

Alkit Reflex

RTP reflector/mixer

Mathias Johanson, Ph.D.

Alkit Communications

Introduction

Real time audio and video communication over IP networks is attracting a lot of interest for applications like audio/video conferencing, IP telephony, distributed collaborative work, telemedicine, and more. The RTP protocol [1] has emerged as the preferred transport protocol for real time data over packet networks. RTP communication can be either point-to-point (e.g. IP telephony) or multipoint (e.g. videoconferencing). In a multipoint communication situation, something has to be done to avoid or minimize the amount of duplicate media packets transmitted over the network links interconnecting the participating hosts. If each participant was to transmit one distinct media stream to each of the other participants, it is easy to see that the network soon will be overloaded. What is needed is some form of group communication mechanism that makes it possible to transmit a data packet to a group of hosts instead of sending a unique packet destined to each participating host. One such mechanism is IP multicast, but unfortunately multicast deployment has been hampered with technical problems, and consequently IP multicast is not widely available in commercial networks.

When IP multicast is unavailable, or only available in limited parts of the network, an application level approach can be taken to support multipoint communication. In this case, dedicated hosts known as *reflectors* are configured to relay the RTP media packets inbetween the hosts participating in a conference session.

Overview

Alkit Reflex is an RTP/SIP based audio/video reflector and mixer for multipoint communication in IP networks with limited or no multicast support. All types of RTP traffic can be reflected: audio, video, or other. Audio mixing and transcoding is supported for GSM, DVI and PCM (8 bit or 16 bit) audio.

The reflector can be used as a packet distribution server for multipoint conferences in networks that do not support IP multicast. It can also be used as a bridge (or gateway) between multicast routing domains and non-multicast capable network domains.

A client application is needed to set up RTP sessions on the reflector. Each participating host must signal its preferred RTP media for a particular session, identified by a port number and a protocol number. (Currently only the UDP protocol is supported, i.e. only RTP sessions over UDP can be handled by the reflector.) The session setup signaling is done using the *Session Initiation Protocol* (SIP) [2]. When a SIP based voice/video application is used as the RTP media transmitter/receiver, the SIP signaling is built into the teleconferencing application. If some other, non-SIP RTP client is used, a standalone session initiation application called 'refsipclient' can be used, to do the session setup signaling.

Design and implementation of Alkit Reflex

Alkit Reflex is a software product, available for many popular operating system platforms, including Windows, Linux, Solaris, HP-UX and Irix. The software is configured through a text file to provide RTP reflection services on a specified range of UDP ports. Access control lists can be specified to limit the usage of the reflector to a well-defined set of hosts. The RTP payload types that will be allowed by the reflector can also be configured.

The full set of configuration options is listed in Appendix 1.

The session concept

The concept of a *conference session* or a *RTP session*, or simply a *session*, is used heavily in this document and in related literature. A session is an abstraction of a number of (two or more) hosts

communicating in real time, typically using audio and/or video. A communication session is limited in time, and typically requires some sort of signaling mechanism to initiate and terminate the session.

In Alkit Reflex, an RTP session is identified by a UDP port number and a textual name (such as "session1" or "project meeting"). By using a textual name in addition to the port number to identify a session, multiple conference sessions using the same UDP port number can be active on the reflector simultaneously.

Session setup signaling

To set up a conference session, the participants signal their intention to participate in the session to the reflector, indicating the UDP port number to be used, and identifying the particular conference session to join by a textual name. The RTP media types to be used are also signaled. Note that different UDP ports are to be used for audio and video RTP sessions. The port numbers that can be used for audio and video respectively, are configured in the configuration file (reflex.conf). RTP data MUST use even port numbers (as specified in RFC3550, [3]). Odd port numbers will be used for RTCP data, providing session management and control, and quality feedback.

As previously mentioned, Alkit Reflex relies on the *Session Initiation Protocol* (SIP) for the setup signaling between the endpoint hosts and the reflector. The *Session Description Protocol* (SDP) [4] is used in the payload of the SIP INVITE messages to specify the RTP payload types (i.e. the media encodings) and the bandwidth limits to be used for the session. The "session name" field of the SDP message is used to convey the textual name identifying the session.

The SDP messages can also contain an attribute instructing the reflector that a particular host is only interested in sending (or receiving) RTP data. By default, RTP data is assumed to be both sent and received by each client. Moreover, a bandwidth limit can be specified during the session setup phase, instructing the reflector not to transmit data at a rate exceeding the limit.

When a participant of a conference session wants to leave the session, a SIP BYE message is transmitted to the reflector, after which the reflector will stop relaying RTP data to the host.

RTP data exchange

Once a session has been set up, Alkit Reflex will start relaying RTP packets inbetween the participants of the session. Session membership is dynamic, and hosts can join and leave at any time. Alkit Reflex enforces that the RTP clients conform to transmitting only RTP traffic of the type specified in the session setup signaling. Thus, if a host wants to be able to switch between different media encodings in the middle of the session, all RTP data types to be used throughout the session must be specified in the session setup phase.

Media transcoding and bandwidth limiting

In case a bandwidth limit is specified in the session setup phase for a particular participant of a session, Alkit Reflex will apply *media transcoding* and *rate limiting* to meet the bandwidth limit. In case of audio, the encoding of the outgoing audio stream to the bandwidth-limited participant will be chosen to meet the specified bandwidth limit; in case of video, the frame rate of each outgoing stream will be adapted to meet the bandwidth limit.

Inactivity timeout

If a host has signaled membership of a conference session, but no RTP or RTCP packets has been received by the reflector for some (configurable) time period, the reflector will consider the client to have left silently, and will delete the host from the conference session. The inactivity timeout period can be configured through the configuration file, or disabled altogether with a command line switch.

Firewall traversal support

Generally, firewalls do not let UDP packet streams, such as audio and video streams, originated from outside the firewall to pass through. This complicates the use of audio/video conferencing tools, since the firewalls must be manually configured to let the traffic pass. In firewalls employing network address translation (NAT), port redirection must be configured on the firewall for each user that wishes to use audio and video conferencing.

To aid the users in this situation, Alkit Reflex implements an automatic firewall traversal mechanism that in many cases makes firewall traversal transparent to the user. However, a requirement is that the reflector itself is not located behind a firewall.

The firewall traversal mechanism implemented by Alkit Reflex is described "Multimedia Communication, Collaboration and Conferencing using Alkit Confero" [5].

Using multiple reflectors

For large multipoint sessions in sparse network environments (i.e. many widely distributed participants), multiple reflectors can be used to further reduce the load on the network. This is particularly appropriate if several clusters of users are interconnected. The hosts of each user cluster then connects to their own reflector, and the reflectors are interconnected and configured to relay the packets from each host cluster inbetween them. An example of this type of usage is given in Figure 1 below.

This type of advanced reflector usage typically requires some degree of knowledge of the topology of the network interconnecting the participating hosts.

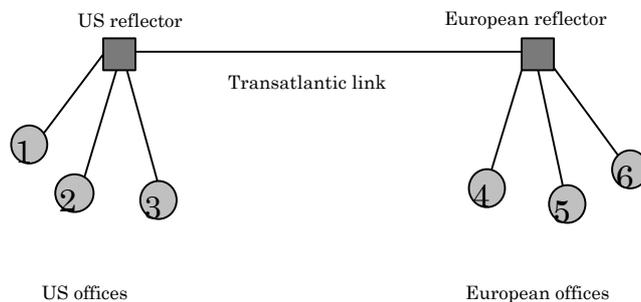


Figure 1 An example of a two reflector configuration interconnecting three US sites with three European sites of a large corporation. This set-up saves bandwidth on the transatlantic link, compared to a one-reflector configuration.

Reflectors and multicast

In situations where IP multicast is supported only in certain parts of an internetwork, the hosts connected to a multicast-capable network segment can communicate using multicast, and then one or more reflectors can be used to interconnect the non-multicast capable hosts. Furthermore, multiple "islands" of multicast-capable networks can be interconnected using the reflector, in which case the reflector acts as an application-level multicast tunnel. Alkit Reflex supports all of the abovementioned multicast/unicast interconnection scenarios.

Summary and conclusions

This document has provided an overview of the Alkit Reflex RTP reflector/mixer. The need for application level RTP reflectors was found to be motivated by the fact that an efficient packet distribution mechanism for group communication is needed, in order to make services such as audio and video conferencing feasible in networks lacking true multicast support. In network environments where IP multicast is only partially available, Alkit Reflex can successfully be used as an application level multicast gateway, interconnecting multicast-capable networks with networks lacking multicast support.

Alkit Reflex relies on state-of-the-art IETF protocols for session setup signaling (SIP) and media transport (RTP), and can be successfully used with a variety of client applications, for many different applications, including audio teleconferencing, videoconferencing, and more. Alkit Reflex also supports a multitude of advanced features, such as firewall traversal support, rate limiting, and access control.

For more information about Alkit Reflex, and related products, please contact Alkit Communications:

Alkit Communications AB
Aurorum 2
S-977 75 Luleå
E-mail: info@alkit.se
<http://www.alkit.se>

References

- [1] H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, "RTP: A transport protocol for real-time applications," IETF RFC 3550, July 2003.
- [2] J. Rosenberg et al., "SIP: Session initiation protocol," IETF RFC 3261, June 2002.
- [3] H. Schulzrinne, S. Casner, "RTP profile for audio and video conferences with minimal control," RFC 3551, July 2003.
- [4] M. Handley and V. Jacobson, "SDP: Session description protocol," IETF RFC 2327, April 1998.
- [5] M. Johanson, "Multimedia Communication, Collaboration and Conferencing using Alkit Confero", Alkit technical report 2004:1, June 2004, document available online at http://confero.alkit.se/confero_whitepaper.pdf.

Appendix 1: Configuration options

Alkit Reflex is configured by a configuration file called 'reflex.conf'. (The name of the configuration file can be changed by using the "-c" command line option.) In the configuration file, a hash symbol (#) at the beginning of a line is treated as a comment. The following parameters can be configured:

```
LICENSE_FILE <license>
```

This indicates to Reflex where the license file is located. <license> should be the path to the license file.

```
TRAFFIC_DESCR <name> <RTP payload type> [<RTP payload type> ...]
```

This line tells Reflex which RTP payload types are legitimate, and gives a collection of RTP payload types a logical name (the <name> parameter), such as "audio" or "video".

```
UDP_PORT <port> <mode> <max_clients> <max_lurkers> <traffic descr>
```

Tells Reflex to start reflecting RTP packets on the specified UDP port. The <port> parameter must be an even number in the range 1024-65535. (Odd port numbers are used for the corresponding RTCP data.) <mode> should be either DUMB, to reflect packets, or MIX to mix together the ingress RTP streams to one egress stream (for audio only). <traffic descr> must match one of the traffic descriptors specified with the TRAFFIC_DESCR lines.

```
UDP_RANGE <low port #> <hi port #> <mode> <max_clients> <max_lurkers>  
<traffic descr>
```

Same as UDP_PORT but for a range of UDP port numbers.

```
SERVICE_CLASS <id> <r|w|rw> <RTP payload type> [<RTP payload type>  
...]
```

A service class is a collection of RTP payload types (belonging to the same traffic descriptor) and a specifier telling whether the RTP stream is read only, write only, or read write. Service classes are used internally in the reflector, but are of little interest otherwise. In most situations it is sufficient to make sure that all RTP types that are going to be used on the reflector are present in some service class, and that this service class is read/write (rw).

```
FW_TRAV_PORT <port> <data port>
```

Enable firewall traversal support for these UDP ports. This can be used to enable a transparent firewall passthrough mechanism that is implemented in Alkit Confero.

```
ADMIN_PORT 5160
```

TCP port number used for sending administrative commands to the reflector. You can telnet to this port and type in HELP to see a list of available administration commands.

```
HOSTS_ALLOW <all | network/mask [network/mask ...]>
```

Clients belonging to the specified networks are allowed to connect.

```
HOSTS_DENY <all | network/mask [network/mask ...]>
```

Clients belonging to the specified networks are NOT allowed to connect.

```
IP_ADDRESS <IP address> (or IP_ADDRESS_EXTERNAL <IP address>)
```

Sets the reflector's IP address. Useful in case the reflector has several network interfaces, or alias IP addresses, or if the reflector for some reason is unable to figure out the proper IP address to use. If not specified, reflex finds the IP address by doing gethostname/gethostbyname (i.e. DNS lookup of the host name).

`IP_ADDRESS_LOCAL <local IP address>`

Specifies the reflector's local IP address. Useful in cases when the reflector is behind a NAT firewall (or on a NAT firewall machine) with several network interfaces, one of which is the externally visible address (specified by `IP_ADDRESS` or `IP_ADDRESS_EXTERNAL`, see above) and one is the local area network interface, serving local clients. Using a configuration with two configured IP addresses like this, the reflector can be successfully used as a firewall traversal bridge, interconnecting local conference participants with externally connected participants.

Note that `IP_ADDRESS_LOCAL` requires that `IP_ADDRESS` (`IP_ADDRESS_EXTERNAL`) is also specified.