Network Working Group
Internet-Draft
Intended status: Informational
Expires: December 22, 2012

M. Westerlund
Ericsson

June 20, 2012

Multiparty Requirements in Congestion Control for Real-Time Interactive Media

draft-westerlund-wscc-multiparty-requirements-00

Abstract

There exist many different types of applications that have need for real-time interactive media transport. Many of these applications communicate with multiple parties concurrently, for example audio conferencing, video conferencing and telepresence. A common method of establishing multi-party applications is to use one or more central nodes providing transport and media functions. This memo discusses these multiparty solutions and what requirements they put on any solution for congestion control for real-time interactive media.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction 3
2. Multi-party Topologies 3
  2.1. Mesh 3
  2.2. Media Mixer 4
  2.3. Media Switching Mixer 5
3. Requirements 6
4. References 7
Author’s Address 7

1. Introduction

Interactive Real-time media is used in a number of different types of communication applications. A fairly large part of those applications support multiparty communication. As availability of IP multicast is limited to only some environments, most of these applications uses some sort of overlay topology. Commonly based on one or more centralized middleboxes receiving media, optionally processing the media, then sending the media to the other participants in the communication session.

There is a trend in the world for the need of standardized components for real-time media transport and congestion control. Such a standard must be able to handle the heterogeneous networks of the world as well as the ever growing upper-limits in bandwidth consumption fueled by high resolution displays and cameras combined with sufficient processing available for relatively low prices. The increased usage of video conference and telepresence has further driven the demand for multi-vendor interoperable solutions and the need for standardization.

It is critical that any standardization work within the field of congestion control takes the various applications into account and meet their requirements. This document focus on presenting a number of ways multi-party communication sessions are established, with a particulare focus on the ones using RTP [RFC3550], and what demands they put on the congestion control. First a number of used topologies are presented and then followed by a derivation of some requirements these put on a solution.

2. Multi-party Topologies

This section considers various multi-party topologies that are in use and highlights what is relevant for congestion control. It will not discuss IP multicast or RTP Transport Translators (Relay), although they are discussed in RTP Topologies [RFC5117]. This is due to that these topologies are not as commonly used. They also present additional restrictions and requirements, making them different problems and likely best handled separately from the below topologies. The below ones are based on unicast and uses middleboxes, with the exception of Mesh, that can modify the sent media streams in the middlebox.

2.1. Mesh

The Mesh topology is when each end-point establishes direct unicast based communication with each of the conference's peers. This is depicted below (Figure 1) where end-point A has one RTP session and media exchange with B and a separate with C, but within the context of a joint conference.

```
                          +---+      +---+
                          | A |<---->| B |
                          +---+      +---+
                            ^          ^
                             \        /
                              \      /
                               v    v
                               +---+
                               | C |
                               +---+
```

Figure 1: Mesh using Multiple Unicast sessions

This creates a situation where one application have multiple flows competing for the capacity, possibly in the same bottleneck. An important differences in the case of real-time media is that the two different media flows to B and C is likely to come from the same set of media sources, they may even be the same encoding to reduce the resource consumption in the end-point. This results in a tight relationship between the flows. These needs to be taken into account when performing congestion control.

The first aspect is when they are the same media source. This results in that each encoding produced from the same media source are likely to have the same variations of bit-rate for variable bit-rate codecs due to that all codec instances will experience the same content properties and variations in difficulties to encode it. Variable bit-rate video is the prime example where an scene change or other significant change of the video image requires significant increase in number of bits to provide similar quality to the previous video frame. Thus several media streams will have bit-rate spikes at the same time, rather than independently.

Secondly, if the media streams sent to the different destinations (B and C above) are produced by a single encoding instance, then congestion control will need to at all times use the restrictions from the path that is most restrictive.

2.2. Media Mixer

A Media Mixer is a central node that is common in deployments today. Its basic operation is to receive media, decode it and use the decoded media stream part of a media mix or composition that is produced and encoded for a particular destination. Figure 2 illustrate this by considering the media to be sent from the Mixer to end-point A. In a basic case that will include media from all the participants B, C and D. Thus for A it decodes B, C and D, creates a mix with just them and encodes it according to the current capacity of the path to A from the Mixer.

```
+---+      +------------+      +---+
| A |<---->|            |<---->| B |
+---+      |            |      +---+
           |   Mixer    |
+---+      |            |      +---+
| C |<---->|            |<---->| D |
+---+      +------------+      +---+
```

Figure 2: RTP Mixer with Unicast Paths

From a congestion control perspective a Media Mixer is easy node type to deal with. It deals with each path independently. When C sends media to the Mixer it can independently adapt the media for the path from C to the Mixer. Then the mixer produces different mixes and encodings from itself down to each of the session participants. Thus almost total independency between the paths. The media quality delivered in the received media stream will of course depend on the combination of the two paths capacity and the media quality reduction created by the decoding, mixing and encoding operations happening in the Mixer.

There exist one optimization that could benefit this implementation, and that is that the Mixer might want to reduce the media properties to lower values than what the path between the encoding end-point and the mixer can sustain. The reason for this is that there is no consuming end-point that can utilize the higher quality. For example the path C to the Mixer may sustain video in Full HD quality but none of the paths from the Mixer to A, B and D support the bit-rate required to utilize the Full HD quality input, instead a lower resolution video at high SNR is a better match for the application. Thus there is a relation between controlling the codec at the media sender and combine this with knowledge of the current capacity on the different paths in use by the multi-party session.

2.3. Media Switching Mixer

A different type of Mixer is the Media Switching mixer, the important property of this class of mixers is that they forward one or more media stream being received by the mixer to a receiver. The set of streams being concurrently received by an end-point will be different between the end-points.

The fact that the mixer forwards rather than re-encodes the media enables higher media quality and less complexity demands on the mixer, thus making this method of implementation attractive. From a congestion control perspective this creates additional challenges as the mixer needs to ensure that congestion is controlled on all paths. Several possible methods for solutions exist here. One could be to expose the original media sender to the fact that its media stream goes over a number of different paths and let the sender ensure that the media matches all the paths, e.g. media sender A will see a path A->Mixer->B, another A->Mixer->C and a third in A->Mixer->D.

Another choice is to have the mixer hide the different paths, but still provide the media sender with a combined set of limitations representing the paths from the Mixer to the receivers. This could at a basic level be accomplished using the Temporary Maximum Media Bit-rate Request (TMMBR) Codec Control Message [RFC5104].

The applications using this type of mixers commonly use either scalable encoding or simulcast to provide the mixer with more than one quality tier, and thus bit-rates to select from when forwarding. This enables at least course grained bit-rate control. However, more fine grained control and adaptation of the scalability layers or simulcast versions to better suit the actual path limitations are of interest.

These mixer have application logic selecting which media streams is the most suitable to provide, a dynamic process that changes with activity in a conference. This logic will interact with the congestion control as the mix of media streams being forwarded over a particular path will change, thus affecting the available capacity for each particular media stream simply by changing the set of media streams.

3. Requirements

The above descriptions points to a couple of different requirements that a real-time interactive congestion control solution of handling multiparty conferences need to deal with:

1. Handle sending multiple flow instances of its own media sources across a shared bottle-neck

2. Handle limitations from media sources due to codec or other path limitations sharing encoding

3. Provide a solution on how a central node can handle the situation of having one path from sender to central node, and then

potentially multiple paths from the central node to the media receivers.

The above requirements are all a result of having multiparty media sessions. These will create additional complexities compared with a solution only targeting single point to point transmissions. However, failure to take these requirements and the above usages into account will significantly reduce the utility of any real-time interactive media congestion control solution.

4. References

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

[RFC5104] Wenger, S., Chandra, U., Westerlund, M., and B. Burman, "Codec Control Messages in the RTP Audio-Visual Profile with Feedback (AVPF)", RFC 5104, February 2008.

[RFC5117] Westerlund, M. and S. Wenger, "RTP Topologies", RFC 5117, January 2008.

Author’s Address

Magnus Westerlund
Ericsson
Farogatan 6
SE-164 80 Kista
Sweden

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