Application Note

Call Logging in an IP Environment

Recording Calls Using Dialogic® HMP Software and Open Source Components

Executive Summary

Call logging is an important function in the contact center. As contact centers move to an IP environment, methods of performing call logging stand to change. This application note describes a "proxied RTP" system for call logging, where IP calls are redirected through the call logging system. It discusses the potential advantages and disadvantages of a proxied RTP system approach, as well as the system's architecture and message flow.

Table of Contents

Introduction

Contact centers use call logging to record calls for a variety of purposes, such as agent training or legal requirements. As contact centers move from a TDM to an IP environment, call logging will continue to be a useful function; however, the method for performing call logging will change.

This application note covers a methodology for call logging in an IP environment. The described approach uses the SIP Express Router (SER), a high-performance, configurable, open source SIP (RFC3261) server, as well as a combination RTP proxy/logger, which is also maintained as one of the adjunct projects related to SER.

The RTP proxy is mainly used for NAT firewall traversal. In call logging, its SER interface and session management capabilities are used, and its RTP packet relay is adapted for use with the IP Media Library (IPML) API and Dialogic® Host Media Processing Software Release 3.1LIN. The result is that a full-duplex audio stream can be placed on the "virtual SCbus," which is created by Dialogic HMP Software, and recorded to disk.

Using Dialogic HMP Software as a media manager in the proxied RTP system described here has a number of advantages:

- Easier application programming using a familiar API
- Increased channel density
- Enables use of low-bit-rate voice encoders, resulting in less disk space needed for recording
- Provides full control of the recording process, including termination
- Easier access to voice streams for operations, such as streaming to an ASR server for word spotting

Two disadvantages to this type of system are as follows:

- The system alters the architecture of a call center when it is implemented
- A point of failure is created when a call is routed through the logging system (if it were to fail, call center operation would be impacted)

The SER is available for use in a UNIX or Linux environment. Since it is a relatively lightweight application, it can reside on the same Linux system as the Dialogic HMP Software 3.1. For this reason, this application note describes a single-chassis approach. However, a larger application that includes additional functionality, or an implementation that requires greater channel density, could follow the basic single-chassis model described here, but distribute the components among multiple boxes.

Note: The methodology described in this application note is not intended to be used as a complete call center solution, which likely would require one or more other components. However, for the sake of clarity, only call logging is discussed.

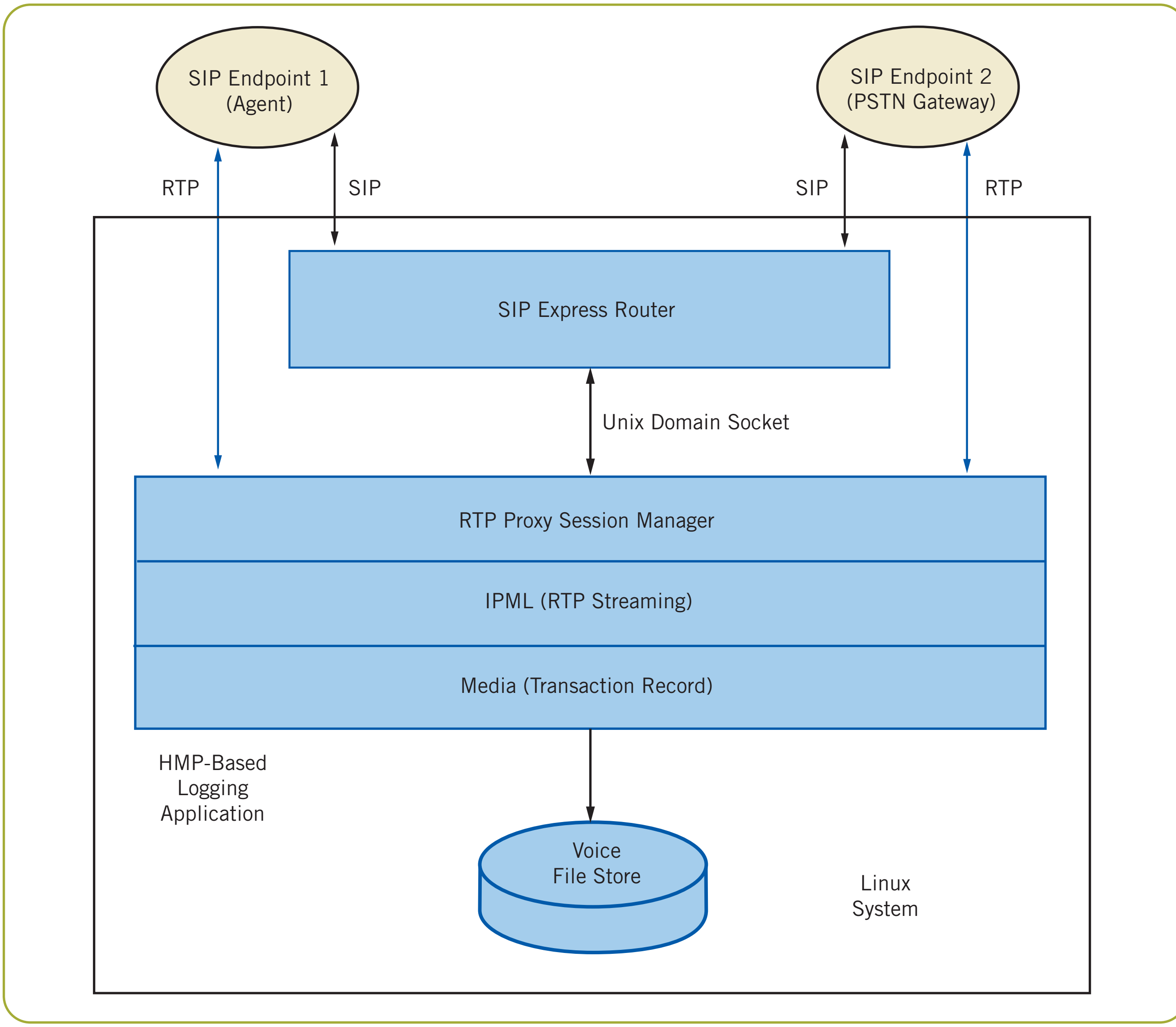

Figure 1. High-Level Architecture

High-Level Architecture

Figure 1 is an illustration of the high-level architecture of the call logging system discussed in this application note. The following sections discuss the two main components, SER and RTP Proxy Session Manager, and how they are used.

SER

The SER functions as a SIP proxy and a control process for the recording component. From the point of view of the SIP endpoints (User Agents or UAs) in the network, SER plays a fairly standard role as a SIP proxy. Outbound calls from the endpoints are directed to the proxy, where a database lookup is used to decide the final destination of the call. The router delivers the SIP message to the destination endpoint, but before doing so, modifies the RTP connection information in the Session Descriptor Protocol (SDP) portion of the SIP message. The RTP streams do not flow directly between the two endpoints, but are rerouted so that they pass through the call logging system. This does not have to happen when a call is initiated (during the INVITE), but can happen when the call is already in progress (through a re-INVITE message). An advantage of using a re-INVITE is that ports in the system can be left free until they are needed for recording.

RTP Proxy Session Manager

When recording begins, the second component comes into play. The SER interfaces with the RTP proxy session manager, and when the session manager is notified that a new call has arrived, it opens a session for the call and supplies a unique port number to the SER. The SER then replaces the original address/port of the calling endpoint with the port number and the IP address of the logging system.

Two RTP streams can now be set up between the logging system and the two SIP endpoints instead of a single stream that directly connects the two endpoints. The application uses the IPML API to establish the streams, which appear on the virtual SCbus and can be routed together, forming a cross-connection. The streams are then available for recording to disk, using the full feature set of the Dialogic® R4 Media API. Some media features that can be used for call logging include:

- Use of a transaction record, which takes both half-duplex RTP streams and performs the necessary processing usually performed by a DSP to mix the streams before writing a single stream to disk. This eliminates the need for post-processing the two separate stream files to combine them into a single synchronized recording.
- Ability to set recording parameters on a per-recording basis, such as file format, data encoding, sampling rate, and bits per sample
- Ability to easily stream recording data into a Binary Large Object (BLOB) in a database or through a socket to a centralized recording serve
- Ability to stop recording with a single API call
- Ability to easily limit recording file size and recording time

A UNIX domain socket (First In, First Out [FIFO] inter-process message queue) connects the SER and the RTP proxy session manager. Communication between these components is through a simple, application-defined message set that is part of the RTP proxy.

Message Flow

Figure 2 illustrates the message flow for a typical recorded call in the Proxied RTP call logging system. The diagram contains numbers that correspond to the numbered paragraphs below.

1. The SER is configured to operate as a stateful proxy that relays the SIP messages generated during a call. When the SER receives an INVITE request from the source UA, it extracts its SIP call ID from the request and communicates it to the RTP proxy session manager via an inter-process socket connection. Using the ID, the session manager looks for an existing session, and if the session exists, it returns the User Datagram Protocol (UDP) port for that session. If no session is found, it creates a new session, binds it to the first empty UDP port from a range specified during configuration/startup, and returns the number of that port to the SER. After receiving a reply from the logging application, the SER replaces the original address/port of the calling endpoint with the port number and the IP address of the logging system and forwards the request as usual.

2. The SER receives a positive SIP reply (OK) from the destination UA. From the SDP portion of the message, the SER again extracts its call ID and sends it to the logging application. In this case it does not allocate a new session if none exists, but performs a lookup among existing sessions and returns either a port number if the session is found or an error indicating that there is no session with that ID. When a positive reply is received from the logging application, the SER replaces the original address/port of the calling endpoint with the port number and the IP address of the logging system and forwards the reply as usual.

3. After the session has been created, the logging application establishes two separate RTP connections between itself and the outside UAs using the IPML API. The logging application then bridges the resulting SCbus timeslots together, creating a full-duplex connection. Call recording can then be initiated using multi-timeslot transaction logging.

4. When a BYE is received from one of the UAs, the SER relays it to the other UA. On receiving an OK, the SER extracts the call ID from the second UA and communicates it to the logging application. The ID is matched with an existing recording session. Recording is terminated and the streaming in both directions is stopped. An OK is relayed to the originator of the BYE message to complete the SIP signaling.

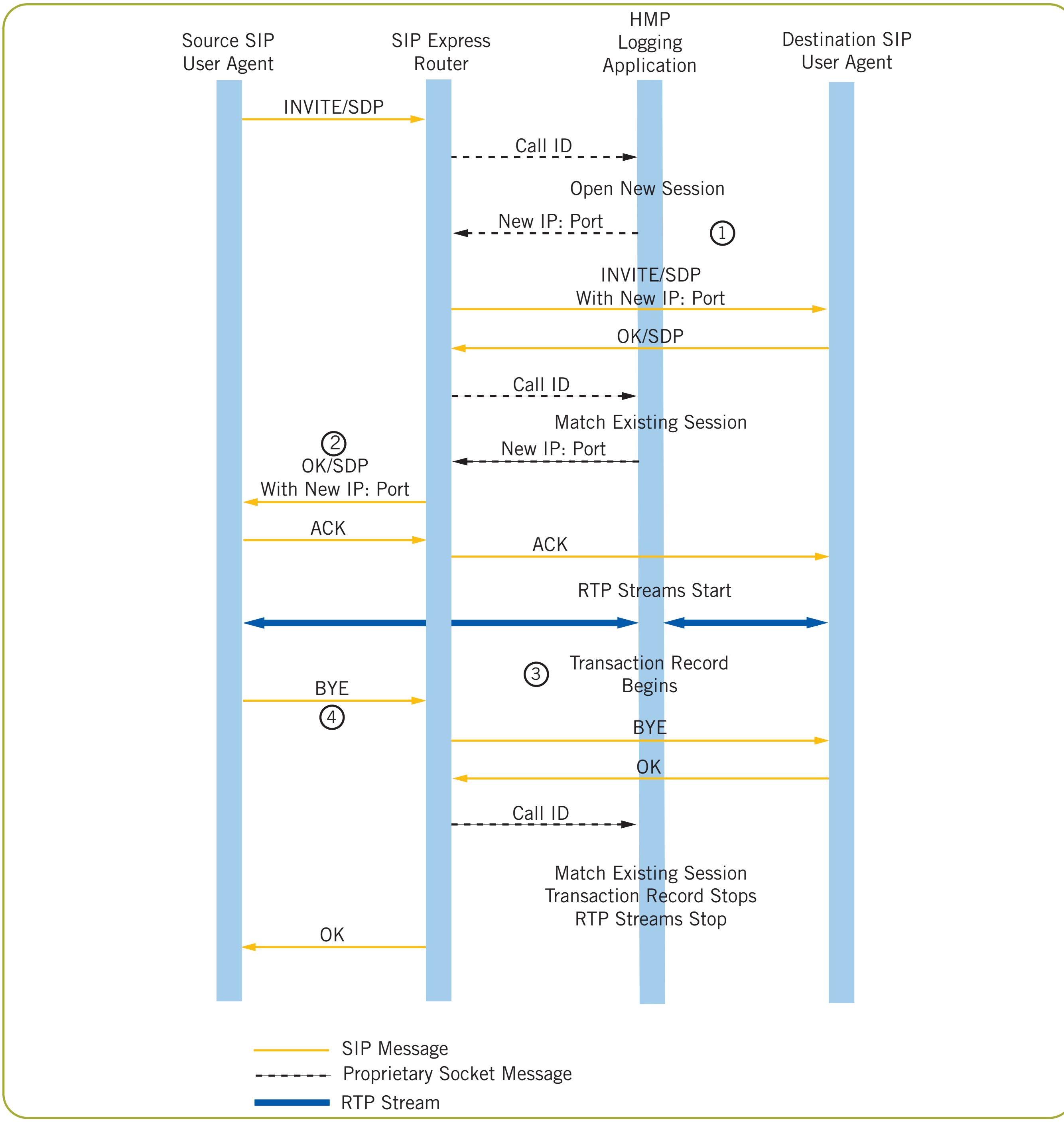

Figure 2. Message Flow for a Typical Recorded Call

Acronyms

API	Application Programming Interface
ASR	Automatic Speech Recognition
BLOB	Binary Large Object
DSP	Digital Signal Processor
FIFO	First In, First Out
HMP	Host Media Processing
IP	Internet Protocol
IPML	IP Media Library
NAT	Network Address Translation
RTP	Real-time Transport Protocol
SDP	Session Descriptor Protocol
SER	SIP Express Router
SIP	Session Initiation Protocol
TDM	Time Division Multiplexing
UA	User Agent
UDP	User Datagram Protocol

For More Information

Dialogic Host Media Processing Software Release 3.1LIN —http://www.dialogic.com/products/ip_enabled/HMP31Linux.htm

Downloads for the Dialogic Host Media Processing Software Release 3.1LIN are available at http://www.dialogic.com/support/helpweb/dxall/hmplinux/hmp31/default.htm

SIP Express Router (SER) and SIP RTP Proxy — http://www.iptel.org/ser/

The SER and SIP RTP Proxy can be downloaded at http://www.iptel.org/download#ser_stable

To learn more, visit our site on the World Wide Web at **http://www.dialogic.com.**

Dialogic Corporation
9800 Cavendish Blvd., 5th floor
Montreal, Quebec
CANADA H4M 2V9

11/07 9543-02