RTP and the Datagram Congestion Control Protocol (DCCP)

Abstract

   The Real-time Transport Protocol (RTP) is a widely used transport for
   real-time multimedia on IP networks. The Datagram Congestion Control
   Protocol (DCCP) is a 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.

Status of This Memo

   This is an Internet Standards Track document.

   This document is a product of the Internet Engineering Task Force
   (IETF). It represents the consensus of the IETF community. It has
   received public review and has been approved for publication by the
   Internet Engineering Steering Group (IESG). Further information on
   Internet Standards is available in Section 2 of RFC 5741.

   Information about the current status of this document, any errata,
   and how to provide feedback on it may be obtained at
   http://www.rfc-editor.org/info/rfc5762.

Table of Contents

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 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: it begins 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. Security considerations are
   discussed 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 [15]. The
   designers of RTP recognised this issue, stating in RFC 3551 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 RTP extensions that incorporate congestion control; and 2)
   provide mechanisms for running RTP over congestion-controlled
   transport protocols. An example of the first approach can be found
   in [16], extending RTP to incorporate feedback information such that
   TCP Friendly Rate Control (TFRC) [17] can be implemented at the
   application level. This will allow congestion control to be added to
   existing applications without operating system or network support,
   and it offers the flexibility to experiment with new congestion
   control algorithms as they are developed. Unfortunately, it also
   passes the complexity of implementing congestion control onto
   application authors, a burden which many would prefer to avoid.

   The second 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 (the Stream Control
   Transmission Protocol (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 the 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 RTP Control Protocol (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 a zero-length DCCP-
Data packet once every 15 seconds during periods when it has no other
data to send. This removes the need for RTP no-op packets [18], and
similar application-level keepalives, when using RTP over DCCP. This
application-level keepalive does not need to be sent if it is known
that the DCCP CCID in use provides a transport-level keepalive, or if
the application can determine that there are no NAT devices on the
path.

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 could 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 (but see [19] and [20] for guidance).

RTP extensions that provide application-level congestion control
(e.g., [16]) will conflict with DCCP congestion control, and MUST NOT
be used.

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., [21]) 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 used in the standard manner with
DCCP. RTCP packets are grouped into compound packets, as described
in Section 6.1 of [1], and each compound RTCP packet is transported
in a single DCCP datagram.

The usual RTCP timing rules apply, with the additional constraint
that RTCP packets MUST obey the DCCP congestion control algorithm
negotiated for the connection. This can prevent a participant from
sending an RTCP packet at the expiration of the RTCP transmission
timer if there is insufficient network capacity available. In such
cases the RTCP packet is delayed and sent at the earliest possible
instant when capacity becomes available. The actual time the RTCP
packet was sent is then used as the basis for calculating the next
RTCP transmission time.

RTCP packets comprise only a small fraction of the total traffic in
an RTP session. Accordingly, it is expected that delays in their
transmission due to congestion control will not be common, provided
the configured nominal "session bandwidth" (see Section 6.2 of [1])
is in line with the bandwidth achievable on the DCCP connection. If,
however, the capacity of the DCCP connection is significantly below
the nominal session bandwidth, RTCP packets may be delayed enough for
participants to time out due to apparent inactivity. In such cases,
the session parameters SHOULD be re-negotiated to more closely match
the available capacity, for example by performing a re-invite with an
updated "b=" line when using the Session Initiation Protocol [22] for
signalling.

   Note: Since the nominal session bandwidth is chosen based on media
   codec capabilities, a session where the nominal bandwidth is much
   larger than the available bandwidth will likely become unusable
   due to constraints on the media channel, and so require
   negotiation of a lower bandwidth codec, before it becomes unusable
   due to constraints on the RTCP channel.

As noted in Section 17.1 of [2], there is the potential for overlap
between information conveyed in RTCP packets and that conveyed in
DCCP acknowledgement options. In general this is not an issue since
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 [2]. There are, however, cases where overlap
does occur: most clearly between the Loss RLE Report Blocks defined
as part of the RTCP Extended Reports [23] 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 and 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 the RTP and RTCP traffic for a single
   RTP session is multiplexed onto a single DCCP connection following
   the guidelines in [7], where possible (it may not be possible in all
   circumstances, for example when translating from an RTP stream over a
   non-DCCP transport that uses conflicting RTP payload types and RTCP
   packet types).

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 SHOULD assume that multiple synchronisation sources may be
   observed when using RTP over DCCP, unless otherwise signalled.

   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. The Codec Control Messages defined in [24] may be used to
   signal congestion state to the media senders, allowing them to adapt
   their transmission. Alternatively, media transcoding may be used to
   perform adaptation: this is computationally expensive, induces delay,
   and generally gives poor-quality results. Depending on the payload,
   it might also be possible to use some form of scalable coding.

   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 Public
   Switched Telephone Network (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 non-trivial.

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
   with the following exceptions:

   o An RTP profile that is intolerant of packet corruption may
      conflict with the DCCP partial checksum feature. An example of
      this is the integrity protection provided by the RTP/SAVP profile,
      which cannot be used in conjunction with DCCP partial checksums.

   o An RTP profile that mandates a particular non-DCCP lower-layer
      transport will conflict with DCCP.

   RTP profiles that fall under these exceptions SHOULD NOT be used with
   DCCP unless the conflicting features can be disabled.

   Of the profiles currently defined, the RTP Profile for Audio and
   Video Conferences with Minimal Control [4], the Secure Real-time
   Transport Protocol [8], the Extended RTP Profile for RTCP-based
   Feedback [9], and the Extended Secure RTP Profile for RTCP-based
   Feedback [10] MAY be used with DCCP (noting the potential conflict
   between DCCP partial checksums and the integrity protection provided
   by the secure RTP variants -- see Section 6).

5. RTP over DCCP: Signalling using SDP

   The Session Description Protocol (SDP) [3] and the offer/answer model
   [11] are widely used to negotiate RTP sessions (for example, using
   the Session Initiation Protocol [22]). 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 = %x6d "=" 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 [12], this memo defines these 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 denotes the DCCP transport protocol [2], but
   not its upper-layer protocol. An SDP "m=" line that specifies the
   "DCCP" protocol MUST further qualify the application-layer protocol
   using a "fmt" identifier (the "fmt" namespace is managed in the same
   manner as for the "UDP" protocol 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; those sessions MUST use a protocol identifier of the form
   "DCCP/RTP/..." as described below.

   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 [8] running over DCCP.

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

   The "DCCP/RTP/SAVPF" protocol identifier refers to RTP using the
   Extended Secure RTP Profile for RTCP-based Feedback [10] running over
   DCCP.

   RTP payload formats used with the "DCCP/RTP/AVP", "DCCP/RTP/SAVP",
   "DCCP/RTP/AVPF", and "DCCP/RTP/SAVPF" protocol identifiers MUST use
   the payload type number as their "fmt" value. If the payload type
   number is dynamically assigned, an additional "rtpmap" attribute MUST
   be included to specify the format name and parameters as defined by
   the media type registration for the payload format.

   DCCP port 5004 is registered for use by the RTP profiles listed
   above, and SHOULD be the default port chosen by applications using
   those profiles. If multiple RTP sessions are active from a host,
   even-numbered ports in the dynamic range SHOULD be used for the other
   sessions. If RTP is to be sent on a separate DCCP connection to
   RTP, the RTCP connection SHOULD use the next higher destination port
   number, unless an alternative DCCP port is signalled using the
   "a=rtcp:" attribute [13]. For improved interoperability, "a=rtcp:"
   SHOULD be used whenever an alternate DCCP port is used.

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 ("dccp-service-code") is defined to signal the DCCP service
   code according to the following ABNF [14]:

       dccp-service-attr = %x61 "=dccp-service-code:" service-code

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

       hex-sc   = %x53 %x43 "=" %x78 *HEXDIG

       decimal-sc   = %x53 %x43 "=" *DIGIT

       ascii-sc   = %x53 %x43 ":" *sc-char

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

   where DIGIT and HEXDIG are as defined in [14]. The service code is
   interpreted as defined in Section 8.1.2 of [2] and may be specified
   using either the hexadecimal, decimal, or ASCII formats. A parser
   MUST interpret service codes according to their numeric value,
   independent of the format used to represent them in SDP.

   The following DCCP service codes are registered for use with RTP:

   o SC:RTPA (equivalently SC=1381257281 or SC=x52545041): an RTP
      session conveying audio data (and OPTIONAL multiplexed RTCP)

   o SC:RTPV (equivalently SC=1381257302 or SC=x52545056): an RTP
      session conveying video data (and OPTIONAL multiplexed RTCP)

   o SC:RTPT (equivalently SC=1381257300 or SC=x52545054): an RTP
      session conveying text media (and OPTIONAL multiplexed RTCP)

   o SC:RTPO (equivalently SC=1381257295 or SC=x5254504f): an RTP
      session conveying any other type of media (and OPTIONAL
      multiplexed RTCP)

   o SC:RTCP (equivalently SC=1381253968 or SC=x52544350): an RTCP
      connection, separate from the corresponding RTP

   To ease the job of middleboxes, applications SHOULD use these service
   codes to identify RTP sessions running within DCCP. The service code
   SHOULD match the top-level media type signalled for the session

(i.e., the SDP "m=" line), with the exception connections using media
types other than audio, video, or text, which use SC:RTPO, and
connections that transport only RTCP packets, which use SC:RTCP.

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

5.3. Connection Management

The "a=setup:" attribute indicates which of the endpoints 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 [12], except that DCCP connections are being
initiated rather than TCP connections.

After the initial offer/answer exchange, the endpoints may decide to
re-negotiate various parameters. The "a=connection:" attribute MUST
be used in a manner compatible with [12] 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. Multiplexing Data and Control

A single DCCP connection can be used to transport multiplexed RTP and
RTCP packets. Such multiplexing MUST be signalled using an "a=rtcp-
mux" attribute according to [7]. If multiplexed RTP and RTCP are not
to be used, then the "a=rtcp-mux" attribute MUST NOT be present in
the SDP offer, and a separate DCCP connection MUST be opened to
transport the RTCP data on a different DCCP port.

5.5. 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" (using the
hexadecimal encoding of the service code in the SDP). RTP and RTCP
packets are multiplexed onto a single DCCP connection:

```
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 5004 DCCP/RTP/AVP 99
a=rtcp-mux
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=rtcp-mux
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
5004 at 192.0.2.47. DCCP port 5004 is used for both the RTP and RTCP
data, and port 5005 is unused. The textual encoding of the service
code is used in the answer, and represents the same service code as
in the offer.

6. Security Considerations

The security considerations in the RTP specification [1] and any
applicable RTP profile (e.g., [4], [8], [9], or [10]) or payload
format apply when transporting RTP over DCCP.

The security considerations in the DCCP specification [2] apply.

The SDP signalling described in Section 5 is subject to the security
considerations of [3], [11], [12], [5], and [7].

The provision of effective congestion control for RTP through use of
DCCP is expected to help reduce 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.

There is a potential conflict between the Secure RTP profiles ([8],
[10]) and the DCCP partial checksum option, since these profiles
introduce, and recommend the use of, message authentication for RTP
and RTCP packets. Message authentication codes of the type used by
these profiles cannot be used with partial checksums, since any bit
error in the DCCP packet payload will cause the authentication check
to fail. Accordingly, DCCP partial checksums SHOULD NOT be used in
conjunction with Secure Real-time Transport Protocol (SRTP)
authentication. The confidentiality features of the basic RTP
specification cannot be used with DCCP partial checksums, since bit

errors propagate. Also, despite the fact that bit errors do not
propagate when using AES in counter mode, the Secure RTP profiles
SHOULD NOT be used with DCCP partial checksums, since the profiles
require authentication for security, and authentication is
incompatible with partial checksums.

7. IANA Considerations

The following SDP "proto" field identifiers have been registered (see
Section 5.1):

Type	SDP Name	Reference
proto	DCCP	[RFC5762]
	DCCP/RTP/AVP	[RFC5762]
	DCCP/RTP/SAVP	[RFC5762]
	DCCP/RTP/AVPF	[RFC5762]
	DCCP/RTP/SAVPF	[RFC5762]

The following new SDP attribute ("att-field") has been registered:

   Contact name: Colin Perkins <csp@csperkins.org>

   Attribute name: dccp-service-code

   Long-form attribute name in English: DCCP service code

   Type of attribute: Media level.

   Subject to the charset attribute? No.

   Purpose of the attribute: see RFC 5762, Section 5.2

   Allowed attribute values: see RFC 5762, Section 5.2

The following DCCP service code values have been registered (see
Section 5.2):

1381257281	RTPA	RTP session conveying audio data (and associated RTCP)	[RFC5762]
1381257302	RTPV	RTP session conveying video data (and associated RTCP)	[RFC5762]
1381257300	RTPT	RTP session conveying text media (and associated RTCP)	[RFC5762]
1381257295	RTPO	RTP session conveying other media (and associated RTCP)	[RFC5762]
1381253968	RTCP	RTCP connection, separate from the corresponding RTP	[RFC5762]

The following DCCP ports have been registered (see Section 5.1):

```
avt-profile-1 5004/dccp  RTP media data       [RFC3551, RFC5762]
avt-profile-2 5005/dccp  RTP control protocol [RFC3551, RFC5762]
```

Note: ports 5004/tcp, 5004/udp, 5005/tcp, and 5005/udp have existing
registrations, but incorrect descriptions and references. The IANA
has updated the existing registrations as follows:

```
avt-profile-1 5004/tcp   RTP media data       [RFC3551, RFC4571]
avt-profile-1 5004/udp   RTP media data       [RFC3551]
avt-profile-2 5005/tcp   RTP control protocol [RFC3551, RFC4571]
avt-profile-2 5005/udp   RTP control protocol [RFC3551]
```

8. Acknowledgements

9. References

9.1. Normative References

[1]   Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson,
       "RTP: A Transport Protocol for Real-Time Applications", STD 64,
       RFC 3550, July 2003.

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

[3]   Handley, M., Jacobson, V., and C. Perkins, "SDP: Session
       Description Protocol", RFC 4566, July 2006.

[4]   Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video
       Conferences with Minimal Control", STD 65, RFC 3551, July 2003.

[5]   Lazzaro, J., "Framing Real-time Transport Protocol (RTP) and
       RTP Control Protocol (RTCP) Packets over Connection-Oriented
       Transport", RFC 4571, July 2006.

[6]   Bradner, S., "Key words for use in RFCs to Indicate Requirement
       Levels", BCP 14, RFC 2119, March 1997.

   [7]   Perkins, C. and M. Westerlund, "Multiplexing RTP Data and
          Control Packets on a Single Port", RFC 5761, April 2010.

   [8]   Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K.
          Norrman, "The Secure Real-time Transport Protocol (SRTP)",
          RFC 3711, March 2004.

   [9]   Ott, J., Wenger, S., Sato, N., Burmeister, C., and J. Rey,
          "Extended RTP Profile for Real-time Transport Control Protocol
          (RTCP)-Based Feedback (RTP/AVPF)", RFC 4585, July 2006.

   [10]   Ott, J. and E. Carrara, "Extended Secure RTP Profile for Real-
          time Transport Control Protocol (RTCP)-Based Feedback (RTP/
          SAVPF)", RFC 5124, February 2008.

   [11]   Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with
          Session Description Protocol (SDP)", RFC 3264, June 2002.

   [12]   Yon, D. and G. Camarillo, "TCP-Based Media Transport in the
          Session Description Protocol (SDP)", RFC 4145, September 2005.

   [13]   Huitema, C., "Real Time Control Protocol (RTCP) attribute in
          Session Description Protocol (SDP)", RFC 3605, October 2003.

   [14]   Crocker, D. and P. Overell, "Augmented BNF for Syntax
          Specifications: ABNF", STD 68, RFC 5234, January 2008.

9.2.   Informative References

   [15]   Floyd, S. and J. Kempf, "IAB Concerns Regarding Congestion
          Control for Voice Traffic in the Internet", RFC 3714,
          March 2004.

   [16]   Gharai, L., "RTP with TCP Friendly Rate Control", Work
          in Progress, July 2007.

   [17]   Floyd, S., Handley, M., Padhye, J., and J. Widmer, "TCP
          Friendly Rate Control (TFRC): Protocol Specification",
          RFC 5348, September 2008.

   [18]   Andreasen, F., Oran, D., and D. Wing, "A No-Op Payload Format
          for RTP", Work in Progress, May 2005.

   [19]   Phelan, T., "Strategies for Streaming Media Applications Using
          TCP-Friendly Rate Control", Work in Progress, July 2007.

   [20]   Phelan, T., "Datagram Congestion Control Protocol (DCCP) User
          Guide", Work in Progress, April 2005.

   [21] Sjoberg, J., Westerlund, M., Lakaniemi, A., and Q. Xie, "RTP
         Payload Format and File Storage Format for the Adaptive Multi-
         Rate (AMR) and Adaptive Multi-Rate Wideband (AMR-WB) Audio
         Codecs", RFC 4867, April 2007.

   [22] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A.,
         Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP:
         Session Initiation Protocol", RFC 3261, June 2002.

   [23] Friedman, T., Caceres, R., and A. Clark, "RTP Control Protocol
         Extended Reports (RTCP XR)", RFC 3611, November 2003.

   [24] Wenger, S., Chandra, U., Westerlund, M., and B. Burman, "Codec
         Control Messages in the RTP Audio-Visual Profile with Feedback
         (AVPF)", Work in Progress, October 2007.

Author's Address

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

   EMail: csp@csperkins.org