to run RTP over
TCP, as defined in [5], but the reliable nature of TCP and the
dynamics of its congestion control algorithm make this inappropriate
for most interactive real time applications (SCTP is inappropriate
for similar reasons). A better fit for such applications may be to
run RTP over DCCP, since DCCP offers unreliable packet delivery and a
choice of congestion control. This gives applications the ability to
tailor the transport to their needs, taking advantage of better
congestion control algorithms as they come available, while passing
complexity of implementation to the operating system. If DCCP should
come to be widely available, it is believed these will be compelling
advantages. Accordingly, this memo defines a mapping of RTP onto
DCCP.

3. Conventions Used in this Memo

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 RFC 2119 [6].

4. RTP over DCCP: Framing

The following section defines how RTP and RTCP packets can be framed
for transport using DCCP. It also describes the differences between
RTP sessions and DCCP connections, and the impact these have on the
design of applications.

4.1. RTP Data Packets

Each RTP data packet MUST be conveyed in a single DCCP datagram.
Fields in the RTP header MUST be interpreted according to the RTP specification, and any applicable RTP Profile and Payload Format. Header processing is not affected by DCCP framing (in particular, note that the semantics of the RTP sequence number and the DCCP sequence number are not compatible, and the value of one cannot be inferred from the other).

A DCCP connection is opened when an end system joins an RTP session, and it remains open for the duration of the session. To ensure NAT bindings are kept open, an end system SHOULD send periodic low rate zero length DCCP-Data packets during periods when it has no RTP data to send. This removes the need for RTP no-op packets [17] when using RTP over DCCP.

RTP data packets MUST obey the dictates of DCCP congestion control. In some cases, the congestion control will require a sender to send at a rate below that which the payload format would otherwise use. To support this, an application should use either a rate adaptive payload format, or a range of payload formats (allowing it to switch to a lower rate format if necessary). Details of the rate adaptation policy for particular payload formats are outside the scope of this memo.

TODO: provide more guidance on implementation of congestion control within an RTP application.

DCCP allows an application to choose the checksum coverage, using a partial checksum to allow an application to receive packets with corrupt payloads. Some RTP Payload Formats (e.g. [18]) can make use of this feature in conjunction with payload-specific mechanisms to improve performance when operating in environments with frequent non-congestive packet corruption. If such a payload format is used, an RTP end system MAY enable partial checksums at the DCCP layer, in which case the checksum MUST cover at least the DCCP and RTP headers to ensure packets are correctly delivered. Partial checksums MUST NOT be used unless supported by mechanisms in the RTP payload format.

4.2. RTP Control Packets

The RTP Control Protocol (RTCP) is be used in the usual manner with DCCP. RTCP packets MUST be sent grouped into compound packets, as described in Section 6.1 of [1]. Each compound RTCP packet MUST be transported in a single DCCP datagram.

The usual RTCP timing rules apply, subject to the constraint that RTCP packets MUST be subject to DCCP congestion control.

TODO: RTCP relies on a configured nominal "session bandwidth" in the
calculation of the reporting interval. It is not clear that this is appropriate for congestion controlled sessions, since the actual RTP rate may vary significantly over time, and may differ for each half of a DCCP connection. There are two options: 1) use the nominal RTP session bandwidth to drive RTCP, accepting that this may over- or under-estimate the RTCP reporting interval; or 2) modify the RTCP reporting interval based on the session bandwidth as determined by the congestion control algorithm. The second choice introduces some considerable complexity, since there is no longer a simple definition for the reporting interval on which all participants will agree, but it will better control the RTCP sending rate.

As noted in Section 17.1 of [2], there is the potential for overlap between the information conveyed in RTCP packets and that conveyed in DCCP acknowledgement options. In general this is not an issue: RTCP packets contain media-specific data that is not present in DCCP acknowledgement options, and DCCP options contain network-level data that is not present in RTCP. Indeed, there is no overlap between the five RTCP packet types defined in the RTP specification [1] and the standard DCCP options defined in [2]. There are, however, other cases where overlap does occur: most clearly between the optional RTCP Extended Reports Loss RLE Blocks [19] and the DCCP Ack Vector option. If there is overlap between RTCP report packets and DCCP acknowledgements, an application should use either RTCP feedback or DCCP acknowledgements, but not both (use of both types of feedback will waste available network capacity, but is not otherwise harmful).

4.3. Multiplexing Data and Control

The obvious mapping of RTP onto DCCP creates two DCCP connections for each RTP flow: one for RTP data packets, one for RTP control packets. A frequent criticism of RTP relates to the number of ports it uses, since large telephony gateways can support more than 32768 RTP flows between pairs of gateways, and so run out of UDP ports. In addition, use of multiple ports complicates NAT traversal. For these reasons, it is RECOMMENDED that RTP and RTCP flows be multiplexed onto a single DCCP connection.

RTP and RTCP packets multiplexed onto a single connection can be distinguished provided care is taken in assigning RTP payload types. The RTP payload type and marker bit(s) occupy the same space in the packet as does the RTCP packet type field. Provided the RTP payload type is chosen such that the payload type, or the payload type plus 128 (when the marker bit is set), does not clash with any of the used RTCP packet types, the two can be demultiplexed. With the RTCP packet types registered at the time of this writing, this implies that RTP payload types 64-65 and 72-79 must be avoided. None of the registered static payload type assignments are in this range, and
typical practice is to make dynamic assignments in the range 96-127, so this restriction is not typically problematic. This multiplexing does not otherwise impact the operation of RTP or RTCP.

There may be circumstances where multiplexing RTP and RTCP is not desired, for example when translating from an RTP stream over non-DCCP transport that uses conflicting RTP payload types and RTCP packet types. As specified in Section 5.1, the "a=rtcp:" SDP attribute MAY be used to signal use of non-multiplexed RTCP.

4.4. RTP Sessions and DCCP Connections

An end system should not assume that it will observe only a single RTP synchronisation source (SSRC) because it is using DCCP framing. An RTP session can span any number of transport connections, and can include RTP mixers or translators bringing other participants into the session. The use of a unicast DCCP connection does not imply that the RTP session will have only two participants, and RTP end systems must assume that multiple synchronisation sources may be observed when using RTP over DCCP.

An RTP translator bridging multiple DCCP connections to form a single RTP session needs to be aware of the congestion state of each DCCP connection, and must adapt the media to the available capacity of each. In general, transcoding is required to perform adaptation: this is computationally expensive, induces delay, and generally gives poor quality results. Depending on the payload, it might be possible to use some form of scalable coding. Scalable media coding formats are an active research area, and are not in widespread use at the time of this writing.

A single RTP session may also span a DCCP connection and some other type of transport connection. An example might be an RTP over DCCP connection from an RTP end system to an RTP translator, with an RTP over UDP/IP multicast group on the other side of the translator. A second example might be an RTP over DCCP connection that links PSTN gateways. The issues for such an RTP translator are similar to those when linking two DCCP connections, except that the congestion control algorithms on either side of the translator may not be compatible. Implementation of effective translators for such an environment is nontrivial.

4.5. RTP Profiles

In general, there is no conflict between new RTP Profiles and DCCP framing, and most RTP profiles can be negotiated for use over DCCP. The only potential for conflict occurs if an RTP profile changes the RTCP reporting interval or the RTP packet transmission rules, since
this may conflict with DCCP congestion control. If an RTP profile conflicts with DCCP congestion control, that profile MUST NOT be used with DCCP.

Of the profiles currently defined, the RTP Profile for Audio and Video Conferences with Minimal Control [4], the Secure Real-time Transport Protocol [7] the Extended RTP Profile for RTCP-based Feedback [8], and the Extended Secure RTP Profile for RTCP-based Feedback [9] MAY be used with DCCP. The RTP Profile for TFRC [15] MUST NOT be used with DCCP, since it conflicts with DCCP congestion control by providing alternative congestion control semantics.

5. RTP over DCCP: Signalling using SDP

The Session Description Protocol (SDP) [3] and the offer/answer model [10] are widely used to negotiate RTP sessions (for example, using the Session Initiation Protocol [20]). This section describes how SDP is used to signal RTP sessions running over DCCP.

5.1. Protocol Identification

SDP uses a media ("m=") line to convey details of the media format and transport protocol used. The ABNF syntax of a media line is as follows (from [3]):

```
media-field = "m=" media SP port ["/" integer] SP proto 1*(SP fmt) CRLF
```

The proto field denotes the transport protocol used for the media, while the port indicates the transport port to which the media is sent. Following [5] and [11] this memo defines the following five values of the proto field to indicate media transported using DCCP:

- DCCP
- DCCP/RTP/AVP
- DCCP/RTP/SAVP
- DCCP/RTP/AVPF
- DCCP/RTP/SAVPF

The "DCCP" protocol identifier is similar to the "UDP" and "TCP" protocol identifiers and describes the transport protocol, but not the upper-layer protocol. An SDP "m=" line that specifies the "DCCP" protocol MUST further qualify the application layer protocol using a fmt identifier. A single DCCP port is used, as denoted by the port field in the media line. The "DCCP" protocol identifier MUST NOT be used to signal RTP sessions running over DCCP.
The "DCCP/RTP/AVP" protocol identifier refers to RTP using the RTP Profile for Audio and Video Conferences with Minimal Control [4] running over DCCP.

The "DCCP/RTP/SAVP" protocol identifier refers to RTP using the Secure Real-time Transport Protocol [7] running over DCCP.

The "DCCP/RTP/AVPF" protocol identifier refers to RTP using the Extended RTP Profile for RTCP-based Feedback [8] running over DCCP.

The "DCCP/RTP/SAVPF" protocol identifier refers to RTP using the Extended Secure RTP Profile for RTCP-based Feedback [9] running over DCCP.

By default, a single DCCP connection on the specified port is used for both RTP and RTCP packets. The "a=rtcp:" attribute [12] MAY be used to specify an alternate DCCP port for RTCP, in which case a separate DCCP connection is opened to transport the RTCP data.

5.2. Service Codes

In addition to the port number, specified on the SDP "m=" line, a DCCP connection has an associated service code. A single new SDP attribute is defined to signal the service code:

dccp-service-attr = "a=dccp-service-code:" service-code

service-code = hex-sc / decimal-sc / ascii-sc

hex-sc = "SC=x" *HEXDIG

decimal-sc = "SC=" *DIGIT

ascii-sc = "SC:" *sc-char

sc-char = %d42-43, %d45-47, %d63-90, %d95, %d97-122

where DIGIT and HEXDIG are as defined in [13]. The service code should be interpreted as defined in Section 8.1.2 of [2]. The following DCCP service codes are registered for use with RTP:

- SC:RTPA an RTP session conveying audio data
- SC:RTPV an RTP session conveying video data
- SC:RTPT an RTP session conveying textual data
To ease the job of middleboxes, applications SHOULD use these service codes to identify RTP sessions running within DCCP.

The "a=dccp-service-code:" attribute is a media level attribute which is not subject to the charset attribute.

5.3. Connection Management

The "a=setup:" attribute indicates which of the end points should initiate the DCCP connection establishment (i.e. send the initial DCCP-Request packet). The "a=setup:" attribute MUST be used in a manner comparable with [11], except that DCCP connections are being initiated rather than TCP connections.

After the initial offer/answer exchange, the end points may decide to re-negotiate various parameters. The "a=connection:" attribute MUST be used in a manner compatible with [11] to decide whether a new DCCP connection needs to be established as a result of subsequent offer/answer exchanges, or if the existing connection should still be used.

5.4. Example

An offerer at 192.0.2.47 signals its availability for an H.261 video session, using RTP/AVP over DCCP with service code "RTPV":

```plaintext
v=0
o=alice 1129377363 1 IN IP4 192.0.2.47
s=-
c=IN IP4 192.0.2.47
t=0 0
m=video 51370 DCCP/RTP/AVP 99
a=rtpmap:99 h261/90000
a=dccp-service-code:SC=x52545056
a=setup:passive
a=connection:new
```

An answerer at 192.0.2.128 receives this offer and responds with the following answer:
v=0
o=bob 1129377364 1 IN IP4 192.0.2.128
s=-
c=IN IP4 192.0.2.128
t=0 0
m=video 9 DCCP/RTP/AVP 99
a=rtpmap:99 h261/90000
a=dccp-service-code:SC:RTPV
a=setup:active
a=connection:new

The end point at 192.0.2.128 then initiates a DCCP connection to port 51370 at 192.0.2.47. Note that DCCP port 51370 is used for both the RTP and RTCP data, and port 51371 is unused.

6. Security Considerations

The security considerations in the RTP specification [1] and any applicable RTP profile (e.g. [4], [7], [8], or [9]) or payload format apply when transporting RTP over DCCP.


The SDP signalling described in Section 5 is subject to the security considerations of [3], [10], [11] and [5].

It is not believed that any additional security considerations are introduced as a result of combining these protocols. Indeed, the provision of effective congestion control for RTP will alleviate the potential for denial-of-service present when RTP flows ignore the advice in [1] to monitor packet loss and reduce their sending rate in the face of persistent congestion.

7. IANA Considerations

The following SDP "proto" field identifiers are registered: "DCCP", "DCCP/RTP/AVP", "DCCP/RTP/SAVP", "DCCP/RTP/AVPF" and "DCCP/RTP/SAVPF" (see Section 5.1 of this memo).

One new SDP attribute ("a=dccp-service-code:" ) is registered (see Section 5.2 of this memo).

The following DCCP service codes are registered: SC:RTPA, SC:RTPV, SC:RTPT, and SC:RTPO (see Section 5.2 of this memo).
8. Acknowledgements

This work was supported in part by the UK Engineering and Physical Sciences Research Council. Thanks are due to Philippe Gentic, Magnus Westerlund and the other members of the DCCP working group for their comments.

9. References

9.1. Normative References


9.2. Informative References


Author’s Address

Colin Perkins
University of Glasgow
Department of Computing Science
17 Lilybank Gardens
Glasgow  G12 8QQ
UK

Email: csp@csperkins.org
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 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 Internet Society (2006). 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.