IAX Protocol Description

Mark Spencer Frank W. Miller

March 23, 2004

1 Introduction

The Inter-Asterisk eXchange (*IAX*) protocol provides control and transmission of streaming media over Internet Protocol (IP) networks. *IAX* can be used with any type of streaming media including video but is targeted primarily at the control of IP voice calls.

The protocol described in the document is actually Version 2 of the IAX protocol, commonly referred to as *IAX2*. *IAX2* is intended to replace the original *IAX* Version 1 protocol and as such, this document simply refers to *IAX*. It is understood that subsequent references to *IAX* refer to *IAX* Version 2.

The design goals for *IAX* derive from experience with existing Voice-over-IP (VoIP) protocols such as the Session Initiation Protocol (SIP) and the Media Gateway Control Protocol (MGCP) for control and the Real-Time Transfer Protocol (RTP) for streaming media transmission.

- Minimize bandwidth usage for both control and media transmissions with specific emphasis on individual voice calls
- Provide Network Address Translation (NAT) transparency
- Support the ability to transmit dialplan information
- Support efficient implementation of intercom and paging features

2 Quick Start

When studying a new protocol, it can be instructive to see examples of what it does before diving headlong into the gory details of message formats and

state machines. To that end, this section provides a couple of message flows for three common situations, call setup, call teardown and a nominal media flow.

2.1 Call Setup

Figure 1 illustrates the basic message flow used to setup a voice call. In this example, Host A initiates the call by sending a NEW message to Host B. Host B immediately sends back an ACCEPT message, indicating to Host A that it has received the request and is beginning to service it. Host A sends an ACK message to Host B indicating receipt of the ACCEPT message. Once Host B begins to ring the phone on its side, it sends back to Host A a RINGING message. Host A sends an ACK message back to Host B indicating receipt of the RINGING message. Finally, when the phone is picked up, Host B sends an ANSWER message to Host A and the call setup is complete. At this point full-duplex voice passes between Host A and Host B.

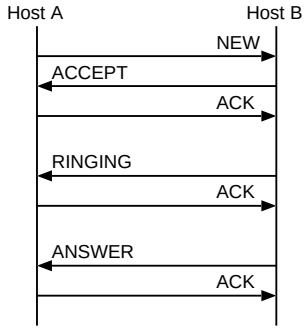

Figure 1: A simple Call Setup using the *IAX* protocol

2.2 Call Teardown

Figure 2 illustrates the message flow for a voice call teardown. In this example, Host A initiates the call teardown by sending a HANGUP message to Host B. Host B is expected to immediately send back an ACK message indicating the receipt of the teardown request and that the call has been torn down on the Host B side.

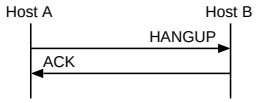

Figure 2: A simple Call Teardown using the *IAX* protocol

2.3 Media Flow

Figure 3 illustrates a simple, one-way *IAX* Media Flow. For a nominal voice call, there would be two of these flow, one flowing in either direction. Each flow is comprised mostly of *IAX* Mini Frames (labeled M in Figure 3) which contain a simple 4-byte header that targets bandwidth efficiency. The flow is supplemented by periodic Full Frames (labeled F in Figure 3) that include synchronization information.

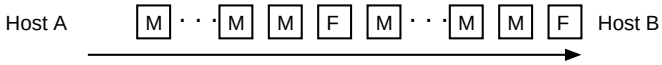

Figure 3: A simple one-way *IAX* Media Flow

Note that the Mini Frames are sent unreliably. That is, the Full Frames that are part of the stream are acknowledged by Host B but the Mini Frames are not.

3 Theory of Operation

IAX is a *peer-to-peer* media and signaling protocol. This means that the endpoints maintain state machines associated with the protocol operations. With respect to the signaling component of the *IAX* protocol design, *IAX* is more analgous to SIP than to MGCP, which is a Master-Slave call control protocol.

The basic design approach for *IAX* multiplexes signaling and multiple media streams over a single User Datagram Protocol (UDP) association

between two Internet hosts. In this facet of its design, it is actually two protocols in one, a protocol for signaling sessions and a protocol for transporting the actual media streams themselves. This approach differs from the overall architecuture of current IETF-based protocols which separate the control (MGCP and SIP) and media stream (RTP/RTCP) components using different protocols. Because signaling and media share the same UDP port number, *IAX* does not suffer from the NAT traversal problems associated with SIP.

Figure 4 illustrates the basic relationship between two Internet hosts. Each host uses the "well-known" UDP port 4569 to communicate all Internet packets. *IAX* Then uses a 15-bit Call Number to multiplex multiple streams over the UDP port number.

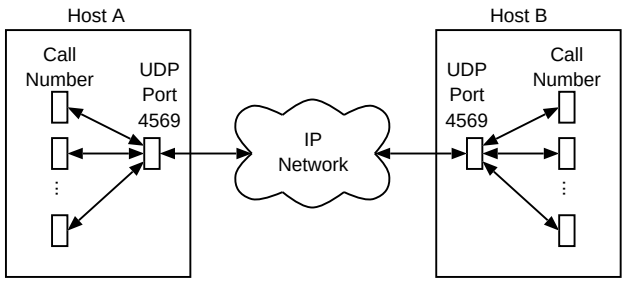

Figure 4: Multiple calls multiplexed over a single UDP port association

The value of zero is a special call number reserved on each host. When attempting to setup a call, the Call Number of the destination host is not yet known. A zero destination call number is used in this situation.

IAX is a binary protocol. This design choice was made for bandwidth efficiency. Further, the protocol is specifically optimized to make very efficient use of bandwidth for individual voice calls. The bandwidth efficiency for other stream types is sacrificed for the sake of individual voice calls.

4 Frame Definitions

IAX messages are called frames. There are several basic frame types and each of these frame types is described in detail in this section. There a

number of common fields within these frames that are explained here for consistency and brevity.

An F bit is used in indicate whether a frame is a Full Frame or not. A value of 1 in this field indicates the frame is a Full Frame and a value 0 indicates the frame is something other than a Full Frame.

A Call Number is a 15-bit unsigned integer that is used to track a media stream endpoint on a host. The value zero is a special Call Number that indicates the Call Number is unknown. A phone call actually has two Call Numbers associated with it, one for either direction.

A Timestamp can be a full 32- or an abridged 16-bit value. In the case of a 16-bit field, the value is actually the lower 16 bits of a full 32-bit timestamp that is maintained by the endpoint host.

4.1 Full Frame

A Full Frame can be used to send signaling, audio, or video information *reliably*. Full Frames are the only frame type are transmitted reliably. This means that the recipient host must return some type of message back to the sending host immediately upon reception. In some cases, the protocol may require a particular message be sent back immediately but if not the recipient must send an explicit acknowledgement. Figure 1 shows both cases. After receiving a NEW message, the recipient host must return an ACCEPT message immediately. In this case, no explicit ACK is required. Later, when a RINGING message is sent to the caller, Host A must send back an explicit ACK message since the *IAX* protocol does not require any other message to be returned at that time.

Figure 5 illistrates the binary format of a Full Frame. Table 1 describes each of the fields in Figure 5. The `R` bit is set to one if the frame is being retransmitted. Retransmission occurs after some timeout period and retransmissions are retried several times, depending on the context. The outbound stream sequence number, `OSeqno`, always begins with 0 and increases monotonically. `OSeqno` is used by the recipient to track the ordering of media frames. `ISeqno` is similar to `OSeqno`, except that it is used to track the ordering of inbound media frames. Specifically, `ISeqno` is the next expected inbound stream sequence number for the incoming media frames. `Frame Type` identifes the class of message. See Table 6 for a list of message classes. The `C` bit determines how the `Subclass` value should be interpreted. If `C` is set to 1, the `Subclass` value is interpreted as a power of two. If `C` is set to 0, `Subclass` is interpreted as a simple 7-bit unsigned integer value.

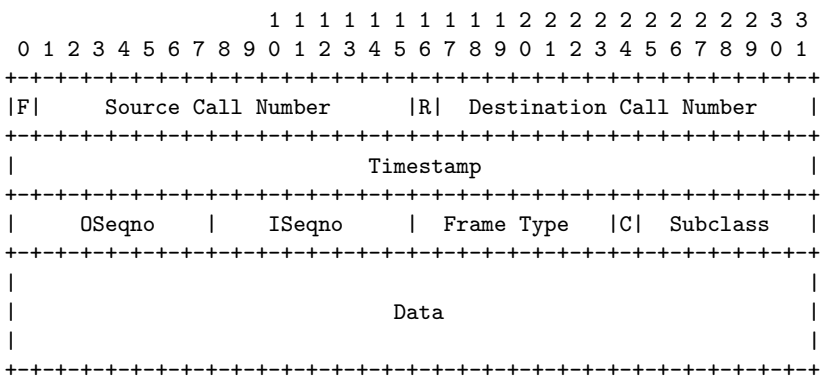

Figure 5: Full Frame binary format

4.2 Mini Frame

A Mini Frame is used to send media with a minimal protocol overhead. Figure 6 illustrates the binary format of a Mini Frame and Table 2 describes the fields present in Figure 6.

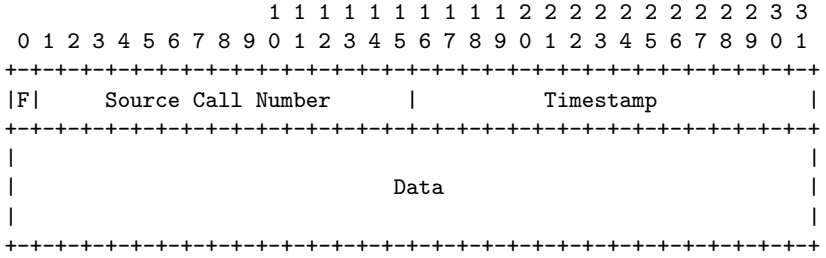

Figure 6: Mini Frame binary format

The `Timestamp` in the Mini Frame is truncated. The client generally maintains a 32-bit full timestamp. When sending Mini Frames, the low-order 16 bits of the timestamp are sent in the `Timestamp` field. When the 16-bit timestamp wraps around, a Full Frame is sent to allow the other end to synchronize its full 32-bit timestamp counter.

Table 1: Full Frame field descriptions

Field	Description
`F`	Set to the value 1 indicating that this is a Full Frame
`Source Call Number`	Call number of the transmitting side of the Full Frame
`R`	Set to the value 1 if this frame is being retransmitted and the value 0 for the initial transmission
`Destination Call Number`	Call number of the receiving side of the Full Frame
`Timestamp`	Full 32-bit timestamp
`OSeqno`	Outbound stream sequence number
`ISeqno`	Inbound stream sequence number
`Frame Type`	Frame type
`C`	Subclass value format
`Subclass`	Subclass

Table 2: Mini Frame field descriptions

Field	Description
`F`	Set to the value 0 indicating that this is not a Full Frame
`Source Call Number`	Call number of the transmitting side of the Mini Frame
`Timestamp`	16-bit timestamp

4.3 Meta Frame

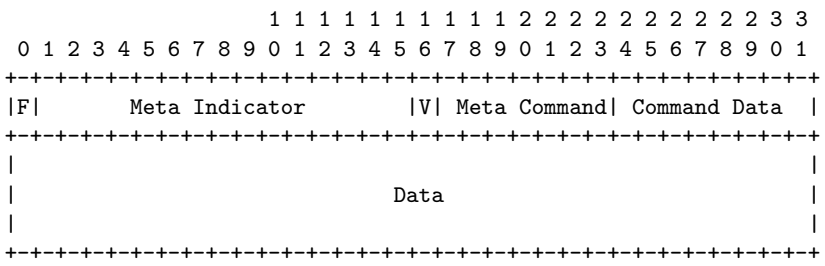

Table 3: Meta Frame field descriptions

Field	Description
F	Set to the value 0 indicating that this is not a Full Frame
Meta Indicator	
V	
Meta Command	
Command Data	

4.4 Meta Trunk Frame

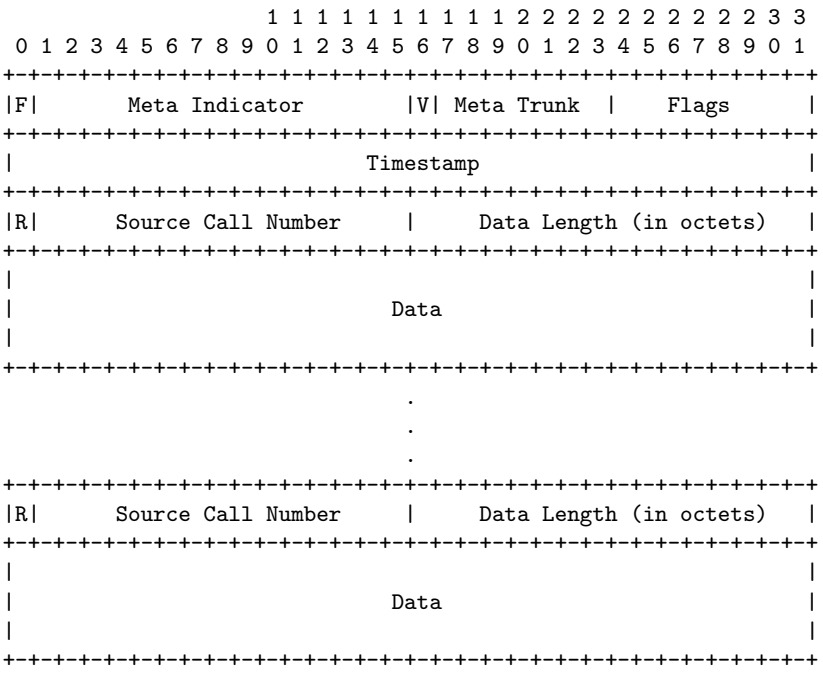

Table 4: Meta Trunk Frame field descriptions

F
: Set to the value 0 indicating that this is not a Full Frame

Meta Indicator

V

Meta Trunk

Flags

Timestamp

R

Source Call Number

Data Length

4.5 Information Element

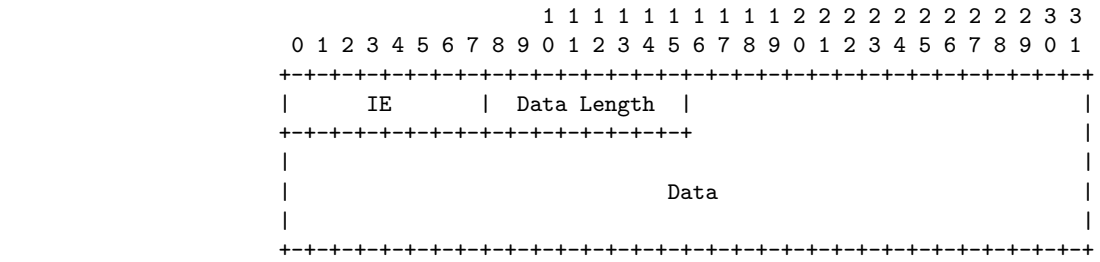

Table 5: Information Element Frame field descriptions

Information Element

Data Length

5 State Machines

This section details the various state machines that are maintained by an *IAX* endpoint.

5.1 Reliable Transmission of Full Frames

Full Frames are the only frame type that is sent *reliably*, i.e. the sender attempts to ensure that the Full Frame is indeed delivered to the intended receiver.

5.2 Ping/Pong

The Ping/Pong message exchange between two hosts is used as a heartbeat function. It provides a general timeout mechanism between two connected (or connecting) hosts.

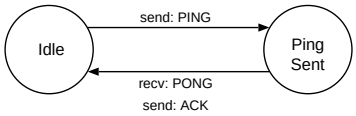

Figure 7: Ping/Pong state machine for the Pinging side

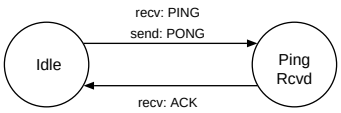

Figure 8: Ping/Pong state machine for the Ponging side

5.3 Call Setup

This section provides the state machines associated with a basic call setup between two hosts. Figure 9 provides the state machine for the initiating side of the call and Figure 10 provides the state machine for the receiving side of the call setup request.

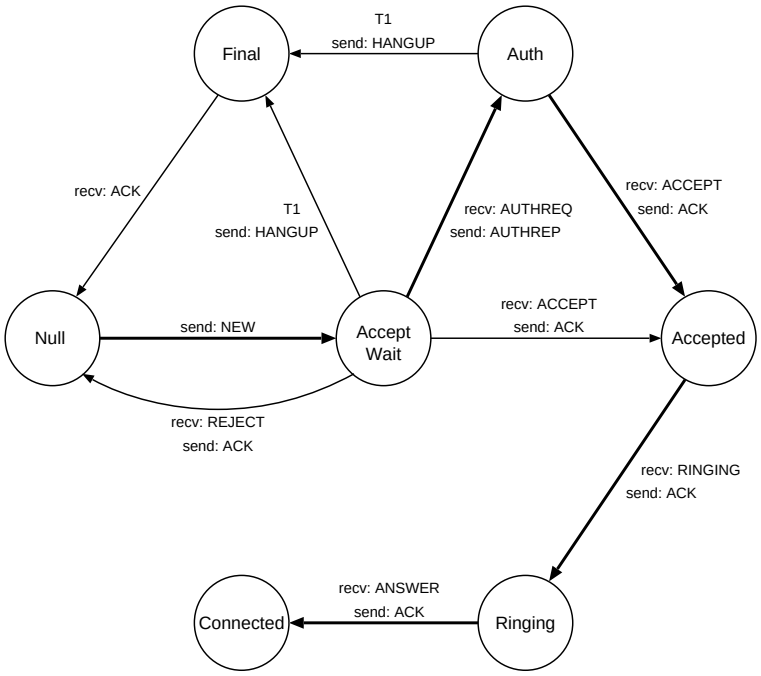

Figure 9: Call Setup state machine for the initiating side of a call

5.4 Call Teardown

This section provides the state machines associated with a basic call teardown between two hosts.

6 Example Call Flows

7 Frame Field Formats

Table 6 lists the values defined for the `Frame Type` field. This table contains the value associated with a Full Frame type, its description, a brief description of how the `Subclass` field in the Full Frame are used for the Full Frame type, and a brief description of the format of the data field. If a cell in Table 6 is left blank, the correspoding field in the Full Frame is not used.

When a Full Frame is used to transport DTMF digits, the `Subclass` contains the actual digit being transported. For voice or video streams, the `Subclass` field specifies the compression format and the data portion of the

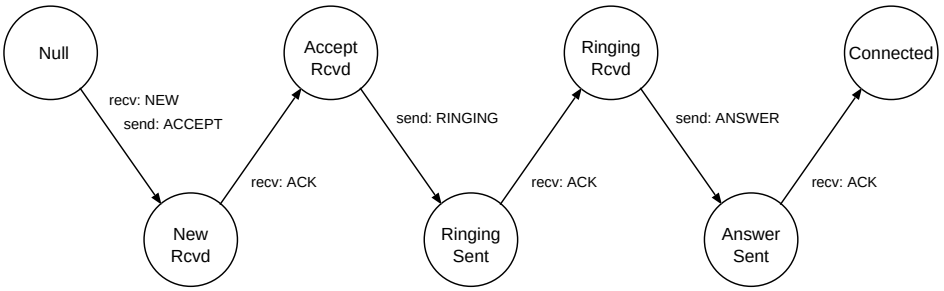

Figure 10: Call Setup state machine for the receiving side of a call

Table 6: Full Frame types

Type	Description	Subclass Description	Data Description
0x01	DTMF	0-9, A-D, , #	
0x02	Voice Data	Audio Compression Format	Raw Voice Data
0x03	Video	Video Compression Format	Raw Video Data
0x04	Control	See Control Frame Types	
0x05	Null		
0x06	*IAX* Control	See *IAX* Protocol Messages	Information Elements
0x07	Text		Raw Text
0x08	Image	Image Compression Format	Raw Image Data
0x09	HTML	See HTML Frame Types	Message Specific

Full Frame contains a packet of raw voice or video data. The compression formats for voice are given in Table 7 and for video in Table 8.

There are two types of control information that are passed between peers using Full Frames, Control Frames and *IAX* Control Frames. Control Frames provide session control, i.e. they refer to control of the devices connected to the *IAX* endpoint. *IAX* Control Frames provide *IAX* protocol specific endpoint management, i.e. they are used to manage *IAX* protocol interactions that are generally independent of the type of endpoints.

Table 7: Full Frame Audio Formats

Format	Description	Length Calculation
0x0001	G.723.1	4, 20, and 24 byte frames of 240 samples
0x0002	GSM Full Rate	33 byte chunks of 160 samples or 65 byte chunks of 320 samples
0x0004	G.711 μ-law	1 byte per sample
0x0008	G.711 a-law	1 byte per sample
0x0010	MP3 (deprecated)	
0x0020	IMA ADPCM	1 byte per 2 samples
0x0040	16-bit linear little-endian	2 bytes per sample
0x0080	LPC10	Variable size frame of 172 samples
0x0100	G.729	20 bytes chunks of 172 samples
0x0200	Speex	Variable
0x0400	ILBC	50 bytes per 240 samples

Table 8: Full Frame Image/Video Formats

Format	Description
0x00010000	JPEG
0x00020000	PNG
0x00040000	H.261
0x00080000	H.263

Table 9: *IAX* Control Frames

Control Type	Description
0x01	Hangup
0x02	Ring
0x03	Ringing (ringback)
0x04	Answer
0x05	Busy Condition
0x08	Congestion Condition
0x09	Flash Hook
0x0a	Wink
0x0b	Option
0x0c	Key Radio
0x0d	Unkey Radio
0x0e	Call Progress

Table 10: *IAX* Protocol Control Frames

Command	Mnemonic	Description
0x01	NEW	Initiate a new call
0x02	PING	Ping request
0x03	PONG	Ping reply
0x04	ACK	Acknowledgement
0x05	HANGUP	Initiate call teardown
0x06	REJECT	Reject
0x07	ACCEPT	Accepted
0x08	AUTHREQ	Authentication request
0x09	AUTHREP	Authentication reply
0x0a	INVAL	Invalid call
0x0b	LAGRQ	Lag request
0x0c	LAGRP	Lag reply
0x0d	REGREQ	Registration request
0x0e	REGAUTH	Registration authenticate
0x0f	REGACK	Registration acknowledgement
0x10	REGREJ	Registration reject
0x11	REGREL	Registration release
0x12	VNAK	Video/Voice retransmit request
0x13	DPREQ	Dialplan request
0x14	DPREP	Dialplan response
0x15	DIAL	Dial
0x16	TXREQ	Transfer request
0x17	TXCNT	Transfer connect
0x18	TXACC	Transfer accept
0x19	TXREADY	Transfer ready
0x1a	TXREL	Transfer release
0x1b	TXREJ	Transfer reject
0x1c	QUELCH	Halt audio/video transmission
0x1d	UNQUELCH	Resume audio/video transmission
0x1e	POKE	Poke request
0x1f	PAGE	Paging call description
0x20	MWI	Message waiting indication
0x21	UNSUPPORT	Unsupported message
0x22	TRANSFER	Remote transfer request

Table 11: Information Elements

Element	Description	Encoding
0x01	Called Number	String
0x02	Calling Number	String
0x03	Calling ANI	String
0x04	Calling Name	String
0x05	Called Context	String
0x06	Username	String
0x07	Password	String
0x08	Capability	32-bit unsigned integer
0x09	Format	32-bit unsigned integer
0x0a	Language	String
0x0b	Version	16-bit unsigned integer
0x0c	ADSI CPE Capability	16-bit unsigned integer
0x0d	DNID (deprecated)	
0x0e	Authentication Model	16-bit unsigned integer
0x0f	Challenge	String
0x10	MD5 Result	String
0x11	RSA Result	String
0x12	Apparent Address	Address
0x13	Refresh Interval	16-bit unsigned integer
0x14	Dialplan Entry Status	16-bit unsigned integer
0x15	Call Number	16-bit unsigned integer
0x16	Cause	String
0x17	Unknown	8-bit unsigned integer
0x18	Messages Waiting	16-bit unsigned integer
0x19	Auto-Answer	
0x1a	Request Music-on-Hold	String (optional)
0x1b	Transfer Identifier	32-bit unsigned integer
0x1c	RDNIS	String

Table 12: Authentication Methods

Method	Description
0x0001	Plaintext
0x0002	MD5
0x0004	RSA

Table 13: Dialplan Status Flags

Flag	Description
0x0001	Exists
0x0002	Cannot Exist
0x0004	Non-existant
0x4000	Retain dialtone (ignorepat)
0x8000	More digits may match number