Network Working Group
Request for Comments: 5117
Category: Informational

M. Westerlund
Ericsson
S. Wenger
Nokia

January 2008

RTP Topologies

Status of This Memo

Abstract

This document discusses multi-endpoint topologies used in Real-time Transport Protocol (RTP)-based environments. In particular, centralized topologies commonly employed in the video conferencing industry are mapped to the RTP terminology.

Table of Contents

1. Introduction

When working on the Codec Control Messages [CCM], considerable confusion was noticed in the community with respect to terms such as Multipoint Control Unit (MCU), Mixer, and Translator, and their usage in various topologies. This document tries to address this confusion by providing a common information basis for future discussion and specification work. It attempts to clarify and explain sections of the Real-time Transport Protocol (RTP) spec [RFC3550] in an informal way. It is not intended to update or change what is normatively specified within RFC 3550.

When the Audio-Visual Profile with Feedback (AVPF) [RFC4585] was

developed the main emphasis lay in the efficient support of point to point and small multipoint scenarios without centralized multipoint control. However, in practice, many small multipoint conferences operate utilizing devices known as Multipoint Control Units (MCUs). MCUs may implement Mixer or Translator (in RTP [RFC3550] terminology)

functionality and signalling support. They may also contain additional application functionality. This document focuses on the media transport aspects of the MCU that can be realized using RTP, as discussed below. Further considered are the properties of Mixers and Translators, and how some types of deployed MCUs deviate from these properties.

2. Definitions

2.1. Glossary

ASM - Any Source Multicast
AVPF - The Extended RTP Profile for RTCP-based Feedback
CSRC - Contributing Source
Link - The data transport to the next IP hop
MCU - Multipoint Control Unit
Path - The concatenation of multiple links, resulting in an end-to-end data transfer.
PtM - Point to Multipoint
PtP - Point to Point
SSM - Source-Specific Multicast
SSRC - Synchronization Source

2.2. Indicating Requirement Levels

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 [RFC2119].

The RFC 2119 language is used in this document to highlight those important requirements and/or resulting solutions that are necessary to address the issues raised in this document.

3. Topologies

This subsection defines several basic topologies that are relevant

for codec control. The first four relate to the RTP system model utilizing multicast and/or unicast, as envisioned in RFC 3550. The last two topologies, in contrast, describe the deployed system models as used in many H.323 [H323] video conferences, where both the media streams and the RTP Control Protocol (RTCP) control traffic terminate at the MCU. In these two cases, the media sender does not receive the (unmodified or Translator-modified) Receiver Reports from all sources (which it needs to interpret based on Synchronization Source (SSRC) values) and therefore has no full information about all the endpoint's situation as reported in RTCP Receiver Reports (RRs). More topologies can be constructed by combining any of the models; see Section 3.8.

The topologies may be referenced in other documents by a shortcut name, indicated by the prefix "Topo-".

For each of the RTP-defined topologies, we discuss how RTP, RTCP, and the carried media are handled. With respect to RTCP, we also introduce the handling of RTCP feedback messages as defined in [RFC4585] and [CCM]. Any important differences between the two will be illuminated in the discussion.

3.1. Point to Point

Shortcut name: Topo-Point-to-Point

The Point to Point (PtP) topology (Figure 1) consists of two endpoints, communicating using unicast. Both RTP and RTCP traffic are conveyed endpoint-to-endpoint, using unicast traffic only (even if, in exotic cases, this unicast traffic happens to be conveyed over an IP-multicast address).

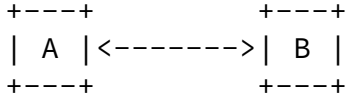


Figure 1 - Point to Point

The main property of this topology is that A sends to B, and only B, while B sends to A, and only A. This avoids all complexities of handling multiple endpoints and combining the requirements from them. Note that an endpoint can still use multiple RTP Synchronization

Sources (SSRCs) in an RTP session.

RTCP feedback messages for the indicated SSRCs are communicated directly between the endpoints. Therefore, this topology poses minimal (if any) issues for any feedback messages.

3.2. Point to Multipoint Using Multicast

Shortcut name: Topo-Multicast

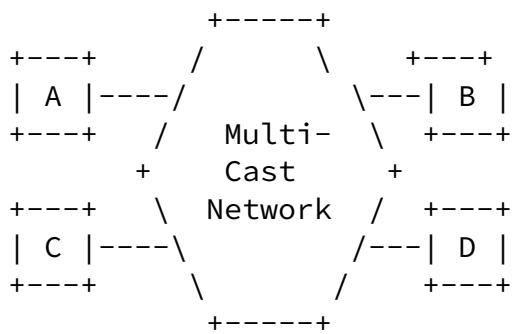


Figure 2 - Point to Multipoint Using Multicast

Point to Multipoint (PtM) is defined here as using a multicast topology as a transmission model, in which traffic from any participant reaches all the other participants, except for cases such as:

- packet loss, or

- when a participant does not wish to receive the traffic for a specific multicast group and therefore has not subscribed to the IP-multicast group in question. This is for the cases where a multi-media session is distributed using two or more multicast groups.

In the above context, "traffic" encompasses both RTP and RTCP traffic. The number of participants can vary between one and many, as RTP and RTCP scale to very large multicast groups (the theoretical limit of the number of participants in a single RTP session is approximately two billion). The above can be realized using Any Source Multicast (ASM). Source-Specific Multicast (SSM) may be also be used with RTP. However, then only the designated source may reach all receivers. Please review [RTCP-SSM] for how RTCP can be made to work in combination with SSM.

This document is primarily interested in that subset of multicast sessions wherein the number of participants in the multicast group is so low that it allows the participants to use early or immediate feedback, as defined in AVPF [RFC4585]. This document refers to those groups as "small multicast groups".

RTCP feedback messages in multicast will, like media, reach everyone (subject to packet losses and multicast group subscription). Therefore, the feedback suppression mechanism discussed in [RFC4585]

is required. Each individual node needs to process every feedback message it receives to determine if it is affected or if the feedback message applies only to some other participant.

3.3. Point to Multipoint Using the RFC 3550 Translator

Shortcut name: Topo-Translator

Two main categories of Translators can be distinguished:

Transport Translators (Topo-Trn-Translator) do not modify the media stream itself, but are concerned with transport parameters. Transport parameters, in the sense of this section, comprise the transport addresses (to bridge different domains) and the media packetization to allow other transport protocols to be interconnected

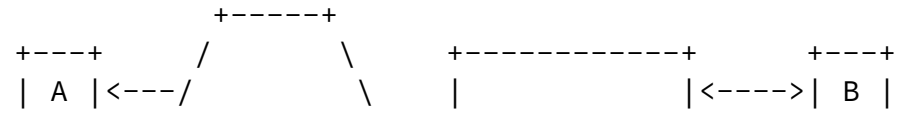
to a session (in gateways). Of the transport Translators, this memo is primarily interested in those that use RTP on both sides, and this is assumed henceforth. Translators that bridge between different protocol worlds need to be concerned about the mapping of the SSRC/CSRC (Contributing Source) concept to the non-RTP protocol. When designing a Translator to a non-RTP-based media transport, one crucial factor lies in how to handle different sources and their identities. This problem space is not discussed henceforth.

Media Translators (Topo-Media-Translator), in contrast, modify the media stream itself. This process is commonly known as transcoding. The modification of the media stream can be as small as removing parts of the stream, and it can go all the way to a full transcoding (down to the sample level or equivalent) utilizing a different media codec. Media Translators are commonly used to connect entities without a common interoperability point.

Stand-alone Media Translators are rare. Most commonly, a combination of Transport and Media Translators are used to translate both the media stream and the transport aspects of a stream between two transport domains (or clouds).

Both Translator types share common attributes that separate them from Mixers. For each media stream that the Translator receives, it generates an individual stream in the other domain. A Translator always keeps the SSRC for a stream across the translation, where a Mixer can select a media stream, or send them out mixed, always under its own SSRC, using the CSRC field to indicate the source(s) of the content.

The RTCP translation process can be trivial, for example, when Transport Translators just need to adjust IP addresses, or they can be quite complex as in the case of media Translators. See Section 7.2 of [RFC3550].



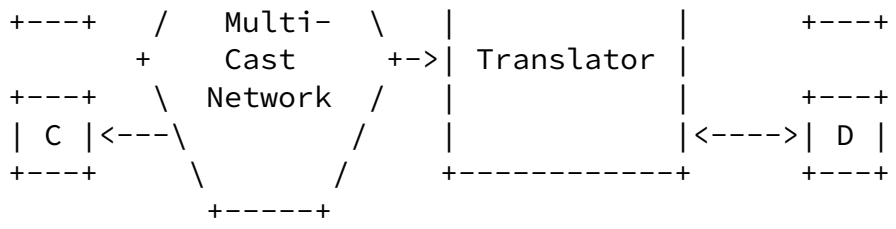


Figure 3 - Point to Multipoint Using a Translator

Figure 3 depicts an example of a Transport Translator performing at least IP address translation. It allows the (non-multicast-capable) participants B and D to take part in a multicast session by having the Translator forward their unicast traffic to the multicast addresses in use, and vice versa. It must also forward B's traffic to D, and vice versa, to provide each of B and D with a complete view of the session.

If B were behind a limited network path, the Translator may perform media transcoding to allow the traffic received from the other participants to reach B without overloading the path.

When, in the example depicted in Figure 3, the Translator acts only as a Transport Translator, then the RTCP traffic can simply be forwarded, similar to the media traffic. However, when media translation occurs, the Translator's task becomes substantially more complex, even with respect to the RTCP traffic. In this case, the Translator needs to rewrite B's RTCP Receiver Report before forwarding them to D and the multicast network. The rewriting is needed as the stream received by B is not the same stream as the other participants receive. For example, the number of packets transmitted to B may be lower than what D receives, due to the different media format. Therefore, if the Receiver Reports were forwarded without changes, the extended highest sequence number would indicate that B were substantially behind in reception, while it most likely it would not be. Therefore, the Translator must translate that number to a corresponding sequence number for the stream the Translator received. Similar arguments can be made for most other fields in the RTCP Receiver Reports.

As specified in Section 7.1 of [RFC3550], the SSRC space is common

for all participants in the session, independent of on which side they are of the Translator. Therefore, it is the responsibility of the participants to run SSRC collision detection, and the SSRC is a field the Translator should not change.

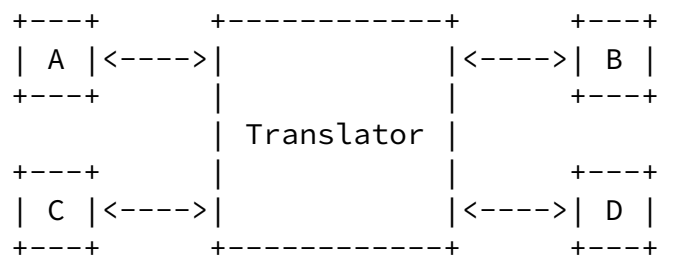


Figure 4 - RTP Translator (Relay) with Only Unicast Paths

Another Translator scenario is depicted in Figure 4. Herein, the Translator connects multiple users of a conference through unicast. This can be implemented using a very simple transport Translator, which in this document is called a relay. The relay forwards all traffic it receives, both RTP and RTCP, to all other participants. In doing so, a multicast network is emulated without relying on a multicast-capable network infrastructure.

A Translator normally does not use an SSRC of its own, and is not visible as an active participant in the session. One exception can be conceived when a Translator acts as a quality monitor that sends RTCP reports and therefore is required to have an SSRC. Another example is the case when a Translator is prepared to use RTCP feedback messages. This may, for example, occur when it suffers packet loss of important video packets and wants to trigger repair by the media sender, by sending feedback messages. To be able to do this it needs to have a unique SSRC.

A media Translator may in some cases act on behalf of the "real" source and respond to RTCP feedback messages. This may occur, for example, when a receiver requests a bandwidth reduction, and the media Translator has not detected any congestion or other reasons for bandwidth reduction between the media source and itself. In that case, it is sensible that the media Translator reacts to the codec control messages itself, for example, by transcoding to a lower media rate. If it were not reacting, the media quality in the media sender's domain may suffer, as a result of the media sender adjusting its media rate (and quality) according to the needs of the slow past-Translator endpoint, at the expense of the rate and quality of all other session participants.

In general, a Translator implementation should consider which RTCP feedback messages or codec-control messages it needs to understand in relation to the functionality of the Translator itself. This is completely in line with the requirement to also translate RTCP messages between the domains.

3.4. Point to Multipoint Using the RFC 3550 Mixer Model

Shortcut name: Topo-Mixer

A Mixer is a middlebox that aggregates multiple RTP streams, which are part of a session, by mixing the media data and generating a new RTP stream. One common application for a Mixer is to allow a participant to receive a session with a reduced amount of resources.

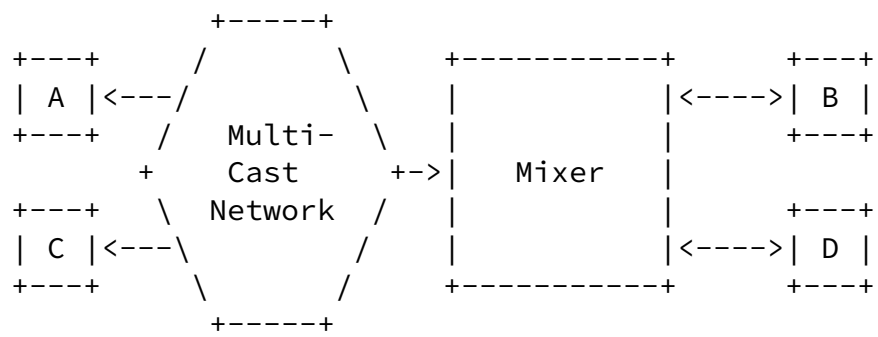


Figure 5 - Point to Multipoint Using the RFC 3550 Mixer Model

A Mixer can be viewed as a device terminating the media streams received from other session participants. Using the media data from the received media streams, a Mixer generates a media stream that is sent to the session participant.

The content that the Mixer provides is the mixed aggregate of what the Mixer receives over the PtP or PtM paths, which are part of the same conference session.

The Mixer is the content source, as it mixes the content (often in the uncompressed domain) and then encodes it for transmission to a participant. The CSRC Count (CC) and CSRC fields in the RTP header are used to indicate the contributors of to the newly generated stream. The SSRCs of the to-be-mixed streams on the Mixer input appear as the CSRCs at the Mixer output. That output stream uses a unique SSRC that identifies the Mixer's stream. The CSRC are forwarded between the two domains to allow for loop detection and identification of sources that are part of the global session. Note that Section 7.1 of RFC 3550 requires the SSRC space to be shared

between domains for these reasons.

The Mixer is responsible for generating RTCP packets in accordance with its role. It is a receiver and should therefore send reception reports for the media streams it receives. In its role as a media sender, it should also generate Sender Reports for those media streams sent. As specified in Section 7.3 of RFC 3550, a Mixer must not forward RTCP unaltered between the two domains.

The Mixer depicted in Figure 5 is involved in three domains that need to be separated: the multicast network, participant B, and participant D. The Mixer produces different mixed streams to B and D, as the one to B may contain content received from D, and vice versa. However, the Mixer only needs one SSRC in each domain that is the receiving entity and transmitter of mixed content.

In the multicast domain, a Mixer still needs to provide a mixed view of the other domains. This makes the Mixer simpler to implement and avoids any issues with advanced RTCP handling or loop detection, which would be problematic if the Mixer were providing non-symmetric behavior. Please see Section 3.7 for more discussion on this topic.

A Mixer is responsible for receiving RTCP feedback messages and handling them appropriately. The definition of "appropriate" depends on the message itself and the context. In some cases, the reception of a codec-control message may result in the generation and transmission of RTCP feedback messages by the Mixer to the participants in the other domain. In other cases, a message is handled by the Mixer itself and therefore not forwarded to any other domain.

When replacing the multicast network in Figure 5 (to the left of the Mixer) with individual unicast paths as depicted in Figure 6, the Mixer model is very similar to the one discussed in Section 3.6 below. Please see the discussion in Section 3.6 about the differences between these two models.

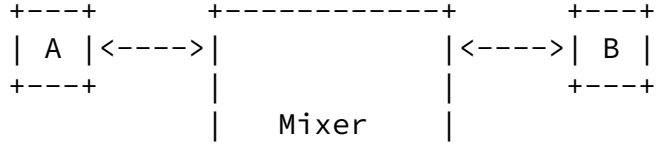


Figure 6 - RTP Mixer with Only Unicast Paths

3.5. Point to Multipoint Using Video Switching MCUs

Shortcut name: Topo-Video-switch-MCU

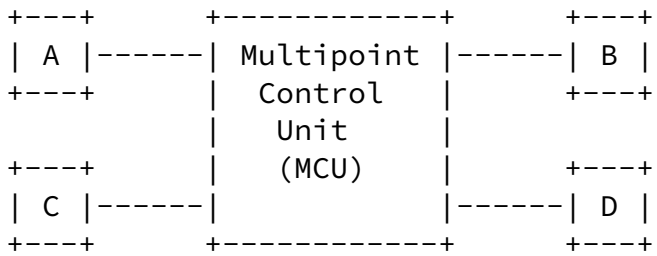


Figure 7 - Point to Multipoint Using a Video Switching MCU

This PtM topology is still deployed today, although the RTCP-terminating MCUs, as discussed in the next section, are perhaps more common. This topology, as well as the following one, reflect today's lack of wide availability of IP multicast technologies, as well as the simplicity of content switching when compared to content mixing. The technology is commonly implemented in what is known as "Video Switching MCUs".

A video switching MCU forwards to a participant a single media stream, selected from the available streams. The criteria for selection are often based on voice activity in the audio-visual conference, but other conference management mechanisms (like presentation mode or explicit floor control) are known to exist as well.

The video switching MCU may also perform media translation to modify the content in bit-rate, encoding, or resolution. However, it still may indicate the original sender of the content through the SSRC. In

this case, the values of the CC and CSRC fields are retained.

If not terminating RTP, the RTCP Sender Reports are forwarded for the currently selected sender. All RTCP Receiver Reports are freely forwarded between the participants. In addition, the MCU may also originate RTCP control traffic in order to control the session and/or report on status from its viewpoint.

The video switching MCU has most of the attributes of a Translator. However, its stream selection is a mixing behavior. This behavior has some RTP and RTCP issues associated with it. The suppression of all but one media stream results in most participants seeing only a subset of the sent media streams at any given time, often a single stream per conference. Therefore, RTCP Receiver Reports only report on these streams. Consequently, the media senders that are not currently forwarded receive a view of the session that indicates

their media streams disappear somewhere en route. This makes the use of RTCP for congestion control, or any type of quality reporting, very problematic.

To avoid the aforementioned issues, the MCU needs to implement two features. First, it needs to act as a Mixer (see Section 3.4) and forward the selected media stream under its own SSRC and with the appropriate CSRC values. Second, the MCU needs to modify the RTCP RRs it forwards between the domains. As a result, it is RECOMMENDED that one implement a centralized video switching conference using a Mixer according to RFC 3550, instead of the shortcut implementation described here.

3.6. Point to Multipoint Using RTCP-Terminating MCU

Shortcut name: Topo-RTCP-terminating-MCU

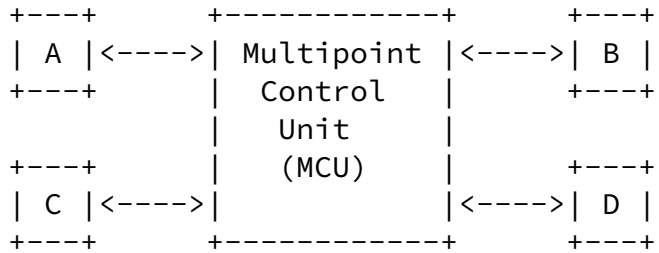


Figure 8 - Point to Multipoint Using Content Modifying MCUs

In this PtM scenario, each participant runs an RTP point-to-point session between itself and the MCU. This is a very commonly deployed topology in multipoint video conferencing. The content that the MCU provides to each participant is either:

a) a selection of the content received from the other participants, or

b) the mixed aggregate of what the MCU receives from the other PtP paths, which are part of the same conference session.

In case a), the MCU may modify the content in bit-rate, encoding, or resolution. No explicit RTP mechanism is used to establish the relationship between the original media sender and the version the MCU sends. In other words, the outgoing sessions typically use a different SSRC, and may well use a different payload type (PT), even if this different PT happens to be mapped to the same media type. This is a result of the individually negotiated session for each participant.

In case b), the MCU is the content source as it mixes the content and then encodes it for transmission to a participant. According to RTP [RFC3550], the SSRC of the contributors are to be signalled using the CSRC/CC mechanism. In practice, today, most deployed MCUs do not implement this feature. Instead, the identification of the participants whose content is included in the Mixer's output is not indicated through any explicit RTP mechanism. That is, most deployed MCUs set the CSRC Count (CC) field in the RTP header to zero, thereby indicating no available CSRC information, even if they could identify the content sources as suggested in RTP.

The main feature that sets this topology apart from what RFC 3550 describes is the breaking of the common RTP session across the centralized device, such as the MCU. This results in the loss of explicit RTP-level indication of all participants. If one were using the mechanisms available in RTP and RTCP to signal this explicitly, the topology would follow the approach of an RTP Mixer. The lack of explicit indication has at least the following potential problems:

1) Loop detection cannot be performed on the RTP level. When carelessly connecting two misconfigured MCUs, a loop could be generated.

2) There is no information about active media senders available in the RTP packet. As this information is missing, receivers cannot use it. It also deprives the client of information related to currently active senders in a machine-usable way, thus preventing clients from indicating currently active speakers in user interfaces, etc.

Note that deployed MCUs (and endpoints) rely on signalling layer mechanisms for the identification of the contributing sources, for example, a SIP conferencing package [RFC4575]. This alleviates, to some extent, the aforementioned issues resulting from ignoring RTP's CSRC mechanism.

As a result of the shortcomings of this topology, it is RECOMMENDED to instead implement the Mixer concept as specified by RFC 3550.

3.7. Non-Symmetric Mixer/Translators

Shortcut name: Topo-Asymmetric

It is theoretically possible to construct an MCU that is a Mixer in one direction and a Translator in another. The main reason to consider this would be to allow topologies similar to Figure 5, where the Mixer does not need to mix in the direction from B or D towards the multicast domains with A and C. Instead, the media streams from

B and D are forwarded without changes. Avoiding this mixing would save media processing resources that perform the mixing in cases where it isn't needed. However, there would still be a need to mix B's stream towards D. Only in the direction B -> multicast domain or D -> multicast domain would it be possible to work as a Translator. In all other directions, it would function as a Mixer.

The Mixer/Translator would still need to process and change the RTCP before forwarding it in the directions of B or D to the multicast domain. One issue is that A and C do not know about the mixed-media stream the Mixer sends to either B or D. Thus, any reports related

to these streams must be removed. Also, receiver reports related to A and C's media stream would be missing. To avoid A and C thinking that B and D aren't receiving A and C at all, the Mixer needs to insert its Receiver Reports for the streams from A and C into B and D's Sender Reports. In the opposite direction, the Receiver Reports from A and C about B's and D's stream also need to be aggregated into the Mixer's Receiver Reports sent to B and D. Since B and D only have the Mixer as source for the stream, all RTCP from A and C must be suppressed by the Mixer.

This topology is so problematic and it is so easy to get the RTCP processing wrong, that it is NOT RECOMMENDED to implement this topology.

3.8. Combining Topologies

Topologies can be combined and linked to each other using Mixers or Translators. However, care must be taken in handling the SSRC/CSRC space. A Mixer will not forward RTCP from sources in other domains, but will instead generate its own RTCP packets for each domain it mixes into, including the necessary Source Description (SDES) information for both the CSRCs and the SSRCs. Thus, in a mixed domain, the only SSRCs seen will be the ones present in the domain, while there can be CSRCs from all the domains connected together with a combination of Mixers and Translators. The combined SSRC and CSRC space is common over any Translator or Mixer. This is important to facilitate loop detection, something that is likely to be even more important in combined topologies due to the mixed behavior between the domains. Any hybrid, like the Topo-Video-switch-MCU or Topo-Asymmetric, requires considerable thought on how RTCP is dealt with.

4. Comparing Topologies

The topologies discussed in Section 3 have different properties. This section first lists these properties and then maps the different

topologies to them. Please note that even if a certain property is supported within a particular topology concept, the necessary functionality may, in many cases, be optional to implement.

4.1. Topology Properties

4.1.1. All to All Media Transmission

Multicast, at least Any Source Multicast (ASM), provides the functionality that everyone may send to, or receive from, everyone else within the session. MCUs, Mixers, and Translators may all provide that functionality at least on some basic level. However, there are some differences in which type of reachability they provide.

The transport Translator function called "relay", in Section 3.3, is the one that provides the emulation of ASM that is closest to true IP-multicast-based, all to all transmission. Media Translators, Mixers, and the MCU variants do not provide a fully meshed forwarding on the transport level; instead, they only allow limited forwarding of content from the other session participants.

The "all to all media transmission" requires that any media transmitting entity considers the path to the least capable receiver. Otherwise, the media transmissions may overload that path. Therefore, a media sender needs to monitor the path from itself to any of the participants, to detect the currently least capable receiver, and adapt its sending rate accordingly. As multiple participants may send simultaneously, the available resources may vary. RTCP's Receiver Reports help performing this monitoring, at least on a medium time scale.

The transmission of RTCP automatically adapts to any changes in the number of participants due to the transmission algorithm, defined in the RTP specification [RFC3550], and the extensions in AVPF [RFC4585] (when applicable). That way, the resources utilized for RTCP stay within the bounds configured for the session.

4.1.2. Transport or Media Interoperability

Translators, Mixers, and RTCP-terminating MCU all allow changing the media encoding or the transport to other properties of the other domain, thereby providing extended interoperability in cases where the participants lack a common set of media codecs and/or transport protocols.

4.1.3. Per Domain Bit-Rate Adaptation

Participants are most likely to be connected to each other with a heterogeneous set of paths. This makes congestion control in a Point to Multipoint set problematic. For the ASM and "relay" scenario, each individual sender has to adapt to the receiver with the least capable path. This is no longer necessary when Media Translators, Mixers, or MCUs are involved, as each participant only needs to adapt to the slowest path within its own domain. The Translator, Mixer, or MCU topologies all require their respective outgoing streams to adjust the bit-rate, packet-rate, etc., to adapt to the least capable path in each of the other domains. That way one can avoid lowering the quality to the least-capable participant in all the domains at the cost (complexity, delay, equipment) of the Mixer or Translator.

4.1.4. Aggregation of Media

In the all to all media property mentioned above and provided by ASM, all simultaneous media transmissions share the available bit-rate. For participants with limited reception capabilities, this may result in a situation where even a minimal acceptable media quality cannot be accomplished. This is the result of multiple media streams needing to share the available resources. The solution to this problem is to provide for a Mixer or MCU to aggregate the multiple streams into a single one. This aggregation can be performed according to different methods. Mixing or selection are two common methods.

4.1.5. View of All Session Participants

The RTP protocol includes functionality to identify the session participants through the use of the SSRC and CSRC fields. In addition, it is capable of carrying some further identity information about these participants using the RTCP Source Descriptors (SDES). To maintain this functionality, it is necessary that RTCP is handled correctly in domain bridging function. This is specified for Translators and Mixers. The MCU described in Section 3.5 does not entirely fulfill this. The one described in Section 3.6 does not support this at all.

4.1.6. Loop Detection

In complex topologies with multiple interconnected domains, it is possible to form media loops. RTP and RTCP support detecting such loops, as long as the SSRC and CSRC identities are correctly set in forwarded packets. It is likely that loop detection works for the MCU, described in Section 3.5, at least as long as it forwards the RTCP between the participants. However, the MCU in Section 3.6 will definitely break the loop detection mechanism.

4.2. Comparison of Topologies

The table below attempts to summarize the properties of the different topologies. The legend to the topology abbreviations are: Topo-Point-to-Point (PtP), Topo-Multicast (Multic), Topo-Trns-Translator (TTrn), Topo-Media-Translator (including Transport Translator) (MTrn), Topo-Mixer (Mixer), Topo-Asymmetric (ASY), Topo-Video-switch-MCU (MCUs), and Topo-RTCP-terminating-MCU (MCUt). In the table below, Y indicates Yes or full support, N indicates No support, (Y) indicates partial support, and N/A indicates not applicable.

Property	PtP	Multic	TTrn	MTrn	Mixer	ASY	MCUs	MCUt
All to All media	N	Y	Y	Y	(Y)	(Y)	(Y)	(Y)
Interoperability	N/A	N	Y	Y	Y	Y	N	Y
Per Domain Adaptation	N/A	N	N	Y	Y	Y	N	Y
Aggregation of media	N	N	N	N	Y	(Y)	Y	Y
Full Session View	Y	Y	Y	Y	Y	Y	(Y)	N
Loop Detection	Y	Y	Y	Y	Y	Y	(Y)	N

Please note that the Media Translator also includes the transport Translator functionality.

5. Security Considerations

The use of Mixers and Translators has impact on security and the security functions used. The primary issue is that both Mixers and Translators modify packets, thus preventing the use of integrity and source authentication, unless they are trusted devices that take part

in the security context, e.g., the device can send Secure Realtime Transport Protocol (SRTP) and Secure Realtime Transport Control Protocol (SRTCP) [RFC3711] packets to session endpoints. If encryption is employed, the media Translator and Mixer need to be able to decrypt the media to perform its function. A transport Translator may be used without access to the encrypted payload in cases where it translates parts that are not included in the encryption and integrity protection, for example, IP address and UDP

port numbers in a media stream using SRTP [RFC3711]. However, in general, the Translator or Mixer needs to be part of the signalling context and get the necessary security associations (e.g., SRTP crypto contexts) established with its RTP session participants.

Including the Mixer and Translator in the security context allows the entity, if subverted or misbehaving, to perform a number of very serious attacks as it has full access. It can perform all the attacks possible (see RFC 3550 and any applicable profiles) as if the media session were not protected at all, while giving the impression to the session participants that they are protected.

Transport Translators have no interactions with cryptography that works above the transport layer, such as SRTP, since that sort of Translator leaves the RTP header and payload unaltered. Media Translators, on the other hand, have strong interactions with cryptography, since they alter the RTP payload. A media Translator in a session that uses cryptographic protection needs to perform cryptographic processing to both inbound and outbound packets.

A media Translator may need to use different cryptographic keys for the inbound and outbound processing. For SRTP, different keys are required, because an RFC 3550 media Translator leaves the SSRC unchanged during its packet processing, and SRTP key sharing is only allowed when distinct SSRCs can be used to protect distinct packet streams.

When the media Translator uses different keys to process inbound and outbound packets, each session participant needs to be provided with the appropriate key, depending on whether they are listening to the Translator or the original source. (Note that there is an architectural difference between RTP media translation, in which participants can rely on the RTP Payload Type field of a packet to

determine appropriate processing, and cryptographically protected media translation, in which participants must use information that is not carried in the packet.)

When using security mechanisms with Translators and Mixers, it is possible that the Translator or Mixer could create different security associations for the different domains they are working in. Doing so has some implications:

First, it might weaken security if the Mixer/Translator accepts a weaker algorithm or key in one domain than in another. Therefore, care should be taken that appropriately strong security parameters are negotiated in all domains. In many cases, "appropriate"

translates to "similar" strength. If a key management system does allow the negotiation of security parameters resulting in a different strength of the security, then this system SHOULD notify the participants in the other domains about this.

Second, the number of crypto contexts (keys and security related state) needed (for example, in SRTP [RFC3711]) may vary between Mixers and Translators. A Mixer normally needs to represent only a single SSRC per domain and therefore needs to create only one security association (SRTP crypto context) per domain. In contrast, a Translator needs one security association per participant it translates towards, in the opposite domain. Considering Figure 3, the Translator needs two security associations towards the multicast domain, one for B and one for D. It may be forced to maintain a set of totally independent security associations between itself and B and D respectively, so as to avoid two-time pad occurrences. These contexts must also be capable of handling all the sources present in the other domains. Hence, using completely independent security associations (for certain keying mechanisms) may force a Translator to handle N*DM keys and related state; where N is the total number of SSRCs used over all domains and DM is the total number of domains.

There exist a number of different mechanisms to provide keys to the different participants. One example is the choice between group keys and unique keys per SSRC. The appropriate keying model is impacted by the topologies one intends to use. The final security properties

are dependent on both the topologies in use and the keying mechanisms' properties, and need to be considered by the application. Exactly which mechanisms are used is outside of the scope of this document.

6. Acknowledgements

The authors would like to thank Bo Burman, Umesh Chandra, Roni Even, Keith Lantz, Ladan Gharai, Geoff Hunt, and Mark Baugher for their help in reviewing this document.

7. References

7.1. Normative References

[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.

[RFC4575] Rosenberg, J., Schulzrinne, H., and O. Levin, Ed., "A Session Initiation Protocol (SIP) Event Package for Conference State", RFC 4575, August 2006.

[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.

7.2. Informative References

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

[H323] ITU-T Recommendation H.323, "Packet-based multimedia communications systems", June 2006.

[RTCP-SSM] J. Ott, J. Chesterfield, E. Schooler, "RTCP Extensions for Single-Source Multicast Sessions with Unicast Feedback," Work in Progress, March 2007.

Authors' Addresses

Magnus Westerlund
Ericsson Research
Ericsson AB
SE-164 80 Stockholm, SWEDEN

Phone: +46 8 7190000
EMail: magnus.westerlund@ericsson.com

Stephan Wenger
Nokia Corporation
P.O. Box 100
FIN-33721 Tampere
FINLAND

Phone: +358-50-486-0637
EMail: stewe@stewe.org

Full Copyright Statement

Intellectual Property