AVTCORE   J. Lennox
Internet-Draft   Vidyo
Updates: 3550 (if approved)   M. Westerlund
Intended status: Standards Track   Ericsson
Expires: January 12, 2014   Q. Wu
                                                              Huawei
                                                          C. Perkins
                                                University of Glasgow
                                                        July 11, 2013

         Sending Multiple Media Streams in a Single RTP Session
                draft-ietf-avtcore-rtp-multi-stream-01

Abstract

   This document expands and clarifies the behavior of the Real-Time
   Transport Protocol (RTP) endpoints when they are sending multiple
   media streams in a single RTP session. In particular, issues
   involving Real-Time Transport Control Protocol (RTCP) messages are
   described.

   This document updates RFC 3550 in regards to handling of multiple
   SSRCs per endpoint in RTP sessions.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

   At the time The Real-Time Transport Protocol (RTP) [RFC3550] was
   originally written, and for quite some time after, endpoints in RTP
   sessions typically only transmitted a single media stream per RTP
   session, where separate RTP sessions were typically used for each
   distinct media type.

Recently, however, a number of scenarios have emerged (discussed
further in Section 3) in which endpoints wish to send multiple RTP
media streams, distinguished by distinct RTP synchronization source
(SSRC) identifiers, in a single RTP session. Although RTP's initial
design did consider such scenarios, the specification was not
consistently written with such use cases in mind. The specifications
are thus somewhat unclear.

The purpose of this document is to expand and clarify [RFC3550]'s
language for these use cases. The authors believe this does not
result in any major normative changes to the RTP specification,
however this document defines how the RTP specification is to be
interpreted. In these cases, this document updates RFC3550.

The document starts with terminology and some use cases where
multiple sources will occur. This is followed by some case studies
to try to identify issues that exist and need considerations. This
is followed by RTP and RTCP recommendations to resolve issues. Next
are security considerations and remaining open issues.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
"SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and
"OPTIONAL" in this document are to be interpreted as described in RFC
2119 [RFC2119] and indicate requirement levels for compliant
implementations.

3. Use Cases For Multi-Stream Endpoints

This section discusses several use cases that have motivated the
development of endpoints that send multiple streams in a single RTP
session.

3.1. Multiple-Capturer Endpoints

The most straightforward motivation for an endpoint to send multiple
media streams in a session is the scenario where an endpoint has
multiple capture devices of the same media type and characteristics.
For example, telepresence endpoints, of the type described by the
CLUE Telepresence Framework [I-D.ietf-clue-framework] is designed,
often have multiple cameras or microphones covering various areas of
a room.

3.2. Multi-Media Sessions

Recent work has been done in RTP
[I-D.ietf-avtcore-multi-media-rtp-session] and SDP

[I-D.ietf-mmusic-sdp-bundle-negotiation] to update RTP's historical assumption that media streams of different media types would always be sent on different RTP sessions. In this work, a single endpoint's audio and video media streams (for example) are instead sent in a single RTP session.

3.3. Multi-Stream Mixers

There are several RTP topologies which can involve a central device that itself generates multiple media streams in a session.

One example is a mixer providing centralized compositing for a multi-capture scenario like that described in Section 3.1. In this case, the centralized node is behaving much like a multi-capturer endpoint, generating several similar and related sources.

More complicated is the Source Projecting Mixer, see Section 3.6 of [I-D.ietf-avtcore-rtp-topologies-update]. This is a central box that receives media streams from several endpoints, and then selectively forwards modified versions of some of the streams toward the other endpoints it is connected to. Toward one destination, a separate media source appears in the session for every other source connected to the mixer, "projected" from the original streams, but at any given time many of them can appear to be inactive (and thus are receivers, not senders, in RTP). This sort of device is closer to being an RTP mixer than an RTP translator, in that it terminates RTCP reporting about the mixed streams, and it can re-write SSRCs, timestamps, and sequence numbers, as well as the contents of the RTP payloads, and can turn sources on and off at will without appearing to be generating packet loss. Each projected stream will typically preserve its original RTCP source description (SDES) information.

4. Multi-Stream Endpoint RTP Media Recommendations

While an endpoint MUST (of course) stay within its share of the available session bandwidth, as determined by signalling and congestion control, this need not be applied independently or uniformly to each media stream. In particular, session bandwidth MAY be reallocated among an endpoint's media streams, for example by varying the bandwidth use of a variable-rate codec, or changing the codec used by the media stream, up to the constraints of the session's negotiated (or declared) codecs. This includes enabling or disabling media streams as more or less bandwidth becomes available.

5. Multi-Stream Endpoint RTCP Recommendations

This section contains a number of different RTCP clarifications or recommendations that enables more efficient and simpler behavior without loss of functionality.

The RTP Control Protocol (RTCP) is defined in Section 6 of [RFC3550], but it is largely documented in terms of "participants". In many cases, the specification's recommendations for "participants" are to be interpreted as applying to individual media streams, rather than to endpoints. This section describes several concrete cases where this applies.

(tbd: rather than think in terms of media streams, it might be clearer to refer to SSRC values, where a participant with multiple active SSRC values counts as multiple participants, once per SSRC)

5.1. RTCP Reporting Requirement

For each of an endpoint's media streams, whether or not it is currently sending media, SR/RR and SDES packets MUST be sent at least once per RTCP report interval. (For discussion of the content of SR or RR packets' reception statistic reports, see [I-D.ietf-avtcore-rtp-multi-stream-optimisation].)

5.2. Initial Reporting Interval

When a new media stream is added to a unicast session, the sentence in [RFC3550]'s Section 6.2 applies: "For unicast sessions ... the delay before sending the initial compound RTCP packet MAY be zero." This applies to individual media sources as well. Thus, endpoints MAY send an initial RTCP packet for an SSRC immediately upon adding it to a unicast session.

This allowance also applies, as written, when initially joining a unicast session. However, in this case some caution needs to be exercised if the end-point or mixer has a large number of sources (SSRCs) as this can create a significant burst. How big an issue this depends on the number of source to send initial SR or RR and Session Description CNAME items for in relation to the RTCP bandwidth.

(tbd: Maybe some recommendation here? The aim in restricting this to unicast sessions was to avoid this burst of traffic, which the usual RTCP timing and reconsideration rules will prevent)

5.3. Compound RTCP Packets

Section 6.1 gives the following advice to RTP translators and mixers:

It is RECOMMENDED that translators and mixers combine individual
RTCP packets from the multiple sources they are forwarding into
one compound packet whenever feasible in order to amortize the
packet overhead (see Section 7). An example RTCP compound packet
as might be produced by a mixer is shown in Fig. 1. If the
overall length of a compound packet would exceed the MTU of the
network path, it SHOULD be segmented into multiple shorter
compound packets to be transmitted in separate packets of the
underlying protocol. This does not impair the RTCP bandwidth
estimation because each compound packet represents at least one
distinct participant. Note that each of the compound packets MUST
begin with an SR or RR packet.

Note: To avoid confusion, an RTCP packet is an individual item, such
as a Sender Report (SR), Receiver Report (RR), Source Description
(SDES), Goodbye (BYE), Application Defined (APP), Feedback [RFC4585]
or Extended Report (XR) [RFC3611] packet. A compound packet is the
combination of two or more such RTCP packets where the first packet
has to be an SR or an RR packet, and which contains a SDES packet
containing an CNAME item. Thus the above results in compound RTCP
packets that contain multiple SR or RR packets from different sources
as well as any of the other packet types. There are no restrictions
on the order in which the packets can occur within the compound
packet, except the regular compound rule, i.e., starting with an SR
or RR.

This advice applies to multi-media-stream endpoints as well, with the
same restrictions and considerations. (Note, however, that the last
sentence does not apply to AVPF [RFC4585] or SAVPF [RFC5124] feedback
packets if Reduced-Size RTCP [RFC5506] is in use.)

Due to RTCP's randomization of reporting times, there is a fair bit
of tolerance in precisely when an endpoint schedules RTCP to be sent.
Thus, one potential way of implementing this recommendation would be
to randomize all of an endpoint's sources together, with a single
randomization schedule, so an MTU's worth of RTCP all comes out
simultaneously.

(tbd: Multiplexing RTCP packets from multiple different sources might
require some adjustment to the calculation of RTCP's avg_rtcp_size,
as the RTCP group interval is proportional to avg_rtcp_size times the
group size).

6. RTCP Considerations for Streams with Disparate Rates

   It is possible for a single RTP session to carry streams of greatly
   differing bandwidth. There are two scenarios where this can occur.
   The first is when a single RTP session carries multiple flows of the
   same media type, but with very different quality; for example a video
   switching multi-point conference unit might send a full rate high-
   definition video stream of the active speaker but only thumbnails for
   the other participants, all sent in a single RTP session. The second
   scenarios occurs when audio and video flows are sent in a single RTP
   session, as discussed in [I-D.ietf-avtcore-multi-media-rtp-session].

   An RTP session has a single set of parameters that configure the
   session bandwidth, the RTCP sender and receiver fractions (e.g., via
   the SDP "b=RR:" and "b=RS:" lines), and the parameters of the RTP/
   AVPF profile [RFC4585] (e.g., trr-int) if that profile (or its secure
   extension, RTP/SAVPF [RFC5124]) is used. As a consequence, the RTCP
   reporting interval will be the same for every SSRC in an RTP session.
   This uniform RTCP reporting interval can result in RTCP reports being
   sent more often than is considered desirable for a particular media
   type. For example, if an audio flow is multiplexed with a high
   quality video flow where the session bandwidth is configured to match
   the video bandwidth, this can result in the RTCP packets having a
   greater bandwidth allocation than the audio data rate. If the
   reduced minimum RTCP interval described in Section 6.2 of [RFC3550]
   is used in the session, which might be appropriate for video where
   rapid feedback is wanted, the audio sources could be expected to send
   RTCP packets more often than they send audio data packets. This is
   most likely undesirable, and while the mismatch can be reduced
   through careful tuning of the RTCP parameters, particularly trr_int
   in RTP/AVPF sessions, it is inherent in the design of the RTCP timing
   rules, and affects all RTP sessions containing flows with mismatched
   bandwidth.

   Having multiple media types in one RTP session also results in more
   SSRCs being present in this RTP session. This increasing the amount
   of cross reporting between the SSRCs. From an RTCP perspective, two
   RTP sessions with half the number of SSRCs in each will be slightly
   more efficient. If someone needs either the higher efficiency due to
   the lesser number of SSRCs or the fact that one can't tailor RTCP
   usage per media type, they need to use independent RTP sessions.

When it comes to configuring RTCP the need for regular periodic reporting needs to be weighted against any feedback or control messages being sent. Applications using RTP/AVPF or RTP/SAVPF are RECOMMENDED to consider setting the trr-int parameter to a value suitable for the application's needs, thus potentially reducing the need for regular reporting and thus releasing more bandwidth for use for feedback or control.

Another aspect of an RTP session with multiple media types is that the RTCP packets, RTCP Feedback Messages, or RTCP XR metrics used might not be applicable to all media types. Instead, all RTP/RTCP endpoints need to correlate the media type of the SSRC being referenced in a message or packet and only use those that apply to that particular SSRC and its media type. Signalling solutions might have shortcomings when it comes to indicating that a particular set of RTCP reports or feedback messages only apply to a particular media type within an RTP session.

6.1. Timing out SSRCs

All SSRCs used in an RTP session MUST use the same timeout behaviour to avoid premature timeouts. This will depend on the RTP profile and its configuration. The RTP specification provides several options that can influence the values used when calculating the time interval. To avoid interoperability issues when using this specification, this document makes several clarifications to the calculations.

For RTP/AVP, RTP/SAVP, RTP/AVPF, and RTP/SAVPF with T_rr_interval = 0, the timeout interval SHALL be calculated using a multiplier of 5, i.e. the timeout interval becomes 5*Td. The Td calculation SHALL be done using a Tmin value of 5 seconds, not the reduced minimal interval even if used to calculate RTCP packet transmission intervals. If using either the RTP/AVPF or RTP/SAVPF profiles with T_rr_interval != 0 then the calculation as specified in Section 3.5.4 of RFC 4585 SHALL be used with a multiplier of 5, i.e. Tmin in the Td calculation is the T_rr_interval.

Note: If endpoints implementing the RTP/AVP and RTP/AVPF profiles (or their secure variants) are combined in a single RTP session, and the RTP/AVPF endpoints use a non-zero T_rr_interval that is significantly lower than 5 seconds, then there is a risk that the RTP/AVP endpoints will prematurely timeout the RTP/AVPF endpoints due to their different RTCP timeout intervals. Since an RTP session can only use a single RTP profile, this issue ought never occur. If such mixed RTP profiles are used, however, the RTP/AVPF session MUST NOT use a non-zero T_rr_interval that is smaller than 5 seconds.

(tbd: it has been suggested that a minimum non-zero T_rr_interval of
4 seconds is more appropriate, due to the nature of the timing
rules).

6.2. Tuning RTCP transmissions

This sub-section discusses what tuning can be done to reduce the
downsides of the shared RTCP packet intervals.

When using the RTP/AVP or RTP/SAVP profiles the tuning one can do is
very limited. The controls one has are limited to the RTCP bandwidth
values and whether the minimum RTCP interval is scaled according to
the bandwidth. As the scheduling algorithm includes both random
factors and reconsideration, one can't simply calculate the expected
average transmission interval using the formula for Td. But it does
indicate the important factors affecting the transmission interval,
namely the RTCP bandwidth available for the role (Active Sender or
Participant), the average RTCP packet size, and the number of SSRCs
classified in the relevant role. Note that if the ratio of senders
to total number of session participants is larger than the ratio of
RTCP bandwidth for senders in relation to the total RTCP bandwidth,
then senders and receivers are treated together.

Let's start with some basic observations:

a. Unless the scaled minimum RTCP interval is used, then Td prior to
    randomization and reconsideration can never be less than 5
    seconds (assuming default Tmin of 5 seconds).

b. If the scaled minimum RTCP interval is used, Td can become as low
    as 360 divided by RTP Session bandwidth in kilobits. In SDP the
    RTP session bandwidth is signalled using b=AS. An RTP Session
    bandwidth of 72 kbps results in Tmin being 5 seconds. An RTP
    session bandwidth of 360 kbps of course gives a Tmin of 1 second,
    and to achieve a Tmin equal to once every frame for a 25 Hz video
    stream requires an RTP session bandwidth of 9 Mbps! (The use of
    the RTP/AVPF or RTP/SAVPF profile allows a smaller Tmin, and
    hence more frequent RTCP reports, as discussed below).

c. Let's calculate the number (n) of SSRCs in the RTP session that
    5% of the session bandwidth can support to yield a Td value equal
    to Tmin with minimal scaling. For this calculation we have to
    make two assumptions. The first is that we will consider most or
    all SSRC being senders, resulting in everyone sharing the
    available bandwidth. Secondly we will select an average RTCP
    packet size. This packet will consist of an SR, containing (n-1)
    report blocks up to 31 report blocks, and an SDES item with at
    least a CNAME (17 bytes in size) in it. Such a basic packet will

be 800 bytes for n>=32. With these parameters, and as the
bandwidth goes up the time interval is proportionally decreased
(due to minimal scaling), thus all the example bandwidths 72
kbps, 360 kbps and 9 Mbps all support 9 SSRCs.

d. The actual transmission interval for a Td value is [0.5*Td/
    1.21828,1.5*Td/1.21828], which means that for Td = 5 seconds, the
    interval is actually [2.052,6.156] and the distribution is not
    uniform, but rather exponentially-increasing. The probability
    for sending at time X, given it is within the interval, is
    probability of picking X in the interval times the probability to
    randomly picking a number that is <=X within the interval with an
    uniform probability distribution. This results in that the
    majority of the probability mass is above the Td value.

To conclude, with RTP/AVP and RTP/SAVP the key limitation for small
unicast sessions is going to be the Tmin value. Thus the RTP session
bandwidth configured in RTCP has to be sufficiently high to reach the
reporting goals the application has following the rules for the
scaled minimal RTCP interval.

When using RTP/AVPF or RTP/SAVPF we get a quite powerful additional
tool, the setting of the T_rr_interval which has several effects on
the RTCP reporting. First of all as Tmin is set to 0 after the
initial transmission, the regular reporting interval is instead
determined by the regular bandwidth based calculation and the
T_rr_interval. This has the effect that we are no longer restricted
by the minimal interval or even the scaling rule for the minimal
rule. Instead the RTCP bandwidth and the T_rr_interval are the
governing factors. Now it also becomes important to separate between
the application's need for regular reports and RTCP feedback packet
types. In both regular RTCP mode, as in Early RTCP Mode, the usage
of the T_rr_interval prevents regular RTCP packets, i.e. packets
without any Feedback packets, to be sent more often than
T_rr_interval. This value is a hard as no regular RTCP packet can be
sent less than T_rr_interval after the previous regular packet
packet.

So applications that have a use for feedback packets for some media
streams, for example video streams, but don't want frequent regular
reporting for audio, could configure the T_rr_interval to a value so
that the regular reporting for both audio and video is at a level
that is considered acceptable for the audio. They could then use
feedback packets, which will include RTCP SR/RR packets, unless
reduced-size RTCP feedback packets [RFC5506] are used, and can
include other report information in addition to the feedback packet
that needs to be sent. That way the available RTCP bandwidth can be
focused for the use which provides the most utility for the
application.

Using T_rr_interval still requires one to determine suitable values
for the RTCP bandwidth value, in fact it might make it even more
important, as this is more likely to affect the RTCP behaviour and
performance than when using RTP/AVP, as there are fewer limitations
affecting the RTCP transmission.

When using T_rr_interval, i.e. having it be non zero, there are
configurations that have to be avoided. If the resulting Td value is
smaller but close to T_rr_interval then the interval in which the
actual regular RTCP packet transmission falls into becomes very
large, from 0.5 times T_rr_interval up to 2.73 times the
T_rr_interval. Therefore for configuration where one intends to have
Td smaller than T_rr_interval, then Td is RECOMMENDED to be targeted
at values less than 1/4th of T_rr_interval which results in that the
range becomes [0.5*T_rr_interval, 1.81*T_rr_interval].

With RTP/AVPF, using a T_rr_interval of 0 or with another low value
significantly lower than Td still has utility, and different
behaviour compared to RTP/AVP. This avoids the Tmin limitations of
RTP/AVP, thus allowing more frequent regular RTCP reporting. In fact
this will result that the RTCP traffic becomes as high as the
configured values.

(tbd: a future version of this memo will include examples of how to
choose RTCP parameters for common scenarios)

There exists no method within the specification for using different
regular RTCP reporting intervals depending on the media type or
individual media stream.

7. Security Considerations

In the secure RTP protocol (SRTP) [RFC3711], the cryptographic context of a compound SRTCP packet is the SSRC of the sender of the first RTCP (sub-)packet. This could matter in some cases, especially for keying mechanisms such as Mikey [RFC3830] which use per-SSRC keying.

Other than that, the standard security considerations of RTP apply; sending multiple media streams from a single endpoint does not appear to have different security consequences than sending the same number of streams.

8. Open Issues

At this stage this document contains a number of open issues. The below list tries to summarize the issues:

1. Further clarifications on how to handle the RTCP scheduler when sending multiple sources in one compound packet.

2. How is the RTCP avg_rtcp_size be calculated when RTCP packets are routinely multiplexed among multiple RTCP senders?

3. Do we need to provide a recommendation for unicast session joiners with many sources to not use 0 initial minimal interval from bit-rate burst perspective?

9. IANA Considerations

No IANA actions needed.

10. References

10.1. Normative References

[I-D.ietf-avtcore-6222bis]
          Begen, A., Perkins, C., Wing, D., and E. Rescorla,
          "Guidelines for Choosing RTP Control Protocol (RTCP)
          Canonical Names (CNAMEs)", draft-ietf-avtcore-6222bis-04
          (work in progress), June 2013.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
          Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V.
          Jacobson, "RTP: A Transport Protocol for Real-Time
          Applications", STD 64, RFC 3550, July 2003.

   [RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K.
              Norrman, "The Secure Real-time Transport Protocol (SRTP)",
              RFC 3711, March 2004.

   [RFC4585] 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.

   [RFC5124] Ott, J. and E. Carrara, "Extended Secure RTP Profile for
              Real-time Transport Control Protocol (RTCP)-Based Feedback
              (RTP/SAVPF)", RFC 5124, February 2008.

   [RFC5506] Johansson, I. and M. Westerlund, "Support for Reduced-Size
              Real-Time Transport Control Protocol (RTCP): Opportunities
              and Consequences", RFC 5506, April 2009.

10.2. Informative References

   [I-D.ietf-avtcore-multi-media-rtp-session]
              Westerlund, M., Perkins, C., and J. Lennox, "Multiple
              Media Types in an RTP Session", draft-ietf-avtcore-multi-
              media-rtp-session-02 (work in progress), February 2013.

   [I-D.ietf-avtcore-rtp-multi-stream-optimisation]
              Lennox, J., Westerlund, M., Wu, Q., and C. Perkins,
              "Sending Multiple Media Streams in a Single RTP Session:
              Grouping RTCP Reception Statistics and Other Feedback ",
              draft-ietf-avtcore-rtp-multi-stream-optimisation-00 (work
              in progress), July 2013.

   [I-D.ietf-avtcore-rtp-topologies-update]
              Westerlund, M. and S. Wenger, "RTP Topologies", draft-
              ietf-avtcore-rtp-topologies-update-00 (work in progress),
              April 2013.

   [I-D.ietf-clue-framework]
              Duckworth, M., Pepperell, A., and S. Wenger, "Framework
              for Telepresence Multi-Streams", draft-ietf-clue-
              framework-10 (work in progress), May 2013.

   [I-D.ietf-mmusic-sdp-bundle-negotiation]
              Holmberg, C., Alvestrand, H., and C. Jennings,
              "Multiplexing Negotiation Using Session Description
              Protocol (SDP) Port Numbers", draft-ietf-mmusic-sdp-
              bundle-negotiation-04 (work in progress), June 2013.

   [RFC3611] Friedman, T., Caceres, R., and A. Clark, "RTP Control
              Protocol Extended Reports (RTCP XR)", RFC 3611, November
              2003.

   [RFC3830] Arkko, J., Carrara, E., Lindholm, F., Naslund, M., and K.
              Norrman, "MIKEY: Multimedia Internet KEYing", RFC 3830,
              August 2004.

Appendix A.   Changes From Earlier Versions

   Note to the RFC-Editor: please remove this section prior to
   publication as an RFC.

A.1. Changes From WG Draft -00

   o Split the Reporting Group Extension from this draft into draft-
      ietf-avtcore-rtp-multi-stream-optimization-00.

   o Added RTCP tuning considerations from draft-ietf-avtcore-multi-
      media-rtp-session-02.

A.2. Changes From Individual Draft -02

   o Resubmitted as working group draft.

   o Updated references.

A.3. Changes From Individual Draft -01

   o Merged with draft-wu-avtcore-multisrc-endpoint-adver.

   o Changed how Reporting Groups are indicated in RTCP, to make it
      clear which source(s) is the group's reporting sources.

   o Clarified the rules for when sources can be placed in the same
      reporting group.

   o Clarified that mixers and translators need to pass reporting group
      SDES information if they are forwarding RR and SR traffic from
      members of a reporting group.

A.4. Changes From Individual Draft -00

   o Added the Reporting Group semantic to explicitly indicate which
      sources come from a single endpoint, rather than leaving it
      implicit.

o Specified that Reporting Group semantics (as they now are) apply
  to AVPF and XR, as well as to RR/SR report blocks.

o Added a description of the cascaded source-projecting mixer, along
  with a calculation of its RTCP overhead if reporting groups are
  not in use.

o Gave some guidance on how the flexibility of RTCP randomization
  allows some freedom in RTCP multiplexing.

o Clarified the language of several of the recommendations.

o Added an open issue discussing how avg_rtcp_size ought to be
  calculated for multiplexed RTCP.

o Added an open issue discussing how RTCP bandwidths are to be
  chosen for sessions where source bandwidths greatly differ.

Authors' Addresses

Jonathan Lennox
Vidyo, Inc.
433 Hackensack Avenue
Seventh Floor
Hackensack, NJ 07601
US

Email: jonathan@vidyo.com

Magnus Westerlund
Ericsson
Farogatan 6
SE-164 80 Kista
Sweden

Phone: +46 10 714 82 87
Email: magnus.westerlund@ericsson.com

Qin Wu
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: sunseawq@huawei.com

Colin Perkins
University of Glasgow
School of Computing Science
Glasgow G12 8QQ
United Kingdom

Email: csp@csperkins.org