Setting up a reflector-reflector interconnection using Alkit Reflex RTP reflector/mixer

Mathias Johanson
Alkit Communications AB

Introduction

The Alkit Reflex reflector/mixer system can be set-up to interconnect two or more reflectors. The conference sessions on the interconnected reflectors will then be synchronized so that the participants of a session can join any one of the interconnected reflectors to participate in a meeting. One of the reasons for such reflector interconnections are to support structured and efficient firewall traversal of real-time conferencing traffic between internal (e.g. enterprise) and external networks (e.g. the Internet). In this situation, participants of a session who are connected to the enterprise network connect to the internal reflector, whereas participants who are outside the enterprise network connect to the external reflector.

Firewall configuration

The firewall configuration example below shows how two reflectors can be interconnected through a corporate firewall. In the dual firewall example configuration shown in Figure 1, one of the reflectors (R1) is connected to the public Internet, while the other (R2) is in a DMZ of an enterprise network.

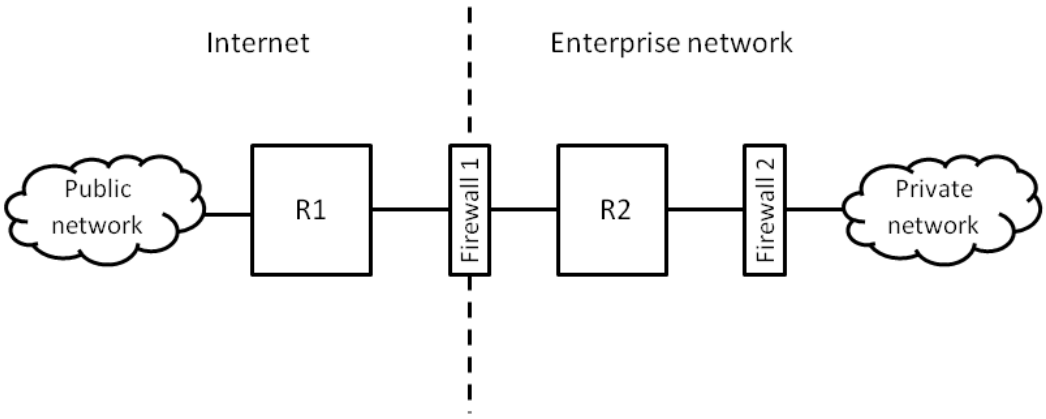

Figure 1: Example of reflector-reflector interconnection for corporate firewall traversal.

With this set-up, the firewall between the two reflectors to be interconnected (Firewall 1 in figure 1) should be configured to let the following traffic pass between the reflectors:

Port number (or range)	Protocol	Description
5564	UDP	RTP relay
5565	UDP	RTCP relay
5565	TCP	File transfer relay
6000 to 6000+n	TCP	Application sharing relay, for n simultaneously shared applications

The firewall rules of Firewall 1 should thus disallow any traffic not originating from one of the two reflectors, and any traffic not having destination port number and protocol as indicated in the table above.

The port number range used for application sharing relay, specified as ranging from 6000 to 6000+n in the table above, reflects the fact that application sharing sessions are multiplexed by port number on the reflector interconnection. Thus, n should be set to the maximum number of simultaneously shared applications that should be supported, e.g. if 100 simultaneous shares are needed, the range of port numbers is from 6000 to 6100.

It should be noted that all port numbers are configurable in the Alkit Reflex system, so the actual ports listed in the table are just examples.

The initiation of the TCP and UDP communication sessions can be from either reflector (R1 or R2 in Figure 1), depending on whether a session is initiated by a user on the enterprise network or from the external network (Internet).

The firewall between the internal reflector (R2) and the private (enterprise) network (Firewall 2) should be configured to let the following traffic pass:

Port number (or range)	Protocol	Description
5060	TCP & UDP	SIP, Session set-up signaling
5565	TCP	File transfers
5566	UDP	RTP audio
5567	UDP	RTCP audio
5568	UDP	RTP video
5569	UDP	RTCP video
5570-5573	UDP	RTP/RTCP NAT traversal keep-alive
12340	TCP	Checkip
6000 to 6000+n	TCP	Application sharing (client), n simultaneous shares
7000 to 7000+n	TCP	Application sharing (server), n simultaneous shares

Firewall 2 must allow traffic between any host on the private network and reflector R2, using the port numbers specified in the table. The firewall can optionally be configured to allow only TCP and UDP sessions initiated from within the private network to pass through. In this case the firewall must not have explicit port forwarding rules for the ports in the table. Network address translation (NAT) is supported.

Reflector configuration

The reflector interconnection is configured in Alkit Reflex by the REFLECTOR_INTERCONNECTION config file parameter. For example, if R1 has IP address 10.11.12.13 and R2 has IP address 10.11.12.14, the following configuration file line in the config file of R1 sets up an interconnection to R2 using UDP port 5564 as the RTP relay port and port 5565 as the RTCP relay port.

```
REFLECTOR_INTERCONNECTION 10.11.12.14 5564
```

(Note that the RTCP relay port is implicitly selected as the RTP relay port plus one.)

Conversely, R2's configuration file should have the following line:

```
REFLECTOR_INTERCONNECTION 10.11.12.13 5564
```

The reflectors will in this case use UDP port number 5564 both as source and destination port number for RTP relay. Port number 5565 will be used for RTCP relay (both source and destination port).

Example scenario

To illustrate how the inter-reflector communication works, consider an example scenario wherein two users (u1 and u2) connected to the private (enterprise) network of Figure 1 communicate using audio, video and application sharing in a conferencing session with two external users user (u3 and u4). The situation is illustrated in Figure 2.

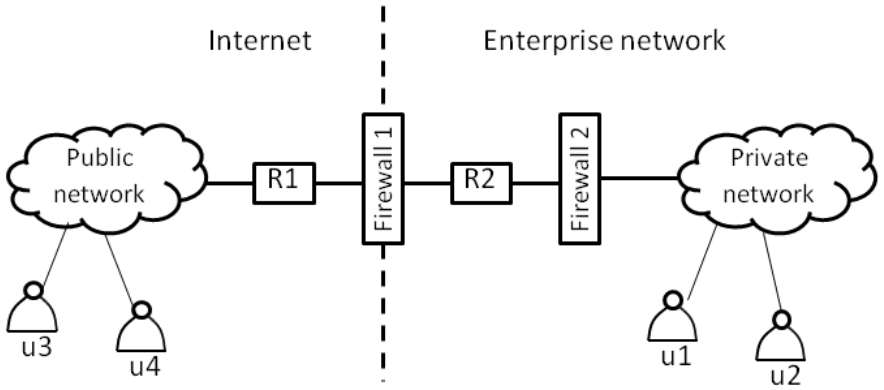

Figure 2: Example scenario for inter-reflector communication in a four-party conference

In this scenario, the users on the enterprise network (u1 and u2) will connect to the reflector R2. This is signaled using the SIP protocol, traversing Firewall 2. Users u1 and u2 are now part of a conferencing session on reflector R2. The session is identified by a textual name, which is part of the SIP INVITE messages. Now u1 and u2 can start sending audio and video streams to each other via the reflector, using the RTP protocol over UDP with port numbers as configured for R2 and Firewall 2. The RTCP protocol is used for session management and control purposes and is also transported over UDP, one RTCP stream per RTP media stream. An RTCP message specifically designed for inter-reflector RTP relay is sent from the R2 reflector to the R1 reflector, since they have been configured to be interconnected, as described above. This makes reflector R1 aware of the fact that there is a session, identified by a

particular text string, which has two participants on reflector R2. So far there is no user in this session on R1, so no relay of RTP streams is initiated.

Now the users u3 and u4 connect to the external reflector, R1, using the SIP protocol for session initiation, with the same session identifier (which has been agreed upon beforehand). The SIP session set-up includes a challenge/response authentication phase with username and password. Authentication can optionally be enabled also for the R2 reflector, but in this scenario the access to the R2 reflector is considered safe, since it is only reachable by SIP from the private network. The joining of u3 and u4 triggers inter-reflector RTCP session join packets from R1 to R2. Both reflectors will now know that there is a session with participants on both reflectors, so the RTP and RTCP streams originated by u1 and u2 are relayed from R2 to R1, which reflects them further on to u3 and u4. Conversely, the RTP and RTCP streams originated by u3 and u4 are relayed from R1 to R2, and reflected from R2 to u1 and u2.

Now let's say that user u1 wants to share an application on his or her desktop with all other users. This is done by sending an RTCP packet requesting an application sharing session to be established on reflector R2. R2 grants the request by sending an RTCP response packet back to u1 containing a TCP port number to use for the sharing and a passkey for the session. The TCP port number is chosen from the configured range of application sharing server ports (i.e. in the 7000 to 7000+n range that Firewall 2 is configured to allow). Upon reception of this RTCP packet, the application sharing component on u1's computer sets up a TCP connection to R2 using the TCP port number supplied. After the TCP connection is established, the passkey is used for authentication of u1. Reflector R2 now sends RTCP announcements to all participants of the session saying that there is an application being shared by user u1. The RTCP announcement contains the IP address of the reflector (R2), a TCP port number from the application sharing client range (i.e. between 6000 and 6000+n, as configured for Firewall 1 and Firewall 2) and a passkey. The RTCP application sharing announcement will be sent to u2 directly from R2, so that u2 upon reception can set up a TCP connection to R2 with the specified port number, authenticated by the passkey, and then the shared application will be accessible to u2. For u3 and u4, the RTCP announcement will be relayed over the inter-reflector RTCP relay connection (using UDP port 5565 through Firewall 1 with the configuration above) to R1, which, before reflecting it to u3 and u4 will modify the IP address and port number of the packet to R1's IP address and an available port in R1's application sharing client range. When u3 and u4 receive the announcement, they establish TCP connections to R1 on the given port, whereupon R1 establishes proxy TCP connections to R2, using the destination port that R2 announced (which was subsequently re-written) in the inter-reflector application sharing client range, which Firewall 1 has been configured to allow to pass through. The TCP connections for application sharing are now set up from u3 and u4 to R2, via proxy connections in R1, so that the application shared by u1 can be accessed by u3 and u4.

When u1 wishes to stop the application sharing, the TCP connection to R1 is closed, triggering the TCP connection from u2 and the proxy connections from u3 and u4 to be closed, terminating the application sharing.

The established inter-reflector relay of media streams will continue until both users leave from either R1 or R2 (which is done through SIP BYE messages). There is also an inactivity timeout feature in the reflector, which will consider a participant to have left if no RTCP packet is seen for two minutes. This is intended to detect participants that are disconnected unexpectedly by e.g. a network disruption or client system crash.