

Freeway[®]
Client-Server
Interface Control Document

DC 900-1303C

Simpact, Inc.
9210 Sky Park Court
San Diego, CA 92123
December 1999

SIMPACT

Simpact, Inc.
9210 Sky Park Court
San Diego, CA 92123
(858) 565-1865

Freeway Client-Server Interface Control Document
© 1994 through 1999 Simpact, Inc. All rights reserved
Printed in the United States of America

This document can change without notice. Simpact, Inc. accepts no liability for any errors this document might contain.

Freeway is a registered trademark of Simpact, Inc.
All other trademarks and trade names are the properties of their respective holders.



Contents

List of Figures	5
Preface	7
1 Overview	13
1.1 Freeway Overview	14
1.2 Freeway Client-Server Environment	17
1.2.1 Defining TSI Configuration	19
1.2.2 Establishing a Freeway Connection	20
1.2.3 Exchanging Data with Freeway	20
1.2.4 Closing a Freeway Connection.	20
2 Freeway Client-Server Protocol	21
2.1 Freeway Packet Protocol.	21
2.1.1 TSI Control Packets	22
2.1.2 TSI Data Packets	26
2.1.2.1 Freeway Control Packets.	26
2.1.2.2 Freeway Data Packets	28
2.2 Example Client-Server Exchange (without Client TSI)	30
2.3 Freeway Packet Header Formats	31
2.3.1 TSI Header Format.	32
2.3.2 Freeway Header Format	35
2.3.3 ICP Header Format.	40
2.3.4 Protocol Header Format	42
3 Packet Formats	45
3.1 TSI Control Packet Formats.	45

3.1.1	Bind Request Packet	45
3.1.2	Unbind Request Packet	47
3.1.3	Unbind Force Request Packet.	47
3.1.4	ACK Packet	48
3.1.5	NAK Packet	48
3.2	Freeway Control Packet Formats	49
3.2.1	Open Session Request Packet	49
3.2.2	Open Session Response Packet	50
3.2.3	Close Session Request Packet	51
3.2.4	Close Session Response Packet	52
3.2.5	Get Version Request Packet	53
3.2.6	Get Version Response Packet	54
3.3	ICP Protocol Packet Format	55
	Index	57



List of Figures

Figure 1–1: Typical Freeway Configuration	15
Figure 1–2: Freeway Configuration with Clients That Do Not Use the DLI	18
Figure 2–1: TSI Connection Data Flow (Without Arbitration)	25
Figure 2–2: TSI Connection Data Flow (With Arbitration)	25
Figure 2–3: Freeway Session Initiation, Data Transfer, and Termination	29
Figure 2–4: Typical ICP Protocol Session	30



Preface

Purpose of Document

This document describes the packets that are exchanged between a client and a Freeway server. It aids the client application developer in the construction of the data packets necessary to open and close sessions to the Freeway server and communicate with the ICP protocol logic. It is intended for use in the development of client applications that choose not to use the Data Link Interface (DLI) provided by Simpact. Developers who may or may not choose to use the Transport Subsystem Interface (TSI) provided by Simpact also need to use this document to establish communications to the ICP protocol logic.

The information in this document supplements that in the programmer's guide for your particular protocol software running on the Freeway ICP. Users of the TSI provided by Simpact also need to reference the TSI Reference Guide in conjunction with this document. All developers should become familiar with the *Freeway User Guide* sections on server configuration of its TSI.

Intended Audience

This document should be read by application programmers. You should understand the concepts of the client-server interface and be familiar with the C programming language. You should also be familiar with your particular protocol programmer's guide, the *Freeway Transport Subsystem Interface Reference Guide*, and the *Freeway User Guide* (see the "Simpact References" section below).

Required Equipment

A Freeway 1100/1150/1200/1300, Freeway 2000/4000, or Freeway 8800 server.

Organization of Document

[Chapter 1](#) is an overview of Freeway and its client/server interface. It also describes the relationship of the various layers (application, DLI, TSI).

[Chapter 2](#) describes the various packet exchanges necessary to initiate and maintain communication with the Freeway server and its protocol logic. Included is a series of data flow charts summarizing the basic data flows. This chapter defines all header formats and valid field ranges for packets.

[Chapter 3](#) outlines the exact format of each packet exchanged between the client and the Freeway server. It is to be used as a reference for constructing individual packets.

Simpact References

The following general product documentation list is to familiarize you with the available Simpact Freeway and embedded ICP products. The applicable product-specific reference documents are mentioned throughout each document (also refer to the “readme” file shipped with each product). Most documents are available on-line at Simpact’s web site, www.simpact.com.

General Product Overviews

- *Freeway 1100 Technical Overview* 25-000-0419
- *Freeway 2000/4000/8800 Technical Overview* 25-000-0374
- *ICP2432 Technical Overview* 25-000-0420
- *ICP6000X Technical Overview* 25-000-0522

Hardware Support

- *Freeway 1100/1150 Hardware Installation Guide* DC-900-1370
- *Freeway 1200/1300 Hardware Installation Guide* DC-900-1537

-
- *Freeway 2000/4000 Hardware Installation Guide* DC-900-1331
 - *Freeway 8800 Hardware Installation Guide* DC-900-1553
 - *Freeway ICP6000R/ICP6000X Hardware Description* DC-900-1020
 - *ICP6000(X)/ICP9000(X) Hardware Description and Theory of Operation* DC-900-0408
 - *ICP2424 Hardware Description and Theory of Operation* DC-900-1328
 - *ICP2432 Hardware Description and Theory of Operation* DC-900-1501
 - *ICP2432 Electrical Interfaces (Addendum to DC-900-1501)* DC-900-1566
 - *ICP2432 Hardware Installation Guide* DC-900-1502

Freeway Software Installation and Configuration Support

- *Freeway Message Switch User Guide* DC-900-1588
- *Freeway Release Addendum: Client Platforms* DC-900-1555
- *Freeway User Guide* DC-900-1333
- *Freeway Loopback Test Procedures* DC-900-1533

Embedded ICP Software Installation and Programming Support

- *ICP2432 User Guide for Digital UNIX* DC-900-1513
- *ICP2432 User Guide for OpenVMS Alpha* DC-900-1511
- *ICP2432 User Guide for OpenVMS Alpha (DLITE Interface)* DC-900-1516
- *ICP2432 User Guide for Solaris STREAMS* DC-900-1512
- *ICP2432 User Guide for Windows NT* DC-900-1510
- *ICP2432 User Guide for Windows NT (DLITE Interface)* DC-900-1514

Application Program Interface (API) Programming Support

- *Freeway Data Link Interface Reference Guide* DC-900-1385
- *Freeway Transport Subsystem Interface Reference Guide* DC-900-1386
- *QIO/SQIO API Reference Guide* DC-900-1355

Socket Interface Programming Support

- *Freeway Client-Server Interface Control Document* DC-900-1303

Toolkit Programming Support

- *Freeway Server-Resident Application and Server Toolkit Programmer Guide* DC-900-1325
- *OS/Impact Programmer Guide* DC-900-1030
- *Protocol Software Toolkit Programmer Guide* DC-900-1338

Protocol Support

- *ADCCP NRM Programmer Guide* DC-900-1317
- *Asynchronous Wire Service (AWS) Programmer Guide* DC-900-1324
- *AUTODIN Programmer Guide* DC-908-1558
- *Bit-Stream Protocol Programmer Guide* DC-900-1574
- *BSC Programmer Guide* DC-900-1340
- *BSCDEMO User Guide* DC-900-1349
- *BSCTRAN Programmer Guide* DC-900-1406
- *DDCMP Programmer Guide* DC-900-1343
- *FMP Programmer Guide* DC-900-1339
- *Military/Government Protocols Programmer Guide* DC-900-1602
- *N/SP-STD-1200B Programmer Guide* DC-908-1359
- *SIO STD-1300 Programmer Guide* DC-908-1559
- *X.25 Call Service API Guide* DC-900-1392
- *X.25/HDLC Configuration Guide* DC-900-1345
- *X.25 Low-Level Interface* DC-900-1307

Document Conventions

This document follows the most significant byte first (MSB) and most significant word first (MSW) conventions for bit-numbering and byte-ordering. In all packet transfers between the client applications and the ICPs, the ordering of the byte stream is preserved. Processes on the client computer are never required to perform byte stream manipulation regardless of the hardware configuration.

The term “Freeway” refers to any of the Freeway server models (for example, Freeway 1100/1150/1200/1300, Freeway 2000/4000, or Freeway 8800).

Physical “ports” on the ICPs are logically referred to as “links.” However, since port and link numbers are usually identical (that is, port 0 is the same as link 0), this document uses the term “link.”

Program code samples are written in the “C” programming language.

Revision History

The revision history of the *Freeway Client-Server Interface Control Document*, Simpact document DC 900-1303C, is recorded below:

Revision	Release Date	Description
DC 900-1303A	December 1994	Original release
DC 900-1303B	April 1999	Correct the Freeway Header sizes (Section 2.3.2 and Chapter 3). Unbind Force (page 23) receives no response.
DC 900-1303C	December 1999	Update references, overview, and area code

Customer Support

If you are having trouble with any Simpact product, call us at 1-800-275-3889 Monday through Friday between 8 a.m. and 5 p.m. Pacific time.

You can also fax your questions to us at (858) 560-2838 or (858) 560-2837 any time. Please include a cover sheet addressed to “Customer Service.”

We are always interested in suggestions for improving our products. You can use the report form in the back of this manual to send us your recommendations.

This document describes Simpack's client/server interface to the Freeway communications server. Its primary purpose is to serve as a reference tool when you are writing client applications that do not use the data link interface (DLI).

Note

Two types of client applications are discussed in this document: ones that use the transport subsystem interface (TSI) and ones that do not. In either case, it is assumed that the applications do not use the data link interface (DLI). That is, applications that use the DLI are not discussed.

If your client applications use the TSI, you need to use this document to determine how each packet is constructed when communicating with the TSI. If your client applications do *not* use the TSI, you need to use this document to determine the sequence and construction of every exchange between the Freeway server and client.

A TSI layer resides on both the Freeway server and client. If your client applications do not use the TSI, they must perform the work that is normally done by the client TSI layer, including all buffer management. However, bypassing the TSI gives the applications complete control of how data is written and read once sessions have been established with the server.

The TSI and DLI provide a consistent, high-level, common interface across multiple hosts, operating systems, and transport services. Note the following similarities and differences between the DLI and TSI:

- The DLI provides *session*-oriented data services to your client application with a subroutine library. This library contains functions that permit your application to access, configure, establish and terminate *sessions*, and transfer data across multiple data link protocols. The DLI uses the TSI for its transport services.
- The TSI provides *connection*-oriented data services to your client application with a subroutine library. This library contains functions that permit your application to access, configure, establish and terminate *connections*, and exchange data with a TSI peer application (normally the Freeway server's TSI).

1.1 Freeway Overview

Simpact's Freeway communications servers enable client systems on a local-area network (LAN) to access specialized wide-area networks (WANs). The Freeway server is user programmable and communicates in real time. It provides multiple data links and a variety of network services to LAN-based clients in financial, defense, telecommunications, and process-control applications. [Figure 1-1](#) shows a Freeway configuration.

Note that all of the clients shown in the figure use the DLI, which is the usual case. This document, however, focuses on clients that do not use the DLI. These types of clients are shown later in [Figure 1-2 on page 18](#).

To maintain high data throughput, Freeway uses a multi-processor architecture to support the LAN and WAN services. The LAN interface is managed by a single-board computer, called the server processor. It uses the commercially available VxWorks operating system to provide a full-featured base for the LAN interface and layered services needed by Freeway.

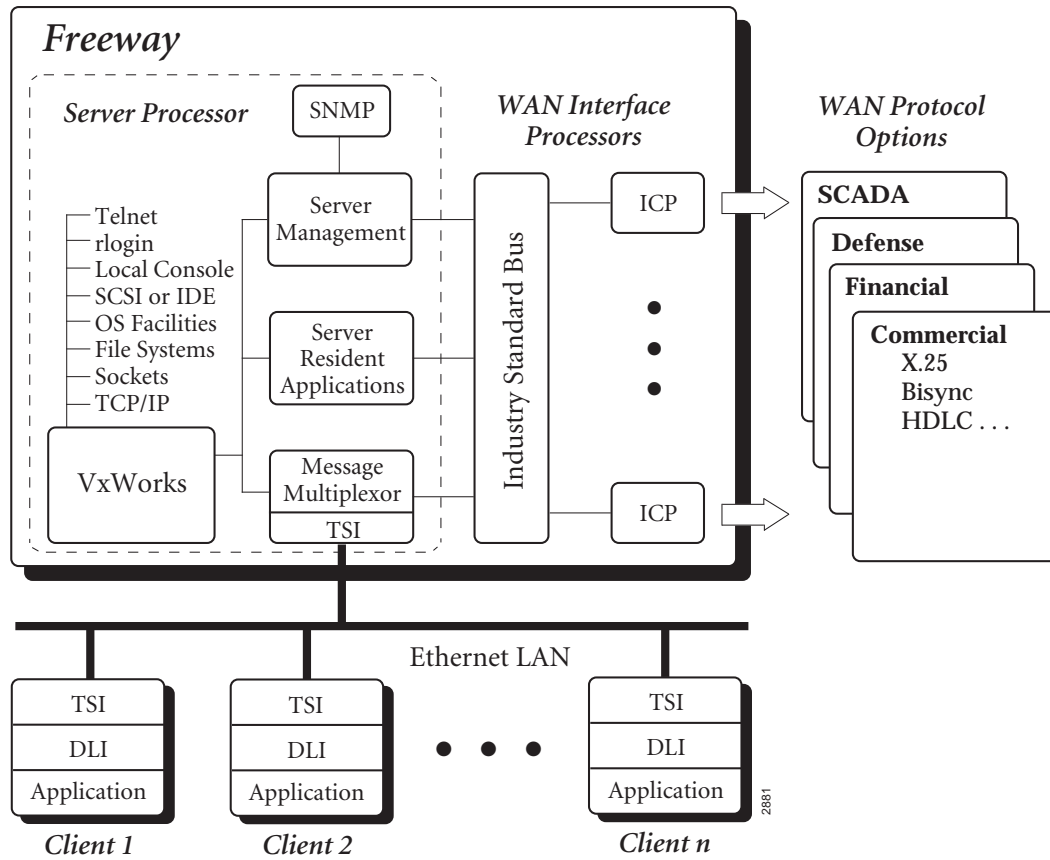


Figure 1-1: Typical Freeway Configuration

Freeway can be configured with multiple WAN interface processor boards, which are Simpack intelligent communications processors (ICPs). Each ICP runs the communication protocol software using Simpack's OS/Impact real-time operating system.

Other Freeway features include:

- Freeway server standard support for Ethernet and Fast Ethernet LANs running the transmission control protocol/internet protocol (TCP/IP)
- Freeway server standard support for FDDI LANs running the transmission control protocol/ internet protocol (TCP/IP)
- Support for multiple ICPs (four, eight, or sixteen communication lines per ICP)
- Wide selection of electrical interfaces including EIA-232, EIA-449, EIA-530, and V.35
- Server management and performance monitoring with the simple network management protocol (SNMP), as well as interactive menus available through a local console, telnet, or rlogin
- Variety of off-the-shelf communication protocols available from Simpack which are independent of the client operating system and hardware platform
- Support for multiple WAN communication protocols simultaneously on a single Freeway server
- Elimination of difficult LAN and WAN programming and systems integration by providing a data link interface (DLI) to access Freeway services through the transport subsystem interface (TSI)
- Creation of customized server-resident and ICP-resident software, using Simpack's software development toolkits

1.2 Freeway Client-Server Environment

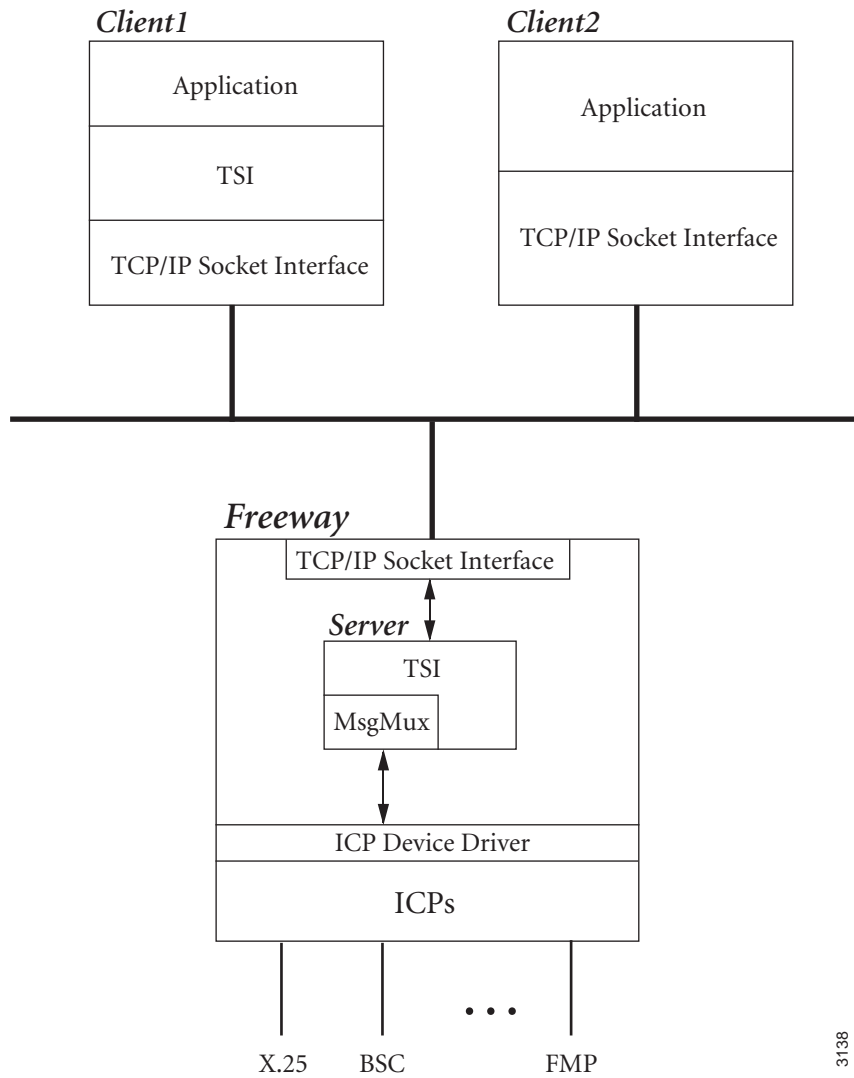
Freeway acts as a gateway that connects a client on a local-area network to a wide-area network. Through Freeway, a client application can exchange data with a remote data link application. Your client application must interact with the Freeway server and its resident ICPs before exchanging data with the remote data link application.

One of the major Freeway components is the message multiplexor (MsgMux) that manages the data traffic between the LAN and the WAN environments. The client application typically interacts with the Freeway MsgMux through a TCP/IP BSD-style socket interface (or a shared-memory interface if it is a server-resident application (SRA)). The ICPs interact with the MsgMux through the shared-memory interface of the industry-standard bus to exchange WAN data.

This document is intended to support clients that use a TCP/IP BSD-style socket interface only. Server-resident applications that use a shared-memory interface are beyond the scope of this document.

[Figure 1–2](#) shows a Freeway connected to locally attached clients by a TCP/IP network across an Ethernet LAN interface. Note the following about the clients in this figure:

- Both clients do not use the DLI.
- Client1 uses the client TSI layer. If this is the case with your client applications, you will only need to reference this document to determine packet header formats and field values. Your primary reference for TSI information will be the *Freeway Transport Subsystem Interface Reference Guide*.
- Client2 does not use the client TSI layer. If this is the case with your client applications, you will need to use this document as your primary development tool.



3138

Figure 1-2: Freeway Configuration with Clients That Do Not Use the DLI

Note

Message formats and protocols used between the Freeway server and its client applications are subject to change in future releases. If you write applications without DLI or TSI layers, you may not be able to migrate the applications easily when the message formats and protocols change. If you write applications that access the TSI layer directly, use caution in sizing buffers submitted to the TSI; refer to the “Buffer Management” section in the *Freeway Transport Subsystem Interface Reference Guide*.

The Freeway trace feature is a valuable debugging tool for client applications. This facility captures each packet transferred over the socket interface at the TSI level, the DLI level, or both levels. See the *Freeway Transport Subsystem Interface Reference Guide* and the *Freeway User Guide* for information on how to configure your client and server for tracing. An appendix in the *Freeway Transport Subsystem Interface Reference Guide* shows an example of trace output.

1.2.1 Defining TSI Configuration

After you install the Freeway software as described in the *Freeway User Guide*, you must configure the TSI:

- Use the server TSI configuration file to configure the Freeway server TSI layer. You must do this regardless of whether or not you use the client TSI layer. The server TSI configuration file is described in the “Customizing System Configuration” chapter in the *Freeway User Guide*.
- If you use the client TSI layer, you must use the client TSI configuration file to define parameters for the TSI connections between your client application and Freeway. The client TSI configuration file is described in the “TSI Configuration” chapter in the *Freeway Transport Subsystem Interface Reference Guide*

1.2.2 Establishing a Freeway Connection

After the TSI configuration is defined, the client application must establish a connection to the Freeway server. Applications that use the client TSI layer only need to make a library call (`tConnect`). Applications that do not use the client TSI layer must exchange “bind” and “open session” packets with the Freeway server TSI layer and `MsgMux`, respectively, as well as connect to the BSD socket.

1.2.3 Exchanging Data with Freeway

After the connection is established, the client application can exchange data packets with the server. Applications that use the client TSI layer only need to make library calls (`tRead`, `tWrite`). Applications that do not use the client TSI layer must submit I/O requests directly to the BSD socket.

1.2.4 Closing a Freeway Connection

When the data transfer is complete, the client application can close the connection. Applications that use the client TSI layer only need to make a library call (`tDisconnect`). Applications that do not use the client TSI layer must exchange “close session” and “unbind” packets with `MsgMux` and the server TSI layer, respectively, and then close the BSD socket.

Freeway Client-Server Protocol

The Freeway client and server communicate via TCP/IP BSD-style sockets. (The server-resident client application that communicates via the shared-memory interface is not discussed in this document). The Freeway server acts as a TCP/IP socket server. It issues a `listen()` call for accepting incoming connections from the Freeway client. The Freeway client then issues the `connect()` call to request a connection to the Freeway server. The socket library calls needed for socket connection, disconnection, and data transfer are beyond the scope of this document and it is assumed that you are familiar with them. You may want to refer to the user guides or man pages for your operating environment. This chapter discusses the format of data packets that are passed between the Freeway client and the Freeway server, and the data packet sequence necessary to establish and maintain Freeway sessions.

2.1 Freeway Packet Protocol

If are using the TSI, you do not need to read “TSI Control Packets” ([Section 2.1.1](#)), but you do need to read “TSI Data Packets” ([Section 2.1.2](#)) to familiarize yourself with the packets and their header structures. If you are *not* using the TSI, you should read both of these sections because you need a detailed knowledge of the sequence of packet exchanges between the client and server.

Use a text editor to examine the contents of the server TSI configuration file and obtain the number of the well-known port on which the Freeway server is listening. The client application uses this port number when requesting connection to the server. For a

description of the well-known port parameter, see the “Customizing System Installation” chapter in the *Freeway User Guide*.

The Freeway server uses the TSI logic and so expects the TSI-level packet exchanges that occur between a client and the server. All packets exchanged between the client and the server contain a TSI header, which indicates whether the packet is control information or data. Control packets coordinate communication between the client and the server TSI level. Data packets are processed at the MsgMux level for session control and ICP data transfer.

The Freeway server design supports multiple sessions per connection. The client application can open and maintain a number of Freeway sessions on one socket connection, or open one session per connection. For client applications using the TSI, note that the client TSI configuration determines buffer size and resource management handled by TSI. Therefore, you should take care when configuring both the client and server TSI.

2.1.1 TSI Control Packets

A TSI control packet is a communication between a client and a server on the TSI level to coordinate the use of the socket connection. TSI control packets are used after the connection has been made to control the use of the socket. The packets are also used just before the socket is disconnected to coordinate shutdown. A TSI control packet consists of a TSI Header, and optionally, a TSI data area. The following sections describe TSI control packets and provide examples of packet data flow.

Bind

This is the first packet that a client sends to the Freeway server after the socket connection has been established. The bind packet announces the communication capabilities of the sender to the remote TSI application (this information is contained in the TSI data area). Based on its TSI configuration, the Freeway server can accept the bind request by responding with an ACK packet, or it can reject it by responding with its own Bind request packet. If the client receives an ACK packet from the server, the exchange

of TSI data packets can begin immediately. [Figure 2–1 on page 25](#) shows an example of this type of data flow.

If the client receives a Bind request back from the server, it can accept the server communication conditions in the Bind request by sending an ACK packet to the server. When the server receives the ACK packet, it is ready to accept the first data packet. [Figure 2–2 on page 25](#) shows an example of this type of data flow.

If the client cannot accept the communication conditions in the Bind request from the server, it sends a NAK packet to the server, followed by an Unbind control packet to initiate termination of the connection.

Unbind

When a TSI application wants to terminate the socket connection gracefully, it sends the remote TSI application an Unbind control packet. When the Unbind request is received, the TSI application checks its internal I/O queues, and if they are not empty, it sends a NAK control packet; otherwise, it sends an ACK control packet. Client applications should not use the Unbind packet, but should use the Unbind Force packet instead.

Unbind Force

When a client application wants to unconditionally terminate a socket connection, without regard to completing active I/O requests, it sends the server an Unbind Force control packet. When it receives this packet, the server TSI removes all I/O requests from its internal I/O queues for the connection and frees the I/O requests by using the `tBufFree` function. The server TSI then closes the socket on the server. No response is issued to the client. The client is responsible for closing the socket on the client. The client application should use the Unbind Force control packet if it wants to close a connection with the server. [Figure 2–1](#) and [Figure 2–2](#) shows examples of how an Unbind Force control packet is used.

Echo request

Not implemented at this time.

Echo reply

Not implemented at this time.

ACK

The ACK control packet indicates acceptance of a Bind or Unbind request packet from a partner TSI application.

NAK

The NAK control packet indicates rejection of a Bind or Unbind request packet from a partner TSI application.

However, note that a TSI application must always accept an Unbind Force request. That is, it must respond with an ACK, not a NAK.

ACK Data

Not implemented at this time.

NAK Data

Not implemented at this time.

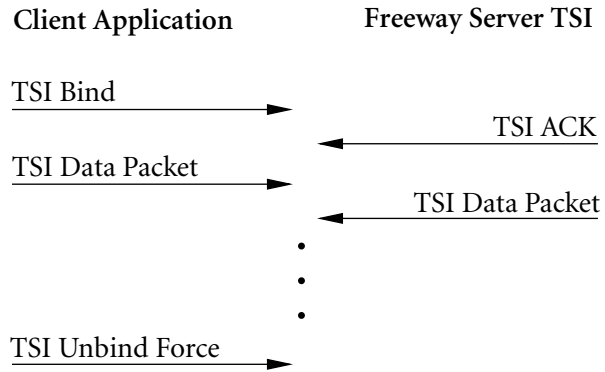


Figure 2–1: TSI Connection Data Flow (Without Arbitration)

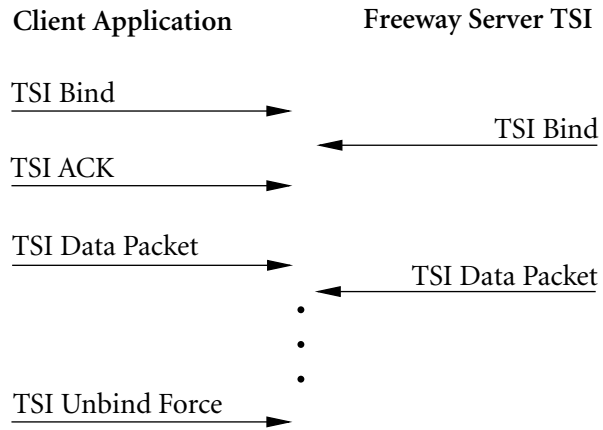


Figure 2–2: TSI Connection Data Flow (With Arbitration)

2.1.2 TSI Data Packets

A TSI data packet is a communication between a client and a server that is passed to or from the next higher level of processing (via `tRead` and `tWrite` calls when using TSI). Data packets are used after the two TSI applications have “bound” and are ready to exchange packets that control Freeway sessions and transfer data to and from the ICP protocol logic. TSI data packets received by the Freeway server are passed to the Freeway `MsgMux` task via `tRead` calls. TSI data packets are sent by the Freeway server via the `tWrite` call. Each TSI data packet contains a TSI header, a Freeway header, and optionally, an ICP header, Protocol header, and any ICP data.

After an ACK control packet has been sent in response to a Bind request, the client and server are ready to exchange data. There are two types of TSI data packets:

- Freeway control packets for opening and closing Freeway sessions (and, in the future, for session or `MsgMux` control)
- Freeway data packets for passing data to and from the Freeway ICPs

The following sections describe these two types of data packets.

2.1.2.1 Freeway Control Packets

Freeway control packet requests are issued by the client. Freeway control packet responses are issued by the Freeway server. The server validates the request. If the packet is invalid or the server cannot honor the request (for example, because of lack of resources), the server sends a response indicating the reason the request cannot be honored. If the packet is valid, the server sends a response indicating acceptance of the packet. These packets consist of a TSI header and a Freeway header, and optionally, data immediately following the Freeway header. The following sections describe each Freeway control packet. [Figure 2–3 on page 29](#) shows an example of packet data flow.

Open Session

This is the first TSI data packet that a client should send for a new Freeway session. This packet requests the Freeway server to open a session to a specific ICP, and specifies a client session ID to associate with this session. The ICP is specified as a string and must match one of the device names specified in the Freeway server boot configuration file (see the *Freeway User Guide*). If the request is valid, the Freeway server reports the Freeway-assigned session ID and the Freeway software version in the Open Session Response packet. Otherwise, the response contains an error code describing the reason for rejection.

Open Session Response

This packet is sent in response to the Open Session control packet. It reports acceptance or rejection of the request, the Freeway-assigned session ID, and the Freeway software version.

Close Session

This is the last TSI data packet that a client should send for an open Freeway session. This control packet requests the Freeway server to close an open session to an ICP; it specifies the Freeway-assigned session ID and the client session ID sent to the server in the Open Session request. The server reports acceptance or rejection of the request in the Close Session Response packet.

Close Session Response

This packet is sent in response to the Close Session control packet. It reports acceptance or rejection of the request.

Set Time

Not implemented at this time.

Get Time

Not implemented at this time.

Get Version

This packet requests the Freeway software version string from the server.

Get Version Response

This packet reports the Freeway software version string in the data area. A typical string returned is Freeway Release 2.1.

2.1.2.2 Freeway Data Packets

Either the client or server can issue Freeway data packets. These packets contain ICP data, that is, data buffers transferred between the ICP and the Freeway server MsgMux task. These packets may or may not contain data transferred over the WAN link. At minimum, a data packet consists of a TSI header, Freeway header, ICP header, and Protocol header; ICP data, if any, immediately follows the Protocol header. The following sections describe each Freeway data packet. [Figure 2-3](#) shows an example of packet data flow.

Data Packet to ICP

After the client has opened a session (using the Open Session Request), this packet may be sent to the server; it must contain the Freeway session ID in the Freeway header. If this packet is invalid, the server will send it back to the client; the status field will indicate the reason for rejection. If this packet passes validation, it is forwarded to the ICP; no response is sent to the client.

Expedite Data Packet to ICP

This packet is identical to the Data Packet to ICP, except it requests an expedited delivery to the ICP. This enables the client to send packets to the ICP that will be processed by the ICP before other data packets queued normally using the Data Packet to ICP.

Data Packet From ICP

This packet contains data received by MsgMux from the ICP. The server reports the client session ID and the Freeway session ID in the Freeway header. Depending on the ICP protocol in use, the server may receive these packets from the ICP immediately following the successful processing of a client Open Session request packet.

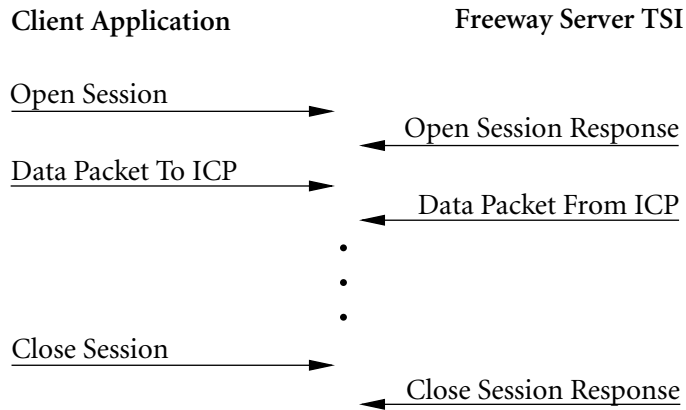


Figure 2–3: Freeway Session Initiation, Data Transfer, and Termination

2.2 Example Client-Server Exchange (without Client TSI)

Figure 2-4 shows an example client-server data exchange for a client application that bypasses the TSI and performs reads and writes directly through the TCP/IP BSD socket. The example consolidates the data flows shown earlier to illustrate the complete flow for a single Freeway session using an ICP protocol. Before the exchange, the client sends socket library calls to open the socket connection. Afterwards, it sends calls to close the connection.

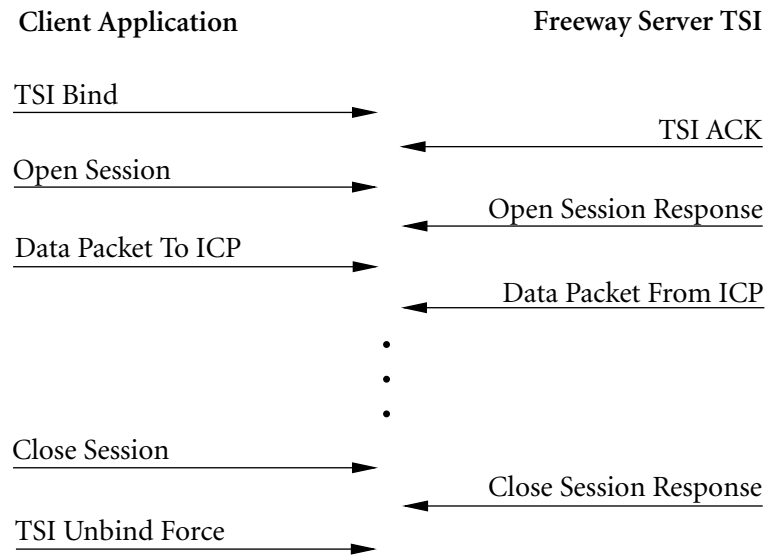


Figure 2-4: Typical ICP Protocol Session

2.3 Freeway Packet Header Formats

Several headers precede ICP data in packets exchanged between the client and the server. This design supports the DLI and TSI. The design also includes headers used by protocol client applications and the ICP-resident protocol logic. Note that applications that use the TSI do not include a pointer to the TSI header in buffers passed to the TSI `tRead` or `tWrite` routine. The TSI builds this header in the storage area immediately preceding the buffer pointer passed in the `tRead` or `tWrite` call. (Buffer management is described in the *Freeway Transport Subsystem Interface Reference Guide*).

The following sections describe the header formats.

2.3.1 TSI Header Format

The TSI header precedes all data sent over the BSD socket. If the client application is using the TSI, it does not need to construct this header; the TSI library routine will construct it and add it to the beginning of the buffer passed to the `tWrite` routine. Client applications not using the TSI must build this header for every packet sent to the server via the BSD socket. The TSI header must be in network byte order (big-endian).

TSI Header

Header Length	4 bytes
Data Length	4 bytes
Sequence Number	4 bytes
Packet Type	2 bytes
Command	2 bytes
First Segment	2 bytes
Last Segment	2 bytes

The fields in the TSI header are as follows:

Header Length (4 bytes)

This field indicates the length of the TSI header from the Header Length field to the Last Segment field, inclusive. Normally, this field contains 20 (decimal). However, Simpact TSI software is subject to change.

Data Length (4 bytes)

This field indicates the length of the data area following the TSI header. The sum of the header length and the data length is the total length of a TSI packet. The valid range of this field is 1 to the `maxBufSize` parameter specified in the TSI configuration file.

Sequence Number (4 bytes)

This field contains a sequential number that is produced by the sender of the packet. This value is an unsigned integer. This field is only unique within one TSI application in one direction; no validation is performed by TSI on this field.

Packet Type (2 bytes)

This field indicates the type of packet. There are two distinct types of packets under the TSI. One is the control packet, and the other is the data packet. If the control packet is used, the Command field is significant. Otherwise, the Command field is ignored. Valid packet types are:

Control packet (1)

A control packet is a packet that is received, validated, and processed by the TSI. A control packet always produces a control response packet unless it is an acknowledge packet (ACK or NAK). No control packet is passed on to the application. Some control packets may have additional data in the data area.

Data packet (2)

A data packet is a packet that is received and validated by the TSI. The data packet is subsequently delivered to the TSI user unless the connection no longer exists.

Command (2 bytes)

This field is only significant if the packet is a control packet. Valid commands are:

Bind (1)

The Data Length field should indicate the size of the Bind request. Currently the data field is 10 bytes.

Unbind (2)

There is no data area accompanying this packet and the Data Length field must be 0.

Unbind force (3)

There is no data area accompanying this packet and the Data Length field must be 0.

Echo request (4)

Not implemented at this time.

Echo reply (5)

Not implemented at this time.

ACK (6)

There is no data area accompanying this packet and the Data Length field must be 0.

NAK (7)

There is no data area accompanying this packet and the Data Length field must be 0.

ACK data (8)

Not implemented at this time.

NAK Data (9)

Not implemented at this time.

First Segment (2 bytes)

This field is not used.

Last Segment (2 bytes)

This field is not used.

2.3.2 Freeway Header Format

The Freeway header immediately follows the TSI header. All TSI Data packets must contain a Freeway header; these packets include Freeway control packets (Open Session and Close Session) and Freeway data packets (ICP protocol data).

The Freeway header contains 44 bytes and must be in network byte order (big-endian).

Freeway Header

Reserved	8 bytes
Shared Header Size	2 bytes
Private Header Size	2 bytes
Type	2 bytes
Command	2 bytes
Status	2 bytes
Client Session ID	2 bytes
Freeway Session ID	2 bytes
Reserved	2 bytes
Private	20 bytes

The fields in the Freeway header are as follows:

Reserved (8 bytes)

This field is a reserved area for use by the Freeway server only.

Shared Header Size (2 bytes)

This field indicates the size of the actual header from the Type field to the Reserved field, inclusive. Normally, this field contains 12 (decimal). However, Freeway server code is subject to change.

Private Header Size (2 bytes)

This field indicates the size of the second reserved area in the header (called “Private” in the above Freeway Header diagram). Normally, this field contains 20 (decimal). However, Freeway server code is subject to change.

Type (2 bytes)

This field is equivalent to the usFWType field in the DLI_OPT_ARGS structure. (See the *Freeway Data Link Interface Reference Guide* for a detailed description of the DLI_OPT_ARGS structure.) The Type field indicates the type of packet. There are two valid types:

Control (1)

This type is for session control requests/responses and all other packets that will not be sent to the ICP.

Data (2)

This type is for packets that are transferred to and from the ICP.

Command/Response (2 bytes)

This field is equivalent to the usFWCommand in the DLI_OPT_ARGS structure. The purpose of the Command field is to request specific services from the Freeway server. If the packet is a control packet, the Command field may be one of the following:

Open session request or response (1)

Close session request or response (2)

Set time request (3)

Get time request (4)

Get version request or response (5)

If the packet is a data packet, the Command field may be one of the following:

Write to ICP request (1)

Expedite write to ICP request (2)

Data packet from ICP (3)

Status (2 bytes)

This field is equivalent to the `usFWStatus` field in the `DLI_OPT_ARGS` structure. This field is ignored by the Freeway server if the packet is from the client application. This field contains specific information about the cause of a failed request packet that originated from the client application. This field contains valid information only if the packet is from the Freeway server. When receiving a packet from the Freeway server, the client application should check this field for one of the following values:

OK (0)

The request packet was processed successfully by the Freeway server or the incoming packet is from the ICP. Again, note that for each control request packet there is a control response packet unless the Freeway resource is not adequate or the network is congested.

Invalid ICP string received (1)

The Open Session request packet was rejected due to an invalid ICP string provided within the request packet. Make sure that your application sends a correct text string; this string must match one of the Device Names declared in the server Boot Configuration file. If you are not sure, consider using the Freeway trace feature to determine exactly what your application has sent. Refer to the *Freeway User Guide* for detailed information about obtaining a trace.

ICP is not operational (2)

The Open Session request was rejected because the target ICP is not operational. Review your Freeway's log for a possible failure in loading the ICP software. Refer to the *Freeway User Guide* for more information.

No session is available (3)

The Freeway server is out of session entries to manage a new session. This is a serious error. Make sure that your application releases unused sessions with the Close Session request. Consult your *Freeway User Guide* for more information.

Invalid Freeway session ID provided (4)

The Freeway Session ID is invalid. Your application provided an invalid Session ID. Consider using the Freeway trace feature to isolate the problem.

Invalid type field provided (5)

The Type field in the Freeway header is not supported. Consider using the Freeway trace feature to isolate the problem.

Invalid command provided (6)

The Command field in the Freeway header is not supported. Consider using the Freeway trace feature to isolate the problem.

Session is no longer active (7)

The session for which the packet is intended is no longer in service. Consider using the Freeway trace feature to isolate the problem.

ICP is off-line (8)

The target ICP is not on-line. Consider viewing the Freeway's log and make sure that your Freeway has the desired ICP installed.

Data received overflow (9)

The amount of data received from the client is less than the data size field specified in the ICP header.

Invalid data received (10)

This code is reserved; it is not currently used.

No more nodes are available (11)

The server could not assign a node to the ICP on an Open Session request because all available nodes are in use. A Close Session request may have failed or the maximum number of ICP nodes (128) may have been reached.

Client Session ID (2 bytes)

When a client application requests the Freeway server to open a new session, the Freeway server copies this field into its internal session table after the session is successfully opened. This value is contained in the Open Session request packet and is returned to the client application in all subsequent packets until the session is closed. The Freeway server does not perform a range check on this field. However, the value of this field on the Close Session request must match the value used in the Open Session request.

Freeway Session ID (2 bytes)

This field contains a unique ID from the Freeway server to the client application after the server successfully opened a session at the request of the client. The client application must use the returned value of this field for all of its subsequent packets to either the Freeway server or to the ICP and its protocol software. This field is unique within a Freeway server only. The range of values for this field is 1 to 1024 (decimal).

Reserved (2 bytes)

This area is reserved for use by the Freeway server only.

Private (20 bytes)

This area is reserved for use by the Freeway server only.

2.3.3 ICP Header Format

The ICP header immediately follows the Freeway header for all packets that are forwarded to the ICP. The ICP header consists of 16 bytes and must be in network byte order (big-endian). The use of the ICP's header is protocol-specific. For detailed description of each field, refer to programmer's guide for the appropriate protocol. The `DLI_OPT_ARGS` structure is defined in the *Freeway Data Link Interface Reference Guide* and is used for specifying packet contents when using the "raw" mode of DLI.

ICP Header

Client ID	2 bytes
Server ID	2 bytes
Length	2 bytes
Command	2 bytes
Status	2 bytes
Parameter 1	2 bytes
Parameter 2	2 bytes
Parameter 3	2 bytes

The fields in the ICP header are as follows:

Client ID (2 bytes)

This field is equivalent to the `usICPClientID` field in the `DLI_OPT_ARGS` structure. This field is used only by the X.25 protocol software that runs on the ICP. Refer to the *X.25 Low-Level Interface* for more information.

Server ID (2 bytes)

This field is equivalent to the `usICPServerID` field in the `DLI_OPT_ARGS` structure. This field is used only by the X.25 protocol software that runs on the ICP.

Length (2 bytes)

This field contains an integer value that is the sum of the length of the Protocol header and the length of the data that follows the Protocol header.

Command (2 bytes)

This field is equivalent to the `usICPCommand` field in the `DLI_OPT_ARGS` structure. For valid entries, refer to protocol specific documents.

Status (2 bytes)

This field is equivalent to the `usICPStatus` field in the `DLI_OPT_ARGS` structure. For valid entries, refer to protocol specific documents.

Parameter 1

This field is equivalent to the `usICPParms[0]` field in the `DLI_OPT_ARGS` structure.

Parameter 2

This field is equivalent to the `usICPParms[1]` field in the `DLI_OPT_ARGS` structure.

Parameter 3

This field is equivalent to the `usICPParms[2]` field in the `DLI_OPT_ARGS` structure.

2.3.4 Protocol Header Format

The Protocol header immediately follows the ICP header for all packets that are forwarded to the ICP.

The Protocol header is 16 bytes and must remain in the byte order of the machine on which the header is produced. The use of the Protocol header is protocol-specific. For a detailed description of each field, refer to the programmer's guide for the appropriate protocol. The `DLI_OPT_ARGS` structure is defined in the *Freeway Data Link Interface Reference Guide* and is used for specifying packet contents when using the "raw" mode of DLI.

Protocol Header

Command	2 bytes
Modifier	2 bytes
Link ID	2 bytes
Circuit ID	2 bytes
Session ID	2 bytes
Sequence Number	2 bytes
Reserved 1	2 bytes
Reserved 2	2 bytes

The fields in the Protocol header are as follows:

Command (2 bytes)

This field is equivalent to the `usProtCommand` field in the `DLI_OPT_ARGS` structure.

Modifier (2 bytes)

This field is equivalent to the `usProtModifier` field in the `DLI_OPT_ARGS` structure.

Link ID (2 bytes)

This field is equivalent to the `usProtLinkID` field in the `DLI_OPT_ARGS` structure.

Circuit ID (2 bytes)

This field is equivalent to the `usProtCircuitID` field in the `DLI_OPT_ARGS` structure.

Session ID (2 bytes)

This field is equivalent to the `usProtSessionID` field in the `DLI_OPT_ARGS` structure.

Sequence Number (2 bytes)

This field is equivalent to the `usProtSequence` field in the `DLI_OPT_ARGS` structure.

Reserved 1 (2 bytes)

This field is equivalent to the `usProtXParms[0]` field in the `DLI_OPT_ARGS` structure.

Reserved 2 (2 bytes)

This field is equivalent to the `usProtXParms[1]` field in the `DLI_OPT_ARGS` structure.

Chapter 3	Packet Formats
----------------------------	-----------------------

This chapter presents the format of each packet exchanged between the client application and the Freeway server. Individual ICP Protocol packets are not presented; please refer to the appropriate Freeway programmer’s guide for the protocol you are using. The sequence of packet data flow is presented in [Chapter 2](#).

3.1 TSI Control Packet Formats

Each TSI control packet contains a TSI header. Optionally, a TSI control packet can contain data immediately following the header.

3.1.1 Bind Request Packet

The bind request consists of a TSI header followed by data. Client applications should use this only if they are not using the client TSI layer.

TSI Header

Header Length	=	20
Data Length	=	10
Sequence Number	=	n ^a
Packet Type	=	1
Command	=	1
First Segment	=	n/a
Last Segment	=	n/a

^a sequence number assigned by client

Data

Max Packet Length	=	maxBufSi ze
Max Header Length	=	20
Segment Indicator	=	n/a
Buffering Indicator	=	n/a
Negotiable Indicator	=	n/a

The data fields in the Bind request packet are as follows:

Max Packet Length (2 bytes)

This field indicates the largest packet length that the sender of this packet is able to transmit and receive. Note that when the Freeway server TSI determines that its configured maximum packet length is smaller than your proposed value, it responds with another bind with the Freeway server TSI's value.

Max Header Length (2 bytes)

This field indicates the header size of the sender of this packet. This value must be the length of the TSI header explained earlier. Note that the Freeway server TSI currently does not support a variable length of the TSI header. Future releases of the TSI may include this feature.

Segment Indicator (2 bytes)

This field is currently not in use and should be set to 0.

Buffering Indicator (2 bytes)

This field is currently not in use and should be set to 0.

Negotiable Indicator (2 bytes)

This field is currently not in use and should be set to 0.

3.1.2 Unbind Request Packet

The unbind request consists of a TSI header. Client applications should use this only if they are not using the client TSI layer.

TSI Header

Header Length	=	20
Data Length	=	0
Sequence Number	=	n ^a
Packet Type	=	1
Command	=	2
First Segment	=	n/a
Last Segment	=	n/a

^a sequence number assigned by client

3.1.3 Unbind Force Request Packet

The unbind force request consists of a TSI header. Client applications should use this only if they are not using the client TSI layer.

TSI Header

Header Length	=	20
Data Length	=	0
Sequence Number	=	n ^a
Packet Type	=	1
Command	=	3
First Segment	=	n/a
Last Segment	=	n/a

^a sequence number assigned by client

3.1.4 ACK Packet

The ACK response consists of a TSI header. Client applications should use this only if they are not using the client TSI layer.

TSI Header

Header Length	=	20
Data Length	=	0
Sequence Number	=	n ^a
Packet Type	=	1
Command	=	6
First Segment	=	n/a
Last Segment	=	n/a

^a sequence number assigned by client

3.1.5 NAK Packet

The NAK response consists of a TSI header. Client applications should use this only if they are not using the client TSI layer.

TSI Header

Header Length	=	20
Data Length	=	0
Sequence Number	=	n ^a
Packet Type	=	1
Command	=	7
First Segment	=	n/a
Last Segment	=	n/a

^a sequence number assigned by client

3.2 Freeway Control Packet Formats

Each Freeway control packet contains a TSI header and a Freeway header. Optionally, a Freeway control packet can contain data immediately following the Freeway header. The Freeway control packet is a type of TSI data packet.

3.2.1 Open Session Request Packet

The Open Session request consists of a TSI header, Freeway header, and an ASCII character string specifying the ICP the client wants to communicate with during this session. The string must match a device name in the server Boot Configuration file.

TSI Header

Header Length	=	20
Data Length	=	44+data
Sequence Number	=	n ^a
Packet Type	=	2
Command	=	n/a
First Segment	=	n/a
Last Segment	=	n/a

^a sequence number assigned by client

Freeway Header

Reserved	
Shared Header Size	= 12
Private Header Size	= 20
Type	= 1
Command	= 1
Status	= n/a
Client Session ID	= n ^a
Freeway Session ID	= n/a
Reserved	
Private	

^a id assigned by client

Data

ICP Device Name ^a (1-45 chars)

^a normally, this is icp*n*, where *n* = 0,1,2...

3.2.2 Open Session Response Packet

The Open Session response consists of a TSI header and a Freeway header, followed by the data area containing an ASCII character string specifying the Freeway software version. If an error has occurred, the server returns an error code in the status field. The Freeway Session ID is returned if the request was accepted and must be included in subsequent packets sent to the server for this session.

TSI Header

Header Length	= 20
Data Length	= 44+data
Sequence Number	= n ^a
Packet Type	= 2
Command	= n/a
First Segment	= n/a
Last Segment	= n/a

^a sequence number assigned by client

Freeway Header

Reserved	
Shared Header Size	= 12
Private Header Size	= 20
Type	= 1
Command	= 1
Status	= return status code
Client Session ID	= n ^a
Freeway Session ID	= m ^b
Reserved	
Private	

^a id assigned by client

^b id assigned by server

Data

Software Version (1-20 chars)

3.2.3 Close Session Request Packet

The Close Session request consists of a TSI header and a Freeway header.

TSI Header

Header Length	=	20
Data Length	=	44
Sequence Number	=	n ^a
Packet Type	=	2
Command	=	n/a
First Segment	=	n/a
Last Segment	=	n/a

^a sequence number assigned by client

Freeway Header

Reserved	
Shared Header Size	= 12
Private Header Size	= 20
Type	= 1
Command	= 2
Status	= n/a
Client Session ID	= n ^a
Freeway Session ID	= m ^b
Reserved	
Private	

^a id assigned by client

^b id assigned by server

3.2.4 Close Session Response Packet

The Close Session response consists of a TSI header and a Freeway header.

TSI Header

Header Length	=	20
Data Length	=	44
Sequence Number	=	n ^a
Packet Type	=	2
Command	=	n/a
First Segment	=	n/a
Last Segment	=	n/a

^a sequence number assigned by client

Freeway Header

Reserved	
Shared Header Size	= 12
Private Header Size	= 20
Type	= 1
Command	= 2
Status	= return status code
Client Session ID	= n ^a
Freeway Session ID	= m ^b
Reserved	
Private	

^a id assigned by client

^b id assigned by server

3.2.5 Get Version Request Packet

The Get Version request consists of a TSI header and a Freeway header.

TSI Header

Header Length	=	20
Data Length	=	44
Sequence Number	=	n ^a
Packet Type	=	2
Command	=	n/a
First Segment	=	n/a
Last Segment	=	n/a

^a sequence number assigned by client

Freeway Header

Reserved	
Shared Header Size	= 12
Private Header Size	= 20
Type	= 1
Command	= 5
Status	= n/a
Client Session ID	= n/a
Freeway Session ID	= n/a
Reserved	
Private	

3.2.6 Get Version Response Packet

The Get Version response consists of a TSI header and a Freeway header, followed by an ASCII character string specifying the Freeway server software version.

TSI Header

Header Length	=	20
Data Length	=	44+data
Sequence Number	=	n ^a
Packet Type	=	2
Command	=	n/a
First Segment	=	n/a
Last Segment	=	n/a

^a sequence number assigned by client

Freeway Header

Reserved
Shared Header Size = 12
Private Header Size = 20
Type = 1
Command = 5
Status = n/a
Client Session ID = n/a
Freeway Session ID = n/a
Reserved
Private

Data

Software Version (1-20 chars)

3.3 ICP Protocol Packet Format

Each ICP Protocol packet contains a TSI header, a Freeway header, an ICP header, and a Protocol header. Optionally, an ICP Protocol packet can contain data immediately following the Protocol header. The ICP Protocol packet is a type of TSI data packet, and is also known as a Freeway data packet. Only a generic ICP Protocol packet is presented here; for individual ICP packet formats and their corresponding data flow, refer to the appropriate protocol programmer's guide.

TSI Header

Header Length	=	20
Data Length	=	76+ICPdata
Sequence Number	=	n ^a
Packet Type	=	2
Command	=	n/a
First Segment	=	n/a
Last Segment	=	n/a

^a sequence number assigned by client

Freeway Header

Reserved		
Shared Header Size	=	12
Private Header Size	=	20
Type	=	2
Command	=	x ^a
Status	=	n/a
Client Session ID	=	n ^b
Freeway Session ID	=	m ^c
Reserved		
Private		

^a 1,2, or 3 (see [Chapter 2](#))

^b id assigned by client

^c id assigned by server

ICP Header^a

Client ID	2 bytes
Server ID	2 bytes
Length	2 bytes
Command	2 bytes
Status	2 bytes
Parameter 1	2 bytes
Parameter 2	2 bytes
Parameter 3	2 bytes

^a The ICP header is 16 bytes; a generic header is shown here. Refer to the appropriate protocol programmer's guide for valid packet formats.

Protocol Header^a

Command	2 bytes
Modifier	2 bytes
Link ID	2 bytes
Circuit ID	2 bytes
Session ID	2 bytes
Sequence Number	2 bytes
Reserved 1	2 bytes
Reserved 2	2 bytes

^a The Protocol header is 16 bytes; a generic header is shown here. Refer to the appropriate protocol Programmer's Guide for valid packet formats.

Data

ICP data	optional
----------	----------

Index

A

ACK 24, 34
ACK data 24, 34
ACK packet 48
Audience, intended 7

B

Bind 22, 33
Bind request packet 45
Bit numbering 10
BSD socket 17, 20, 21
Byte ordering 10

C

Circuit ID 43
Client ID 40
Client session ID 39
Client-server environment 17
Client-server exchange example 30
Client-server protocol 21
Close session 27
Close session request or response 36
Close session request packet 51
Close session response 27
Close session response packet 52
Closing a Freeway connection 20
Command 33, 41, 42
Command/response 36
Connect call 21
Connection, closing 20
Connection, establishing with Freeway 20
Control 36
Control packet 33
Control packet formats
 Freeway 49

TSI 45

Control packets

 Freeway 26

 TSI 22

Conventions, document 10

Customer support 11

D

Data 36

Data length 32

Data link interface, see DLI

Data packet 33

Data packet from ICP 29, 37

Data packet to ICP 28

Data packets

 Freeway 28

 TSI 26

Data received overflow 38

Data, exchanging with Freeway 20

DLI

Document

 conventions 10

 organization 8

 purpose 7

Documents

 reference 8

E

Echo reply 24, 34

Echo request 24, 34

Electrical interface 16

Equipment, required 8

Establishing a Freeway connection 20

Ethernet 16

Example client-server exchange 30

Exchanging data with Freeway 20
Expedite data packet to ICP 29
Expedite write to ICP request 36

F

Features

Freeway 16
First segment 34
Freeway
 client-server environment 17
 features 16
 overview 14
Freeway client-server protocol 21
Freeway control packet formats 49
Freeway control packets 26
Freeway data packets 28
Freeway header format 35
Freeway packet header formats 31
Freeway packet protocol 21
Freeway session ID 39

G

Get time 28
Get time request 36
Get version 28
Get version request or response 36
Get version request packet 53
Get version response 28
Get version response packet 54

H

Header formats
 Freeway 35
 Freeway packet 31
 ICP 40
 protocol 42
 TSI 32
Header length 32
History of revisions 11

I

ICP header format 40
ICP is not operational 37
ICP is off-line 38
ICP protocol packet format 55

Intended audience 7
Invalid command provided 38
Invalid data received 38
Invalid Freeway session ID provided 38
Invalid ICP string received 37
Invalid type field provided 38

L

LAN interface processor 14
Last segment 34
Length 40
Link ID 42
Listen call 21

M

Modifier 42
MsgMux 17, 20

N

NAK 24, 34
NAK data 24, 34
NAK packet 48
No more nodes are available 39
No session is available 38

O

OK 37
Open session 27
Open session request or response 36
Open session request packet 49
Open session response 27
Open session response packet 50
Opening a Freeway connection 20
Organization of document 8
OS/Impact 16
Overview 13
 Freeway 14

P

Packet formats 45
Packet protocol
 Freeway 21
Packet type 33
Parameter 1 41
Parameter 2 41

- Parameter 3 41
- Private header size 35
- Product support 11
- Protocol header format 42
- Protocol packet format
 - ICP 55
- Protocol, client-server 21
- Purpose of document 7

- R**
- Reference documents 8
- Required equipment 8
- Reserved 35
- Reserved 1 43
- Reserved 2 43
- Revision history 11
- rlogin 16

- S**
- Sequence Number 43
- Sequence number 32
- Server ID 40
- Server processor 14
- Session ID 43
- Session is no longer active 38
- Set time 27
- Set time request 36
- Shared header size 35
- SNMP 16
- Status 37, 41
- Support, product 11

- T**
- TCP/IP 16
- Technical support 11
- telnet 16
- Trace feature 19
- Transport subsystem interface, see TSI
- TSI
 - TSI configuration file
 - client 19
 - server 19
 - TSI configuration, defining 19
 - TSI control packet formats 45
 - TSI control packets 22
 - TSI data packets 26
 - TSI header format 32
 - Type 36

- U**
- Unbind 23, 33
- Unbind force 23, 33
- Unbind force request packet 47
- Unbind request packet 47

- V**
- VxWorks 14

- W**
- WAN interface processor 16
- Write to ICP request 36

Customer Report Form

We are constantly improving our products. If you have suggestions or problems you would like to report regarding the hardware, software or documentation, please complete this form and mail it to Simpack at 9210 Sky Park Court, San Diego, CA 92123, or fax it to (858)560-2838.

If you are reporting errors in the documentation, please enter the section and page number.

Your Name: _____

Company: _____

Address: _____

Phone Number: _____

Product: _____

Problem or
Suggestion: _____

Simpact, Inc.
Customer Service
9210 Sky Park Court
San Diego, CA 92123