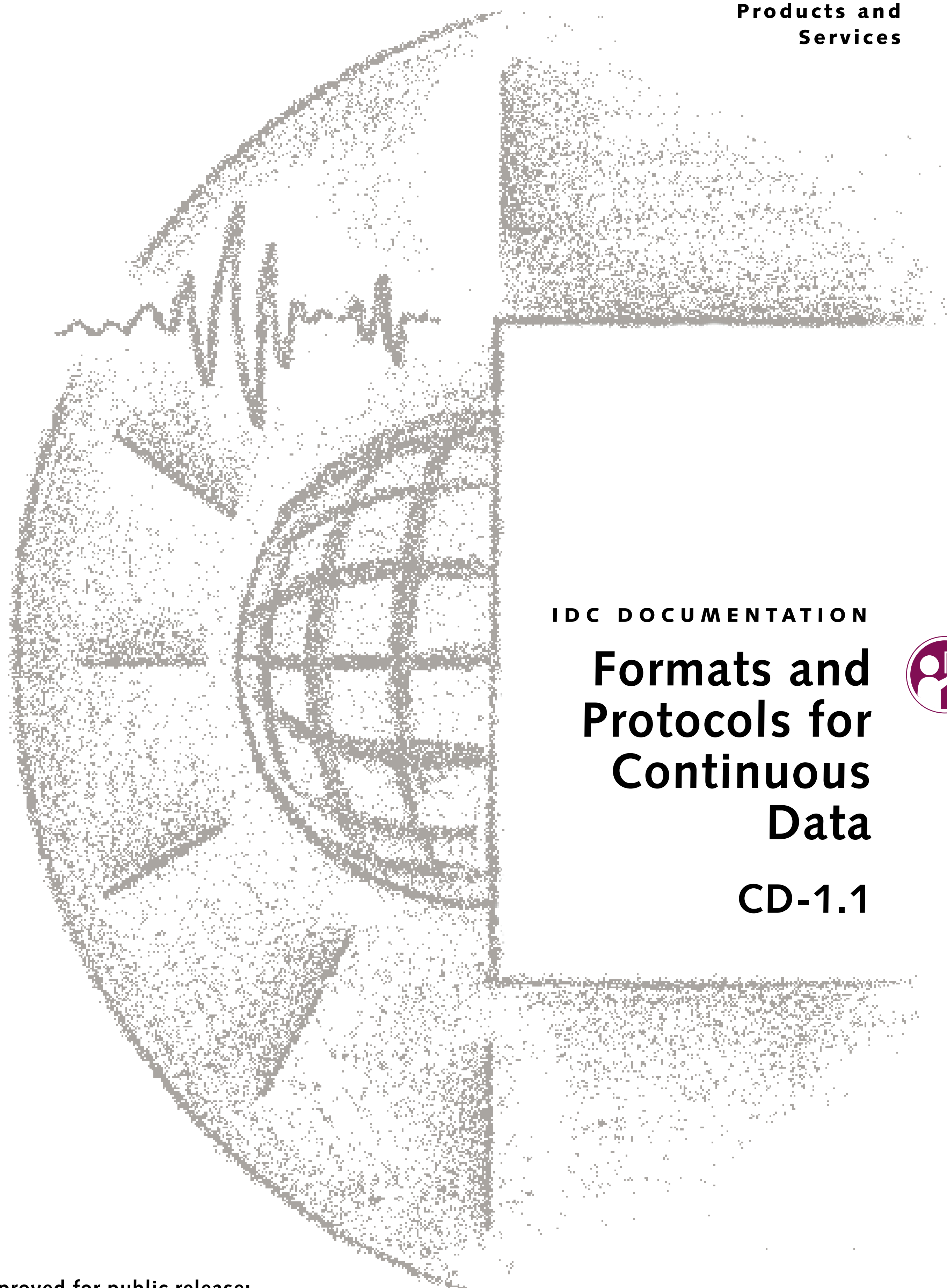
Products and Services



IDC DOCUMENTATION

Formats and Protocols for Continuous Data

CD-1.1

Approved for public release; distribution unlimited

Revision 0.2

Notice

This document was published December 2001 by the Monitoring Systems Operation of Science Applications International Corporation (SAIC) as part of the International Data Centre (IDC) Documentation. It was first published in July 2000 and was republished electronically as Revision 0.1 in September 2001 and as Revision 0.2 in December 2001 to include minor changes (see the following Change Page for Revision 0.2). IDC documents that have been changed substantially are indicated by a whole revision number (for example, Revision 1).

Contributors

Jon Berger, University of California, San Diego
Roger Bowman, Science Applications International Corporation
James Burns, Telcordia, a subsidiary of Science Applications International Corporation
David Chavez, Engineering Services & Software
Raymond Cordova, Science Applications International Corporation
John Coyne, Provisional Technical Secretariat of the CTBTO
William E. Farrell, Science Applications International Corporation
Petr Firbas, Provisional Technical Secretariat of the CTBTO
Warren Fox, Science Applications International Corporation
Jeff Given, Science Applications International Corporation
Tony McAuley, Telcordia, a subsidiary of Science Applications International Corporation
James Wang, Science Applications International Corporation

Trademarks

Ordering Information

The ordering number for this document is SAIC-01/3027Rev0.2.

This document is cited within other IDC documents as [IDC3.4.3Rev0.2].

Change Page

This document is Revision 0.2 of the Formats and Protocols for Continuous Data CD-1.1. The following changes have been made for this publication:

Page	Change
as needed	Added change bars (vertical lines that appear in the margin) to identify new or revised material.
Notice	Changed Notice Page to include an explanation of the Revision number, a reference to the Change Page, a new SAIC number, and a new document number.
Notice	Inserted trademarks for IEEE, Motorola, SAIC, and Sun.
Change Page	Updated this Change Page.
ii	Added text describing the revisions to the Preface.
ii	Changed Preface text about previous changes to past tense and added a reference to the previous version of the document.
iv	Deleted "The standard will also be …"
iv	Deleted "issued by the IDC"
v	Inserted "core"
v	Replaced "is in development. It …" with "capability has been developed and …"
All	Document name changed from "Formats and Protocols for Continuous Data CD-1.1–Revision 0.1" to "Formats and Protocols for Continuous Data CD-1.1–Revision 0.2"
All	Document number updated to reflect revision; "IDC-3.4.3Rev0.1" to "IDC-3.4.3Rev0.2"
All	Publish date changed from "September 2001" to December 2001"
1	Deleted "Negotiating Options" bullet
2	Replaced "Either" with "Currently, "
2	Deleted "or the data consumer …"
2	Deleted "with its port number …"
2	Inserted "of a data consumer …"
3	Replaced "The" with "All"
3	Replaced "incorporates" with "frames contain a"

Page	Change
3	Replaced “numbers for all frames, which” with “number field. Sequence numbers”
4	Replaced “negotiate” with “identify”
4	Replaced “Immediately following” with “Following”
4	Inserted “aid in establishing connections and ”
5	Deleted “Negotiating Options” heading.
5	Inserted “Currently”
5	Replaced “may only be” with “are only”
6	Replaced “Digital Signature Standard (DSS)” with “authentication signature”
6	Replaced “DSS ” with “authentication”
7	Replaced “DSS ” with “authentication signature”
7	Replaced “or to request …” with “and may be employed to …”
7	Deleted “An assumption in designing the …”
7	Inserted “Given a fundamentally reliable underlying circuit,”
10	Deleted “Frames are clearly data structures of applications.”
10	Replaced “nine” with “ten”
10	Replaced “Five” with “Six”
11	Inserted “A CD-1 encapsulation frame …”
12	Deleted “The header includes the authentication …”
12	Inserted “Frame Header”
12	Deleted “, a series number,”
12	Replaced “field” with “fields”
12	Inserted “and destination”
12	Inserted paragraph beginning with “The frame creator and frame destination fields …”
12	Inserted “Sequence numbers are used to …”
13	Inserted “(with the same creator and destination designations)”
13	Inserted “Sequence numbers are assigned …”

Page	Change
13	Inserted “Connection Request” bullet
13	Deleted “This authentication of the communication …”
15	Replaced “pointer to” with “identifier of”
15	Inserted “Digital Signature Standard”
16	Replaced “These ” with “Unless otherwise specified, all ”
16	Inserted “left justified and ”
16	Inserted “Connection Request Frames employ …”
17	Replaced “IDC, IMS, NDC” with “requester type (e…”
17	Inserted “Connection Response Frames employ …”
18	Inserted “only ”
18	Replaced “After establishing a connection” with “However”
18	Replaced “these frames ” with “future developments ”
18	Replaced “be used by either side …” with “support using these frames ”
18	Replaced “Multiple ” with “, where multiple ”
18	Replaced “Examples of possible options are ” with “The currently recognized option is ”
22	Replaced “null” with “null (if necessary)”
22	Replaced “null” with “null (if necessary)”
22	Replaced “null.” with “null (if necessary) …”
22	Inserted “For example, the site …”
23	Inserted “See Table 17: …”
24	Inserted “A frame set is a …”
25	Replaced “Details about ” with “When an Acknack Frame is …”
25	Replaced “use ” with “same as at any other …”
25	Replaced “this frame are ” with “sequence number gaps. Additional …”
25	Inserted “of this frame are ”
25	Replaced “lowest valid sequence number in …” with “lowest valid sequence number sent …”

Page	Change
25	Replaced “highest valid sequence number in …” with “highest valid sequence number considered …”
26	Inserted “Because of their very …”
40	Replaced “alphabetic” with “alphanumeric”
40	Inserted “0 if destination is indeterminate”
41	Inserted “(space, not new …”
41	Deleted “frame ”
41	Replaced “” with “3”
41	Replaced “required” with “0”
41	Replaced “0 for current data series …” with “0, positive integers …”
41	Replaced “alphabetic” with “alphanumeric”
42	Replaced “ASCII alphanumeric string” with “FDSN location name”
43	Replaced “30” with “20”
44	Replaced “As previously stated, ” with “The duration of ”
44	Deleted “IMS requires that a ”
44	Replaced “hold 10 (or ≤ 30 …” with “will be the same as …”
44	Replaced “10, or 30 seconds …” with “described in the previous section”
45	Deleted “and series ”
45	Replaced “Station ” with “Data Provider ”
45	Replaced “station ” with “data provider ”
45	Inserted “In some cases, data …”
46	Replaced “the same standards as for …” with “IMS station naming conventions[…”
46	Replaced “naming ” with “names for seismic stations ”
46	Replaced “for instrument and channel …” with “. Channel names that are …”
46	Replaced “defines names for seismic, …” with “shall conform to IMS ”
46	Replaced “types. Standard channel names …” with “naming conventions [IMS01]”
47	Inserted “. Acknack Frames employ the …”

Page	Change
47	Replaced “Examples are ” with “An example would be a ”
47	Replaced “requests or requests for a …” with “request”
48	Inserted “The sequence of frames …”
50	Inserted “bit 1 1 = …”
51	Inserted new references
G1	Deleted “Comprehensive Nuclear Test-Ban Treaty” from Glossary
G1	Replaced “Comprehensive Nuclear Test-Ban Treaty” with “Comprehensive Nuclear-Test-Ban Treaty”
G2	Deleted “Frame Store” from Glossary
G2	Replaced “Element of a …” with “Data store defined for each source …”
G3	Replaced “Provisional Technical Secretariat” with “PTS”
G3	Replaced “(PTS) body” with “Provisional Technical Secretariat”
G3	Replaced “control protocol - internet protocol” with “Transmission Control Protocol/ Internet Protocol”
G3	Deleted “Technical Secretariat” from Glossary
G3	Deleted “Treaty Users” from Glossary

Formats and Protocols for Continuous Data CD-1.1

CONTENTS

Formats and Protocols for Continuous Data CD-1.1

FIGURES

Formats and Protocols for Continuous Data CD-1.1

TABLES

About this Document

This chapter describes the organization and content of the document and includes the following topics:

- Preface
- Purpose
- Scope
- Audience
- Background
- Related Information
- Using this Document

About this Document

PREFACE

This second minor revision to Continuous Data (CD-1.1) formats and protocols clarifies the specification of CD-1.1. It addresses questions and requests for additional information from several organizations that are planning independent implementations of CD-1.1 or are using Science Application International Corporation's (SAIC) implementation of CD-1.1. The second revision is limited to clarification and does not change or extend the formats and protocols themselves. SAIC prepared this revision in cooperation with representatives of the Provisional Technical Secretariat.

The first revision to Continuous Data (CD-1.1) formats and protocols redefined some attributes of the *channel description* fields of the Channel Subframe and introduced the CD-1 Encapsulation Frame.

The following changes were made in the first revision [IDC3.4.3Rev0.1]:

1. The *channel description* format in Table 10 on page 23 was revised to incorporate a new CSS-3.0 data type and include calibration information. The name of byte 2 was changed from "compression" to "transformation" and references were inserted to bytes 15–16. Byte 4, previously unused, is now an option flag indicating whether or not calibration information is provided in bytes 17–24. Bytes 17–20 and 21–24, previously unused, now contain the calibration factor and calibration period, respectively.
2. A new CD-1 Encapsulation Frame was added to the protocol so that CD-1 data can be transmitted through the CD-1.1 protocol ("CD-1 Encapsulation Frame" on page 28). CD-1 Encapsulation Frame rows

were added to Table 1 on page 11 and Table 2 on page 14, and a new data type for CD-1 encapsulated data was added to Table 17 on page 34.

The original document [IDC3.4.3] incorporated minor format changes to the draft posted on the Expert Communications System (ECS) of the Provisional Technical Secretariat (PTS) in August 1999. These changes were recommended by the developers of the reference implementation of the software supporting the protocol. The following changes were incorporated in the original document [IDC3.4.3]:

1. Two new fields, *channel string* and *channel string count*, were added to the Channel Subframe Header (Table 9 on page 22). These fields allow the application software to manage the data frames.
2. The definition of the *channel description* field of the Data Frame Channel Subframe (Table 10 on page 23) was augmented to include a full description of the data type.
3. In the Command Request and Command Response Frames (Table 13 on page 27 and Table 14 on page 28), the descriptions of the site, channel, and location were made consistent with usage in the Channel Subframe.
4. The section on the layered protocol model was deleted. This section no longer reflected the actual structure of the reference implementation.
5. Chapter 4: Policies for Continuous Data Transmission was added to define policies used with the CD-1.1 formats and protocols. This chapter provides a table of valid values for many elements of the protocol. The values indicated in Table 20 on page 40 are used to check adherence to the protocol.

PURPOSE

This document describes the formats and protocols of Continuous Data (CD-1.1), a new standard to be used for transmitting continuous, time series data from stations of the International Monitoring System (IMS) to the International Data Centre (IDC) and for transmitting these data from the IDC to National Data Centers (NDCs). The protocol is designed to support multicasting of continuous data, which entails more complex data flow topologies.

SCOPE

This document describes the CD-1.1 application-level data structures (formats) and the protocols for establishing connections, transmitting data, and communicating with stations. The formats have numerous options to support features of the protocol, such as mode of data compression, provision of ancillary data, and so on. Not all options are defined at this time. In addition, a variety of policies need to be agreed upon, such as maximum data frame sizes, the method to be used to initiate connections, and the handling of station commands.

This document does not describe the software for sending, receiving, or forwarding continuous data.

AUDIENCE

This document is intended for software developers, systems engineers, and users of the IMS and IDC concerned with the exchange of data from stations of the primary seismic network, the hydroacoustic network, and the infrasonic network.

BACKGROUND

The first protocol for the exchange of time series data over Internet Protocol (IP) networks was defined in 1995 [GSE95a]. It was developed for the Group of Scientific Experts Third Technical Test (GSETT-3) and has been used world-wide ever since. Originally this was called the "alpha" protocol, a word that appears in older documents and is embedded in many software names. Now, this protocol is called CD-1. CD-1 is well established and has many years of use remaining because of the large, installed base.

Operational experience and changing requirements called for a revamping and improvement of CD-1. To this end, an Informal Technical Workshop on Formats and Protocols for Continuous Data was held at the Center for Monitoring Research (CMR) on 12–13 February, 1998 [Bow98], [SAI98a]. The recommendations of the workshop were presented in [SAI98b].

The workshop report was presented to Working Group B at the seventh session in August, 1998. Afterwards, it was posted to the PTS ECS website for discussion and comment. Comments from this and other sources have been considered in preparing this description of CD-1.1.

Software that implements core CD-1.1 capability has been developed and was delivered to the IDC in December 2000.

RELATED INFORMATION

CD-1 is described in [IDC3.4.2]. The immediate background for the new protocol is given in [Bow98] and [SAI98a].

USING THIS DOCUMENT

This document is part of the overall documentation architecture for the IDC. It is part of the Products and Services document category, which provides descriptions of IDC products and their formats.

This document is organized as follows:

- Chapter 1: Continuous Data Protocol

  This chapter provides a high-level description of the protocol that is followed to establish a connection and transmit continuous data.

- Chapter 2: Frame Formats

  This chapter describes the formats of the frames that are used to establish connections, transfer data, and alter connections.

- Chapter 3: Data Formats

  This chapter provides the formats of the data that are transferred.

- Chapter 4: Policies for Continuous Data Transmission

  This chapter describes the policies that govern the correct implementation of the protocol.

- References

  This section lists the sources cited in this document.

- Glossary

  This section defines the terms, abbreviations, and acronyms used in this document.

- Index

  This section lists topics and features provided in this document along with page numbers for reference.

Conventions

This document uses a variety of conventions, which are described in the following tables. Table I shows the conventions for data flow. Table II illustrates typographical conventions. Table III describes terms that are not part of the standard Glossary, which is found at the end of this document.

TABLE I: DATA FLOW SYMBOLS

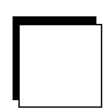
Description	Symbol
external source or sink of data	
control flow	– – – →
data flow	⟶

Table II: Typographical Conventions

Element	Font	Example
field or user-defined argument	*italics*	*frame type*
processes and software units		*inetd*
computer code and output	`courier`	`>(list 'a 'b 'c)`
filenames, directories, and websites		`pub/incoming/docs`
text to be typed as shown		`edit-filter-dialog`

Table III: Terminology

Term	Description
CD-1	Continuous Data format version 1 (also referred to as the "alpha" protocol)
CD-1.1	Continuous Data format version 1.1
compressed data	data that have been reduced significantly in size to make transmission more efficient
CRC	Cyclic Redundancy Check
data consumer (dc)	computer that is receiving the continuous data
data producer (dp)	computer that is sending the continuous data
ECS	Expert Communications System
format	data structure used by application programs to pass information across an interface
forward byte ordering	see network byte ordering
frame	logical collection of digital information that is transmitted as a unit from application to application
hexadecimal	base 16 mathematics; A = 10, B = 11, and so on
inetd	UNIX process that waits for network connections
multicast	one-to-many communication channel

TABLE III: TERMINOLOGY (CONTINUED)

Term	Description
network byte ordering	byte order for integers and floats in which the most significant byte is transmitted first (also called forward byte ordering)
port or port number	16-bit number indicating an address of a computer-to-computer connection based on the Internet Protocol (IP)
protocol	set of conventions that establishes the sequence of frames transmitted between sender and receiver
unicast	one-to-one communication channel

Chapter 1: Continuous Data Protocol

This chapter describes the Continuous Data (CD-1.1) protocol, which is used to establish a connection, transmit data, alter a connection, and, optionally, deliver a command to a station. The following topics are included:

- Introduction
- Establishing Connections
- Transmitting Data
- Commanding Stations
- Terminating Connections
- Security and Reliability

Chapter 1: Continuous Data Protocol

INTRODUCTION

The new Continuous Data (CD-1.1) protocol provides a method for reliably sending a continuous data stream from application software at a data provider to application software at a (or possibly several) data consumer. Data providers will be stations, NDCs, or the IDC itself, depending on topology. Data consumers will be the IDC and NDCs. Stations may be consumers as well, if this protocol is used to pass commands to stations.

Currently, the data provider (dp, the source of the data) instigates a connection over an IP network by sending a connection request to a well-known address and port of a data consumer (dc, the sink of the data). The receiver of the connection request replies by returning address and port information for a connection. After an initial handshake, the dp application commences to transmit its continuous data in frames to the assigned address and port.

The data may be provided in compressed or uncompressed format. Ancillary channel status information is provided along with the data. The data will include authentication signatures for stations meeting IMS specifications. Data from stations running legacy systems that have yet to be upgraded to IMS specifications may omit the signatures.

The protocol uses units of information called frames. Frames are application-level data structures, which consist of numerous fields or data items. Each frame conveys a specific type of information. For all frames, the CD-1.1 protocol incorporates a communication verification field, which uses a Cyclic Redundancy Check (CRC) [Sar88] to verify the correct transmission of frames. It incorporates an optional authentication field, which may be used to establish the identity of the frame's sender or the integrity of the frame's data. All protocol frames contain a sequence number field. Sequence numbers are assigned sequentially by the frame creator for non-transient frame types, for example, data frames. The sequence number size (64 bits) allows frames to be unique, even if the application runs for decades. The protocol further includes Acknack Frames. These are exchanged periodically and indicate which sequence numbers have been received and sent. They provide error recovery mechanisms for application-level error control of all frames. Acknack Frames flow in both directions and provide the mechanisms for keep-alive and for obtaining information about the state of the application at the other end of the connection.

The CD-1.1 protocol does not assume that the underlying network is reliable. As with the present protocol [IDC3.4.2], applications may exploit the error control of Transmission Control Protocol (TCP) to reduce packet loss. Alternatively, the implementation of the CD-1.1 protocol may elect to run on top of User Datagram Protocol (UDP) or another transport protocol. The CD-1.1 protocol supports sending to multiple receivers, using either forwarding or IP multicast.

The CD-1.1 protocol optionally provides a means for the data consumer to use the circuit to issue commands destined for a station of the data provider. In addition, the protocol supports the echoing of commands and command responses, if stations or NDCs choose to implement this feature. The suite of valid commands is not presently defined. The frame authentication field, described above, provides the means to validate commands.

Figure 1 indicates the succession of frames in the simplest case of a dp sending to a single dc. Time in this figure flows from top to bottom, and the dc ports to which the frames are addressed are indicated with capital letters. Usually, one frame at most is being transmitted (or processed) at any instant.

The Connection Request Frame is sent from an ephemeral port to a well-known port W of a connection manager. The connection manager is an application, which continuously monitors the well-known port. The connection manager could be at the dp, the dc, or could be a well-known address of an independent system. The latter possibility is pertinent to a multicast topology.

The Connection Request and Connection Response Frames create a connection between the dp and one or more dcs on an assigned port A and identify the protocol version to be used. In general, only data providers (data senders) initiate connection requests, although the protocol does not preclude initiation by data consumers.

Following the connection, the provider may begin data transfers. At regular intervals thereafter, Acknack Frames are exchanged, through which data frames missing from the sequence are identified for retransmission. The Acknack Frame also provides a keep-alive signal if data are flowing perfectly or if the circuit is idle.

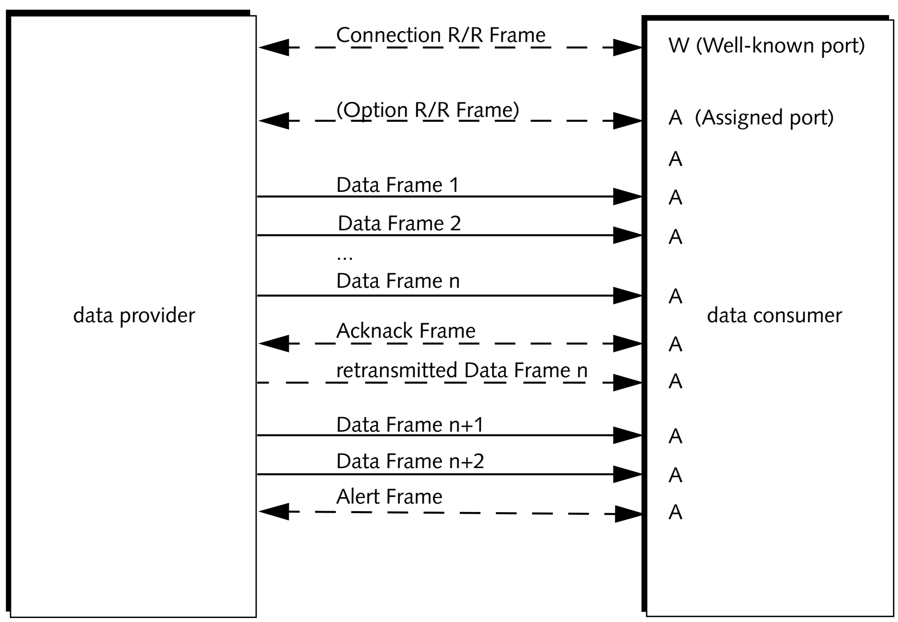


FIGURE 1. CONTINUOUS DATA COMMUNICATIONS PROTOCOL

The pair of Option Frames (Option Request and Option Response) are available to aid in establishing connections and set parameters of the protocol or the applications if required. Data Frames, which comprise the bulk of the traffic, transfer the data. Periodic Acknack Frames are sent by both the dp and dc to provide information about their state and to improve reliability. Alert Frames are used for gracefully terminating the connections.

Command Frames (not indicated in Figure 1) are optionally used to initiate an action at a station or to echo commands and responses of stations requested by other means.

ESTABLISHING CONNECTIONS

To initially establish a connection, the dp sends a Connection Request Frame to the dc using a well-known IP address and port. A typical example would be `ts.idc.org:8000`. The dc responds with a Connection Response Frame. The Connection Request and Connection Response Frames contain information about the connection type (for example, TCP, UDP unicast, or UDP multicast) and the IP address and port to use for the new connection. When a Connection Request Frame is received, the identity of the sender is optionally validated by verifying the frame authentication field.

All entities supporting Connection Request Frames must have receiver software (such as the UNIX process *inetd*) running at well-known port W at all times (Figure 1). Port W is fixed, but ports A may vary from connection to connection. Upon receiving the Connection Response Frame, the initiator of the request drops the original connection to W and uses port A for all subsequent communications.

The connection activity allows for negotiation of the protocol version to use. The protocol also defines a second IP address and port, which could be used for possible future support of multicasting.

After the two ends are connected, the applications exchange Option Frames. These frames set various parameters and conventions of the protocol. Currently options are only negotiated before the dp sends data frames. The set of valid options is defined in the policy of this protocol.

TRANSMITTING DATA

The dp may send Data Frames as long as the connection remains open. Data Frames are self-describing and contain the formatting information needed to interpret the data. Some station information is also contained in the Data Frames.

COMMANDING STATIONS

The protocol includes two frames for carrying station commands and command responses (Command Request and Command Response Frames). The method of implementing commands at a station is based on the policy established by the dp and is not a part of this protocol.

TERMINATING CONNECTIONS

Connections may be intentionally terminated by either end. The party terminating the connection sends an Alert Frame and then drops the connection.

Connections may be unintentionally terminated due to a slow or failed data provider system, data consumer system, or communications system. The conditions for unintentional terminations are detected by the provider and consumer applications and are reported in an application log file.

SECURITY AND RELIABILITY

The security of sensor data is supported by authentication signature fields within the Channel Subframe of a Data Frame. Ultimately, these signatures will be required to be applied at the sensor, but for the interim they may be applied at an intermediate point.

The protocol provides security of communications as an option. This security is exercised through implementation of authentication fields in the trailer of all frames. Use of frame authentication makes the system resistant to certain kinds of hostile attacks.

The CD-1.1 protocol provides security of commands. This security is accomplished through use of authentication signature fields in the trailer of the Command Request Frame.

The CD-1.1 protocol provides many ways to improve the reliability of transmissions. Sequence numbers are used to provide unambiguous indication of missing frames. Acknack Frames are introduced to exchange information between the provider and consumer. This frame has fields that can be used to confirm that frames have been properly received and may be employed to trigger retransmission of frames. Given a fundamentally reliable underlying circuit, Acknack Frames can be sent infrequently and will consume negligible bandwidth. The basic information in this frame is the highest and lowest sequence number received and any unfilled gaps in sequence numbers. The provider and consumer applications will use this information to supply missing frames and to note the end of the transaction for successful transmissions.

The protocol is intended for use with either TCP or UDP datagrams and to support reliable multicasting of UDP datagrams as a possible future enhancement. Software implementing the protocol over the UDP transport layer will need to incorporate an internal reliability mechanism with features similar to those provided by TCP.

Chapter 2: Frame Formats

This chapter describes the formats of frames required for continuous data exchange and includes the following topics:

- Introduction
- General Frame Structure
- Standards
- Connection Request Frame
- Connection Response Frame
- Option Request Frame
- Option Response Frame
- Data Frame
- Acknack Frame
- Alert Frame
- Command Request Frame
- Command Response Frame
- CD-1 Encapsulation Frame

Chapter 2: Frame Formats

INTRODUCTION

A frame of the protocol is an interface data structure. As such it is defined by its format, which explains the meaning of the sequence of bytes passed between the sending application and the receiving application.

The CD-1.1 protocol defines ten types of frames. The original protocol, CD-1, defined five frames [IDC3.4.2]. The mapping between CD-1 frames and CD-1.1 frames is indicated in Table 1. The table shows that four CD-1 frames have direct counterparts in CD-1. One CD-1 frame, Data Format, has been subsumed into the Data Frame. Six new frames are introduced.

The Option Request/Option Response Frame pair is introduced to permit data providers and data consumers to negotiate options of the protocol or to initialize the transfer. An Acknack Frame is introduced to support transactional integrity of data transfers. These frames are exchanged periodically to confirm that frames have been received correctly and written to a durable store.

The CD-1.1 Data Frame combines both the Data and Data Format Frames of CD-1. The CD-1.1 Channel Subframe is designed to be self describing, and it therefore does not require a separate format description frame. In addition, and following CD-1, a certain amount of information about the station is contained in the Channel Subframe.

The Command Request/Command Response Frame pair is introduced. These optional frames are provided for conveying commands to stations, or, in the case of independent subnetworks, to station operators. They may also be used to echo commands and command responses if stations and NDCs elect to implement this capability.

A CD-1 encapsulation frame is defined to support capture of a CD-1.0 data frame, especially its contained authentication signature within the CD-1.1 protocol.

TABLE 1: SUMMARY OF CHANGES IN FRAME STRUCTURES

Frame Names		Changes
CD-1.1	**CD-1**	
Connection Request	same	adds standard header and trailer adds protocol *version* field and (optional) second port
Connection Response	Port Assignment	adds standard header and trailer adds protocol *version* field and (optional) second port
Option Request	not applicable	new frame with three fields for each option
Option Response	not applicable	new frame with three fields for each option
none	Data Format	dropped as a distinct frame all fields are embedded in the Data Frame
Data	same	combines the current Data and Data Format Frames adds standard header and trailer omits *maximum frame size*, *description*, and *description length* fields allows a variety of status fields
Acknack	not applicable	new frame
Command Request	not applicable	new frame to carry commands from the dc (or other authority) to the station or the station interface for independent subnetworks
Command Response	not applicable	new frame to carry the result of a command, which may be just an acknowledgement or a quantity of data
Alert	same	adds standard header and trailer
CD-1 Encapsulation	not applicable	new frame

GENERAL FRAME STRUCTURE

All frames have a similar structure, consisting of a header, a payload, and a trailer. Each frame has a unique type identifier, which appears in the first field of the frame header. The list of frames and identifiers is given in Table 2. The structures of the headers and trailers are defined in Tables 3 and 4.

Headers and trailers introduce a number of integrity features to the protocol. These features are designed to increase reliability and security and to simplify software.

Frame Header

The header includes the authentication offset, which allows an application program to jump directly to the frame trailer. The total frame size is easily calculated from this number and data in the trailer. The header also has a sequence number and fields to identify the frame creator and destination. These three fields serve to specify a frame uniquely.

The frame creator and frame destination fields of the frame header are used to establish the name of a data store containing data received from a data provider by a data consumer. This data store is called a frame set. The naming convention uses the name of the creator followed by a colon (:), followed by the destination name. For example, for frames other than Data Frames that have frame creator = 'KCC' and frame destination = 'IDC' would have an associated frame set name 'KCC:IDC'. Because Data Frames may be forwarded to more than one destination, the frame destination is indeterminate, which is codified with an ASCII zero (0). For the creator in the example above, this would result in the frame set name 'KCC:0'. The frame creator and frame destination may be different for different frames during a single connection between a data provider and a data consumer. Frame set names derived from frame header fields are used in the frame set acked field of the Acknack Frame.

Sequence numbers are assigned by the creator of a frame. Sequence numbers are used to support application level error detection and used to track a frame's delivery to a protocol peer. The first sequence number ever generated by a given creator should be 1. Sequence numbers are assigned in an increasing sequence without gaps thereafter. After a graceful or ungraceful termination and restart, sequence numbers resume with the next number after the last one created before the termination. No two distinct frames (with the same creator and destination designations) are assigned the same sequence number. Sequence numbers are assigned to frames, not to the transaction of a frame. An assigned sequence number is valid for all senders and recipients of the frame. Sequence numbers are not needed for the following frame types:

- Connection Request
- Connection Response
- Option Request - when used for connection establishment
- Option Response - when used for connection establishment
- AckNack

All frame types use a common frame header; in the frame types identified above the sequence number field may be assigned a constant value.

Frame Trailer

The frame trailer provides three fields for frame authentication. These fields are *authentication key identifier*, *authentication size*, and *authentication value*. The three frame authentication fields are provided in the case where the receiver of the frame needs assurance the sender is correctly identified and the contents have not been tampered with. The scope of the authentication field is the frame header and payload. The last field of each frame trailer is the *comm verification* value used to verify the correct transmission of the frame. The *comm verification* is a 64-bit CRC value calculated over the entire frame (header, payload, and trailer). The polynomial use for the CRC calculation is as follows:

$$x^{64} + x^{4} + x^{3} + x^{1} + x^{0}$$

TABLE 2: FRAME TYPE IDENTIFIER

Frame Type	Frame Type Identifier
Connection Request	1
Connection Response	2
Option Request	3
Option Response	4
Data	5
Acknack	6
Alert	7
Command Request	8
Command Response	9
CD-1 Encapsulation Frame	13

TABLE 3: FORMAT OF FRAME HEADER

Field	Format	Description
frame type	IEEE integer	numeric identifier of this frame type
trailer offset	IEEE integer	byte offset from first byte of the frame to the beginning of the trailer
frame creator	8-byte ASCII	assigned identifier of the creator of the frame
frame destination	8-byte ASCII	identifier of the destination of the frame
sequence number	IEEE long	sequence number assigned by the frame creator
series	IEEE integer	series number assigned by the frame creator

TABLE 4: FORMAT OF FRAME TRAILER

Field	Format	Description
authentication key identifier	IEEE integer	identifier of the certificate with the public key required to verify the authentication value field; if non-zero, then authentication is used to verify communications
authentication size	IEEE integer	unpadded length in bytes of next field; zero when no authentication; that is, authentication key identifier is zero
authentication value	N data bytes	Digital Signature Standard (DSS) signature, padded to be divisible by four. DSA is 40 bytes. This field may have a length of zero.
comm verification	IEEE long	error detection for transmission verification

STANDARDS

The protocol adheres to standards for number and time representation as described in the following paragraphs.

Representation of Numbers

The format uses the Institute for Electrical and Electronic Engineers (IEEE) standard for numerical representation. IEEE integers (or shorts or longs) having four (or two or eight) bytes are used exclusively for fixed point numbers. IEEE floats are used for floating point numbers. The IEEE representation implies forward byte ordering of multibyte values (also known as "network byte order") and is used natively on Motorola and Sun computers, but not Intel Central Processing Units (CPUs). Values generated on machines that do not use IEEE as their native numerics may need to swap bytes before transmitting data.

Representation of Strings

Strings are represented with the ASCII character set and are padded as described in the following sections. Unless otherwise specified, all strings are left justified and only NULL-terminated if so dictated by the padding rules.

Byte Alignment

All portions of this format that are of variable length have a length that is a multiple of four. A variable-length field that does not have a length that is a multiple of four must be padded to the next multiple of four with trailing bytes set to NULL (0).

Universal Coordinated Time

Protocol time is Universal Coordinated Time (UTC) and is represented as a 20-byte ASCII string (yyyyddd_hh:mm:ss.ttt), where "_" indicates space. This format is compliant with ISO 8601: 1988 Date/Time Representations. UTC is the representation used by Global Positioning System (GPS) and most other time-keeping systems. This time has a variable number of seconds in a year because of leap years and leap seconds.

CONNECTION REQUEST FRAME

To establish a connection, a data provider submits a Connection Request Frame (Table 5) to a well-known port. Major and minor version numbers represent a protocol compliance specification. These values are mapped to the version of this document as follows. The *major version* value represents the first-order revision number of this document. The *minor version* value represents the second-order revision number. For example, if this document was revision 1.3, *major version* would contain a value of 1, and *minor version* would contain a value of 3. Connection Request Frames employ a standard frame header containing a sequence number. Given the transitory nature of the Connection Request Frame, sequence numbers are not assigned to these frames.

TABLE 5: CONNECTION REQUEST FRAME

Field	Format	Description
frame header	see Table 3	(frame type = 1)
major version	IEEE short	major protocol version requested by sender of the frame
minor version	IEEE short	minor protocol version requested by the sender of the frame
station name	8-byte ASCII	station requesting the connection
station type	4-byte ASCII	requester type (e.g., IMS, IDC, NDC)
service type	4-byte ASCII	TCP, UDP, and so on
IP address	IEEE integer	IP address of the requester
port	IEEE short	port of the requester
second IP address	IEEE integer	reserved for possible multicasting use
second port	IEEE short	reserved for possible multicasting use
frame trailer	see Table 4	

CONNECTION RESPONSE FRAME

The Connection Response Frame is returned to the connection originator by a process listening at the well-known port W (see Figure 1 on page 4). The structure of the Connection Response Frame is shown in Table 6. See the previous section, Connection Request Frame, for a description of version identification values. Connection Response Frames employ a standard frame header containing a sequence number. Given the transitory nature of the Connection Response Frame, sequence numbers are not assigned to these frames.

TABLE 6: CONNECTION RESPONSE FRAME

Field	Format	Description
frame header	see Table 3	(frame type = 2)
major version	IEEE short	major protocol version requested by sender of the frame
minor version	IEEE short	minor protocol version requested by the sender of the frame
responder name	8-byte ASCII	name of station responding to the connection request
responder type	4-byte ASCII	IDC, IMS, NDC
service type	4-byte ASCII	TCP, UDP, and so on
IP address	IEEE integer	responder's IP address for frame transfer
port	IEEE short	responder's port to be used for frame transfer
second IP address	IEEE integer	reserved for possible multicasting use
second port	IEEE short	reserved for possible multicasting use
frame trailer	see Table 4	

OPTION REQUEST FRAME

This frame and its companion, the Option Response Frame, are only exchanged as part of the connection establishment process. However, future developments may support using these frames to designate desired parameters such as a start time or list of channels, where multiple Option Request Frames may be sent if necessary. Table 7 defines the Option Request Frame. The currently recognized option is provided in "Policies for Continuous Data Transmission" on page 39.

TABLE 7: OPTION REQUEST FRAME

Field	Format	Description
frame header	see Table 3	(frame type = 3)
option count (n)	IEEE integer	number of option requests
option 1 type	IEEE integer	numeric identifier of option requested
option 1 size	IEEE integer	unpadded length in bytes of next field
option 1 request	N-byte ASCII	value of Option Request 1, padded to be divisible by four
option 2 type	IEEE integer	numeric identifier of option requested
option 2 size	IEEE integer	unpadded length in bytes of next field
option 2 request	N-byte ASCII	value of Option Request 2, padded to be divisible by four
...		
option n type	IEEE integer	numeric identifier of option requested
option n size	IEEE integer	unpadded length in bytes of next field
option n request	N-byte ASCII	value of Option Request n, padded to be divisible by four
frame trailer	see Table 4	

OPTION RESPONSE FRAME

The Option Response Frame is used to echo option requests. If the echoed value is different from the requested value, further pairs of Option frames are exchanged until agreement is reached. Table 8 defines the Option Response Frame.

TABLE 8: OPTION RESPONSE FRAME

Field	Format	Description
frame header	see Table 3	(frame type = 4)
option count (n)	IEEE integer	number of option responses
option 1 type	IEEE integer	numeric identifier of option requested
option 1 size	IEEE integer	unpadded length in bytes of next field
option 1 response	N-byte ASCII	value of Option Response 1, padded to be divisible by four
option 2 type	IEEE integer	numeric identifier of option requested
option 2 size	IEEE integer	unpadded length in bytes of next field
option 2 response	N-byte ASCII	value of Option Response 2, padded to be divisible by four
…		
option n type	IEEE integer	numeric identifier of option requested
option n size	IEEE integer	unpadded length in bytes of next field
option n response	N-byte ASCII	value of Option Response n, padded to be divisible by four
frame trailer	see Table 4	

DATA FRAME

The CD-1.1 Data Frame contains both a description of its data and the actual data values themselves. Thus, the frame combines the fields of both the Data and Data Format Frames of the CD-1 protocol. In addition, the *status* field of the Channel Subframe is allowed to be of variable size, and its definition may vary among different implementations of CD-1.1. As with all frames, the standard frame header and frame trailer surround the payload.

In normal operation, Data Frames comprise the bulk of the transmission. Each Data Frame consists of the standard Frame Header, a header for the Channel Subframes, one or more Channel Subframes, and a standard Frame Trailer. Figure 2 illustrates these relationships. The Channel Subframe Header and Channel Subframes are described in Tables 9 and 10.

The Channel Subframe Header describes the Channel Subframes to follow. The nominal time field is the Data Frame's time signature; times for each Channel Subframe are given therein. Other fields in the Channel Subframe Header list the number of channels and their site/channel/location identification.

The Channel Subframe contains a channel description, the current channel status, and the actual data. The data must be in a designated data type and may be compressed or uncompressed. The data are followed by an optional data authentication signature. The number of Channel Subframes must match the number specified in the Channel Subframe Header.

When uncompressed, data must be provided in network byte order; no translation is provided by transport processing.[1]

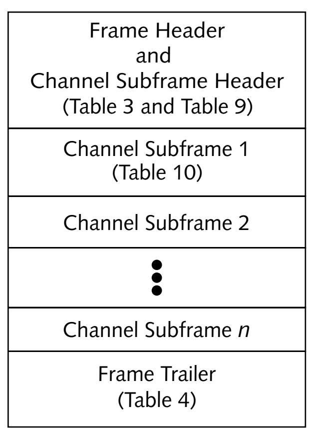


FIGURE 2. DATA FRAME COMPONENTS

1. The protocol allows the specification of Steim data compression; however, currently the Steim algorithm is not supported by IDC processing.

TABLE 9: DATA FRAME HEADER AND CHANNEL SUBFRAME HEADER

Field	Format	Description
frame header	see Table 3	(frame type = 5)
number of channels	IEEE integer	number of channels in this frame
frame time length	IEEE integer	time in milliseconds this frame encompasses
nominal time	20-byte ASCII	nominal UTC start time of all channels in frame; see "Time Synchronization" on page 43.
channel string count	IEEE integer	unpadded length in bytes of the channel string; must be ten times the number of channels field
channel string	N-byte ASCII	channel string listing the Channel Subframes to follow, 10 bytes per subframe The entire channel string is null-padded to a multiple of four bytes. Each 10-byte string is formatted as follows: five bytes for the site name left justified and padded with ASCII null (if necessary), three bytes for the channel name left justified and padded with ASCII null (if necessary), and two bytes for the location name left justified and padded with ASCII null (if necessary).[1] For example, the site name 'KCC' would be followed by two null characters before the specification of a channel name. Note that if only one channel of data is provided, then the channel description field must be padded with 2 null bytes (at the end) to satisfy the requirement of being evenly divisible by 4.

1. The protocol allows definition of a channel location; however, currently the IDC software does not read, use or store the location field.

TABLE 10: CHANNEL SUBFRAME OF DATA FRAME

Field	Format	Description
channel length	IEEE integer	length, in bytes and divisible by four of this Channel Subframe, not counting this integer
authentication offset	IEEE integer	byte offset from the first byte of the frame to the authentication key identifier
channel description	data bytes	flags and identifiers for this channel
	byte 1	authentication (0 = off; 1 = on)
	byte 2	transformation (0 = no transformation [see bytes 15–16 for data type]; 1 = Canadian compression applied before signature [bytes 15–16 irrelevant]; 2 = Canadian compression applied after signature [see bytes 15–16 for authenticated data type]; 3 = Steim compression applied before signature [bytes 15–16 irrelevant]; 4 = Steim compression applied after signature [see bytes 15–16 for authenticated data type]; other values as required)
	byte 3	sensor type (0 = seismic; 1 = hydroacoustic; 2 = infrasonic; 3 = weather; >3 = other)
	byte 4	option flag (0 = unused; 1 = calib and calper provided in bytes 17–24)
	bytes 5–9	site name; left justified, padded with ASCII null bytes as required
	bytes 10–12	channel name; left justified, padded with ASCII null bytes as required
	byte 13–14	location name; left justified, padded with ASCII null bytes as required
	bytes 15–16	uncompressed data format (CSS 3.0 data type) ASCII characters, set before signature if frame is signed. See Table 17: Data Types.
	bytes 17–20	CSS 3.0 calibration factor when byte 4 = `1` (IEEE float)
	bytes 21–24	CSS 3.0 calibration period when byte 4 = `1` (IEEE float)

TABLE 10: CHANNEL SUBFRAME OF DATA FRAME (CONTINUED)

Field	Format	Description
time stamp	20-byte ASCII	UTC start time for first sample of this channel
subframe time length	IEEE integer	time in milliseconds spanned by this channel data
samples	IEEE integer	number of samples in Channel Subframe
channel status size	IEEE integer	unpadded length in bytes of next field
channel status data	data bytes	status data for channel, padded to be divisible by four
data size	IEEE integer	unpadded length in bytes of next field
channel data	N data bytes	data for channel, padded to be divisible by four
subframe count	IEEE integer	subframe count as assigned by digitizer; zero for digitizers that do not support this count
authentication key identifier	IEEE integer	pointer to the certificate with the public key to be used for verifying the authentication value field
authentication size	IEEE integer	unpadded length in bytes of next field
authentication value	data bytes	DSS signature over the following fields: channel description, timestamp, subframe time length, samples, channel status size, channel status data, data size, channel data, and subframe count. This field is padded as necessary to be divisible by four.

ACKNACK FRAME

The Acknack Frame (Table 11) is provided for error control. It also delivers a heartbeat (to prevent timeout) if there are no Data Frames or other frames to send. Status fields include the name of the frame set being acknowledged, the range of valid sequence numbers, and a list of gaps in the sequence. A frame set is a container for the frames that will or have been exchanged with a protocol peer. Each frame set contains frames from a unique source. The gaps are presented in increasing order and must be non-overlapping intervals within the range of *lowest seq no* through *highest seq no*. When an Acknack Frame is sent for heartbeat/keep-alive purposes, its content is the same as at any other time, in other words, the list of sequence number gaps. Additional information on the use of this frame are provided in "Policies for Continuous Data Transmission" on page 39.

TABLE 11: ACKNACK FRAME

Field	Format	Description
frame header	see Table 3	(frame type = 6)
frame set acked	20-byte ASCII	full name of the frame set being acknowledged (for example, "SG7:0")
lowest seq no.	IEEE long	lowest valid sequence number sent considered during the current connection for the set (0 until a frame set is no longer empty)
highest seq no.	IEEE long	highest valid sequence number considered during the current connection for the set (–1 until a frame set is no longer empty)
gap count	IEEE integer	number of subsequent gaps
gap-begin-1	IEEE long	lowest missing sequence number
gap-end-1	IEEE long	next present sequence number
gap-begin-2	IEEE long	next lowest missing sequence number
gap-end-2	IEEE long	next present sequence number
...		
gap-begin-n	IEEE long	next lowest missing sequence number
gap-end-n	IEEE long	next present sequence number
frame trailer	see Table 4	

ALERT FRAME

Alert Frames are sent by either the data provider or consumer to notify the other party that the connection is going to be terminated. Table 12 defines the Alert Frame. Because of their very nature an Alert Frame will not be addressed by an Acknack Frame. For example, if an Alert Frame were to be re-sent, the receiving protocol peer would terminate the connection.

TABLE 12: ALERT FRAME

Field	Format	Description
frame header	see Table 3	(frame type = 7)
size	IEEE integer	unpadded length of message
message	N-byte ASCII	alert message, padded to be divisible by four, currently unspecified
frame trailer	see Table 4	

COMMAND REQUEST FRAME

A command requestor sends commands to stations (or to the system interfaces of partitioned subnetworks). These commands request actions such as "perform a calibration," "change the data authentication key," and so on. The Command Request Frame is provided as one possible mechanism to deliver commands for this purpose (Table 13).

The Command Request Frame has the standard header and trailer. Four fields indicate the entity being commanded. The method of specification is identical to the scheme used to label the originator of a Channel Subframe. The command message itself is allowed to be of variable type and length. The details are established by the policy of this protocol.

TABLE 13: COMMAND REQUEST FRAME

Field	Format	Description
frame header	see Table 3	(frame type = 8)
station name	8-byte ASCII	identifies IMS station
site/channel/loc name	data bytes	site/channel/location string
	bytes 1–5	site name; left justified padded with ASCII null bytes as required
	bytes 6–8	channel name; left justified padded with ASCII null bytes as required
	bytes 9–10	location name; left justified padded with ASCII null bytes as required
	bytes 11–12	required ASCII null padding bytes
time stamp	20-byte ASCII	UTC time of command origination
command message size	IEEE integer	unpadded length of next field, *command message*
command message	N-byte ASCII	actual command message, which is a variable number of ASCII bytes, padded to be divisible by four
frame trailer	see Table 4	

COMMAND RESPONSE FRAME

The Command Response Frame echoes the Command Request Frame (Table 14) and includes any returned data of the request. The data provider sends this frame back to the command initiator to confirm that the requested command has been properly received. The representation of responder station, sensor, and channel follow the same convention established for the Command Request Frame.

TABLE 14: COMMAND RESPONSE FRAME

Field	Format	Description
frame header	see Table 3	(frame type = 9)
responder station	8-byte ASCII	identifies IMS station
site/channel/loc name	data bytes	site/channel/location string
	bytes 1–5	site name; left justified padded with ASCII null bytes as required
	bytes 6–8	channel name; left justified padded with ASCII null bytes as required
	bytes 9–10	location name; left justified padded with ASCII null bytes as required
	bytes 11–12	required ASCII null padding bytes
time stamp	20-byte ASCII	UTC time of command completion
command request message size	IEEE integer	unpadded length of next field
command request message	N-byte ASCII	actual command request message, including header and trailer, for which the next three fields provide the response
response message size	IEEE integer	unpadded length of next field
response message	N-byte ASCII	command response message, which is a variable number of ASCII bytes, padded to be divisible by four
frame trailer	see Table 4	

CD-1 ENCAPSULATION FRAME

The CD-1 Encapsulation Frame is a special frame type (Table 15) that provides a means for handling frames produced by legacy software with newer CD-1.1 software. This frame type is only used by a data center; frames of this type are not expected from data providers. Encapsulated data consist of the CD-1 Channel Subframe described in the Formats and Protocols for CD-1 [IDC3.4.2]. The data are written to the *channel data* field of the CD-1.1 Channel Subframe (Table 10). One or more CD-1 Channel Subframes may be contained in a single CD-1.1 Data Frame. When multiple CD-1 Channel Subframes are provided, the subframes are included in the Data Frame as illustrated in Figure 2 on page 21. Each CD-1 Channel Subframe is encapsulated in its own CD-1.1 Channel Subframe within the overall Data Frame. The number of CD-1.1 Channel Subframes must match the number specified in the Data Frame Header. Data in the fields of the CD-1.1 Data Frame Header and Channel Subframe are provided from information in the CD-1 Data Format Frame, Data Frame Header, and Channel Subframe. Table 16 provides a mapping of CD-1 components to the CD-1 Encapsulation Frame.

TABLE 15: CD-1 ENCAPSULATION FRAME

Field	Format	Description
Data Frame Header	see Table 9	Data Frame Header as described in Table 9 with the exception that frame type = 13
Channel Subframe(s)	see Table 10	one or more Channel Subframes per frame according to the number of channels field of the Data Frame Header a CD-1 Channel Subframe is encapsulated in the channel data field of the CD-1.1 Channel Subframe definition (see Table 10)
frame trailer	see Table 4	

TABLE 16: CD-1 TO CD-1.1 DATA MAPPING

CD-1 Field	CD-1 Location	CD-1.1 Field	CD-1.1 Location
number of channels	Data Format Frame	number of channels	Data Frame Header
frame time length	Data Format Frame	frame time length	Data Frame Header
nominal time	Data Frame Header	nominal time	Data Frame Header

TABLE 16: CD-1 TO CD-1.1 DATA MAPPING (CONTINUED)

CD-1 Field	CD-1 Location	CD-1.1 Field	CD-1.1 Location
site and channel flags, calibrations, and names bytes 13–28	Data Format Frame	channel string count	Data Frame Header
site and channel flags, calibrations, and names bytes 13–28	Data Format Frame	channel string	Data Frame Header
packet length - provides size value, application to CD-1.1 must be computed	Channel Subframe	channel length	Channel Subframe
packet length - provides size value, application to CD-1.1 must be computed	Channel Subframe	authentication offset	Channel Subframe
site and channel flags, calibrations, and names byte 1, authentication	Data Format Frame	channel description byte 1, authentication	Channel Subframe
site and channel flags, calibrations, and names byte 2, compression	Data Format Frame	channel description byte 2, transformation	Channel Subframe
non-existent		channel description byte 3, sensor type Zero filled	Channel Subframe
non-existent		channel description byte 4, option flag 1 if calibration factor and period values are provided in bytes 17–24; 0 otherwise	Channel Subframe
site and channel flags, calibrations, and names bytes 13–28	Data Format Frame	channel description bytes 5–9, site name	Channel Subframe

TABLE 16: CD-1 TO CD-1.1 DATA MAPPING (CONTINUED)

CD-1 Field	CD-1 Location	CD-1.1 Field	CD-1.1 Location
site and channel flags, calibrations, and names bytes 13–28	Data Format Frame	channel description bytes 10–12, channel name	Channel Subframe
site and channel flags, calibrations, and names bytes 13–28	Data Format Frame	channel description bytes 13–14, location name	Channel Subframe
non-existent		channel description bytes 15–16, data type Set to CD for encapsulation	Channel Subframe
site and channel flags, calibrations, and names bytes 5–8	Data Format Frame	channel description bytes 17–20, calibration factor (if byte 4 = 1)	Channel Subframe
site and channel flags, calibrations, and names bytes 9–12	Data Format Frame	channel description bytes 21–24, calibration period (if byte 4 = 1)	Channel Subframe
time stamp	Channel Subframe	time stamp	Channel Subframe
frame time length	Data Format Frame	subframe time length	Channel Subframe
samples	Channel Subframe	samples	Channel Subframe
status	Channel Subframe	channel status size	Channel Subframe
status	Channel Subframe	channel status data	Channel Subframe
packet length	Channel Subframe	data size	Channel Subframe
Channel Subframe in its entirety	Channel Subframe	channel data	Channel Subframe
non-existent		subframe count set to 1	Channel Subframe

TABLE 16: CD-1 TO CD-1.1 DATA MAPPING (CONTINUED)

CD-1 Field	CD-1 Location	CD-1.1 Field	CD-1.1 Location
non-existent	Channel Subframe	authentication key identifier zero filled	Channel Subframe
non-existent	Channel Subframe	authentication size set to `40` if authentication is contained in CD-1 Channel Subframe	Channel Subframe
authentication	Channel Subframe	authentication value	Channel Subframe

Chapter 3: Data Formats

This chapter describes the data formats for continuous data exchange and includes the following topics:

- Introduction
- Uncompressed Format
- Canadian Compressed Format

Chapter 3: Data Formats

INTRODUCTION

Continuous time series data may be transmitted in either compressed or uncompressed format and may contain authentication signature information. Both formats transfer data without loss. However, compressed schemes reduce the cost of data transmission because fewer bits are required to transmit the same number of uncompressed data samples. Although the protocol allows specification of Steim compression, currently only Canadian decompression is supported at the IDC. Accordingly, only Canadian compression is addressed in the following paragraphs.

UNCOMPRESSED FORMAT

The uncompressed format specifies the representation of data samples according to the data format options. Data format options are given in Table 17. When a 4-byte IEEE integer representation is given, the representation implies forward byte ordering of all multibyte values. Values generated on machines that do not use IEEE as their native numerics will need to swap bytes before transmitting data.

TABLE 17: DATA TYPES

Data Type Name	Size in Bytes	Description
s4	4	Sun Microsystems IEEE integer (default)
s3	3	Sun Microsystems IEEE integer, packed
s2	2	Sun Microsystems IEEE short integer
i4	4	4-byte integer
i2	2	2-byte integer
CD	N/A	Encapsulated CD-1 data

CANADIAN COMPRESSED FORMAT

The Canadian compression scheme is used for continuous data transmission. Like many "lossless" compression schemes, it is based on an indexing of the second difference of the data. The Canadian compression scheme operates on blocks of 20 samples. If the number of samples is not a multiple of 20, then the time series should be padded to a multiple of 20 samples before compression with the extra samples discarded after decompression.

Difference Scheme

Let "S" be the sampled data so that S(1) is the first sample, S(2) is the second sample, and so on. The samples, S, are represented as 32-bit, 2's-complement integers, with a range of -2^{31} to $+2^{31}-1$. Any data sample whose value is unknown is given an integer value of -2^{31}.

Define "D" as the first difference between two subsequent samples so that:

$$D(2) = S(2) - S(1)$$
$$D(3) = S(3) - S(2)$$
$$\ldots$$
$$D(j) = S(j) - S(j-1)$$

Further define "D_2" as the second difference between two subsequent first difference samples so that:

$$\begin{aligned} D_2(j) &= D(j) - D(j-1) \\ &= S(j) - 2S(j-1) + S(j-2) \end{aligned}$$

A sequence of N+1 numbers consisting of S, D, and D_2 values constitutes a packet of data representing N data values:

$$S(1), D(2), D_2(3), D_2(4), \ldots, D_2(N+1)$$

or, in terms of S:

$$S(1), S(2) - S(1), S(3) - 2S(2) + S(1), \ldots, S(N+1) - 2S(N) + S(N-1)$$

N+1 samples are in the data block. The last sample in the sequence contains the value of the first sample in the next data packet and is used as an independent error check of the compression and decompression processes.

Index Scheme

The "N" difference values of the sequence derived above (the first sample is not a difference) are represented as 4, 6, 8, 10, 12, 14, 16, 18, 20, 24, 28, or 32-bit, 2's-complement integers with ranges from –8 to +7 for four bits, –32 to +31 for six bits, and so on, up to -2^{31} to $+2^{31}-1$ for 32 bits.

An index is used to specify how many bits are used to encode the difference values of the sequence. The index contains a number of 16-bit entries, one for each block of 20 differences. Thus, a 10-second data packet sampled at 40 samples per second will have an index containing 20 entries totaling 40 bytes.

Each index entry encodes the bit lengths of the difference values of the sequence in groups of four as shown in Table 18.

TABLE 18: 16-BIT INDEX

Most Significant Bit													Least Significant Bit		
h	a	a	a	b	b	b	c	c	c	d	d	d	e	e	e

- h = 0

  All 20 samples can be encoded in ≤ 18 bits per sample.

  h = 1

  At least one sample needs ≥ 20 bits for encoding.

- aaa

  These three bits encode the number of bits per sample required to specify the first four data sample differences (for example, for samples 2 through 5); its interpretation depends on whether h = 0 or 1.

- bbb

  These three bits encode the number of bits per sample for samples 6 through 9.

- ccc

  These three bits encode the number of bits per sample for samples 10 through 13.

- ddd

  These three bits encode the number of bits per sample for samples 14 through 17.

- eee

  These three bits encode the number of bits per sample for samples 18 through 21.

Table 19 shows the number of bits assigned to samples based on the 3-bit sample codes and the value of h.

TABLE 19: BIT LENGTHS FOR VARIOUS SAMPLE CODES

	Sample Bit Length	
3-bit Sample Code	**h = 0**	**h = 1**
000	4	4
001	6	8
010	8	12
011	10	16
100	12	20
101	14	24
110	16	28
111	18	32

Transmission of Compressed Data Blocks

A time series is compressed as one or more index entries and one or more compressed data blocks. All index entries are transmitted in order, followed by all compressed data blocks in order. Each index entry is a 16-bit short integer and is sent in network byte order. Each compressed data block is a bit string and is sent with no padding between successive compressed data blocks. Each compressed data block contains 20 integers of varying lengths, not necessarily aligned on byte boundaries. Each of these 20 integers is ordered within the bit string with its high-order bits toward the start of the string and low-order bits toward the end of the string.

Chapter 4: Policies for Continuous Data Transmission

This chapter describes the policies for continuous data transmission and includes the following topics:

- Field Values
- Descriptive Information
- Size and Time Limits
- Grouping and Ordering
- Other Policies
- Option Request Options
- Channel Status Data Field of Channel Subframe

Chapter 4: Policies for Continuous Data Transmission

FIELD VALUES

The valid values for key fields of the protocol are indicated in Table 20.

TABLE 20: VALID VALUES FOR PROTOCOL FIELDS

Fields	Table	Type	Default	Values	Example
frame creator	3, 11	ASCII	required	ASCII alpha-numeric string, left justified, padded with ASCII nulls; first character must be alphabetic	SG
frame destination	3	ASCII	required	ASCII alpha-numeric string, left justified, padded with ASCII nulls; first character must be alpha-numeric	IDC 0 if destination is indeterminate
authentication key identifier	4, 10	integer	0	integer assigned by CTBTO	1
comm verification	4	long	required	per polynomial algorithm	N/A
major version	5, 6	short	0	see "Connection Request Frame" on page 16	1

TABLE 20: VALID VALUES FOR PROTOCOL FIELDS (CONTINUED)

Fields	Table	Type	Default	Values	Example
minor version	5, 6	short	0	See "Connection Request Frame" on page 16	1
station name, *responder name*	5, 6, 13	ASCII	required	ASCII alpha-numeric string, left justified, padded with ASCII nulls; first character must be alphabetic	WRA, IDC
station type, *responder type*	5, 6	ASCII	required	IDC, IMS, or NDC	IMS
service type	5, 6	ASCII	required	TCP or UDP	TCP
IP address	5, 6	unsigned integer	required	valid 32-bit IP address	12345678
port	5, 6	integer	required	0-65535 value of zero only valid in Connection Request Frame	13282
nominal time, *time stamp*	9, 13	ASCII	required	any correct time in the format yyyyddd hh:mm:ss.mmm	2000012 19:33:06.000 (space, not new line)
series	3	integer	0	0, positive integers. No longer used	0
site name	10, 13, 14	ASCII	required	ASCII alpha-numeric string, left justified, padded with ASCII nulls; first character must be alpha-numeric	WB2

TABLE 20: VALID VALUES FOR PROTOCOL FIELDS (CONTINUED)

Fields	Table	Type	Default	Values	Example
channel name	10, 13, 14	ASCII	required	FDSN channel name	BHZ
location name	10, 13, 14	ASCII	NULL	FDSN location name	1
data type	10	ASCII	required	see Table 17 on page 34	s4

DESCRIPTIVE INFORMATION

These policies pertain to ancillary information not defined by the protocol but required by application software.

Static Configuration Data

Descriptive information is required to configure software to receive and process continuous data. These data will be provided separately from this protocol. The exact list is not specified, but will include the IP address of the sending computer, station codes, instrument frequency response, and so on.

Unknown Station, Site, and Channel Names

Data that are received with station, site, or channel codes that have not been included in the static configuration database are discarded.

SIZE AND TIME LIMITS

These policies pertain to constraints on Data Frames.

Maximum Number of Channels

Up to 100 channels can be transmitted as part of a single Data Frame.

Time Synchronization

All Channel Subframes of a Data Frame will have the same start time and duration to within one sample time unit. There is one Channel Subframe for each channel in a Data Frame. In addition, policy requires the first sample to be taken on an integral multiple of 10 seconds and synchronized to the minute.

Under PTS waiver, asynchronous Channel Subframes can be contained in a single Data Frame under the following conditions: (1) The nominal time field is set equal to the start time of the earliest Channel Subframe, and (2) no Channel Subframe has a start time later than the nominal time plus the frame time length.

Duration and Size of Channel Subframe

The IMS requires that Data Frames contain ≤ 10 seconds of data for seismic (short-period and broad-band) and hydroacoustic sensors and ≤ 30 seconds for infrasonic and seismic long-period data. However, to avoid excessive authentication processing load and frame processing overhead, it is recommended that frames contain exactly 10 seconds or 20 seconds of data, respectively. Because each channel in a Data Frame has one Channel Subframe, these constraints apply to the Channel Subframe as well.

For illustrative purposes, it is interesting to see the sizes of the various elements of a Channel Subframe. Table 21 shows the approximate sizes, in bytes, for the stated conditions, using 40 sps, 240 sps, and 10 sps for the respective instrument types, and assuming a 3-byte digitizer and 50 percent reduction from data compression.

TABLE 21: REPRESENTATIVE SIZES OF CHANNEL SUBFRAMES

Fields	Seismic	Hydroacoustic	Infrasonic
data	600	3,600	450
signature	60	60	60
description and status	84	84	84
subframe total	744	3,744	594

Applications should be able to handle 100-second Channel Subframes.

Duration and Size of Data Frame

The duration of the Data Frame will be the same as that of the Channel Subframes, described in the previous section. The total size of the Data Frame will be the sum of the sizes of the Channel Subframes plus approximately 200 bytes for the frame header and trailer. In the case of a four-element infrasonic array, the total size is approximately 2,600 bytes. In the case of a 10-element seismic array, the total size is approximately 7,940 bytes.

The IMS requirements on frame duration hold for continuous transmission of real time data from the seismic, hydroacoustic, and infrasonic instruments. This condition does not apply in three situations: (1) the transmission of old data, which may accumulate during an outage; (2) the transmission of ancillary data, such as weather readings, and (3) transmission of data segments, if policy permits such use of the protocol. Policy has not been established for these cases.

GROUPING AND ORDERING

This section describes policies related to the optional grouping of Channel Subframes into Data Frames and ordering of transmission of frames.

Mixed Sample-rate Data

Channel Subframes from a single station and with different sample rates may be included in a single Data Frame, as long as all other policies are satisfied. For seismic arrays, this policy requires inclusion of broad-band and short-period data in the same Data Frame.

Missing Data Channels

When Channel Subframe data are not available, the Channel Subframe shall be dropped from the Data Frame. When the data subsequently become available, they will be sent on a bandwidth-available basis and in reverse chronological order in a new set of Data Frames. Channel Subframes previously sent will not be retransmitted.

Missing Data Frames

Data Frames with sequence numbers that have not been received by a data consumer will be reported to a data provider in Acknack Frames, and the data provider will retransmit these Data Frames using the original sequence numbers.

Data Ordering

During normal operations, data are sent from the provider to the consumer as soon as they become available. When service has been interrupted, the policy is to send the most recent data first and to transmit older data, if any accumulated during the outage, as bandwidth permits. Older data will be transmitted in reverse time order. This data ordering scheme is referred to as Last In, First Out (LIFO).

Data Provider Buffering

The data provider must retain all frames of the protocol until an Acknack Frame is received, indicating that the data have been safely received and stored. In some cases, data storage limits at the data provider may cause frames to be discarded prior to acknowledgement from the destination.

OTHER POLICIES

This section describes miscellaneous policies.

Refusal of Connection Request

If the data consumer denies a connection request from a data provider, no response is returned to the requester.

Station and Site Names

The station name identifies the location of a sensor site or a collection of sensor sites. The station name for seismic arrays is the beam point for signal processing. The site name is the location of a specific sensor.

Seismic station and site names shall be registered with the International Seismic Centre (ISC) or the National Earthquake Information Center (NEIC) of the U.S. Geological Survey. Seismic station and site names shall conform to standards given in [USG85]. For example, station names shall be three to five characters in length, and therefore the station field will never be completely filled. No counterpart registry exists for hydroacoustic and infrasonic station or site names, but station and site name formats shall conform to IMS station naming conventions [IMS01].

Channel and Location Names

The channel name is the three-character string that labels the bandwidth, gain, and orientation of the sensor. Channel names for seismic stations shall conform to the standards of the Federation of Digital Seismograph Networks (FDSN) [IRI93]. Channel names that are not defined by the FDSN shall conform to IMS channel naming conventions [IMS01].

The location field is provided in anticipation of the need to distinguish between two or more sensors at one site with the same channel name. Although a location identifier is included in the SEED format [IRI93], no convention has been defined for location naming.

Disconnection Due to Time-out

A time-out is defined to be an elapsed interval of time when no Acknack Frames have been sent. The time-out is provisionally defined to be 2.5 times the heartbeat interval. Under these conditions both the provider and consumer will disconnect. This policy provides a well-defined failure mode in the event of an unrecoverable fault in either the hardware or application software at either end of the link.

Acknack Frames and Heartbeat Interval

Acknack Frames provide heartbeat signals. The provisional policy is that Acknack frames will be sent at least once a minute. Acknack Frames employ the standard frame header (Table 3). However, the header sequence number is undefined in the Acknack Frame. Further, Acknack Frames do not report on the delivery of other Acknack Frames. This would not be possible because Acknack frames have no sequence number.

Long Outages of Communication

When a prolonged interruption of service causes imminent overflow of the data provider's data buffer, data shall be written on physical media and delivered to the data consumer by other means.

Use of Command Frames

Command Frames are available to pass data that affect or pertain to the operational condition of a station. An example would be a calibration request. The ability to submit commands is restricted to a privileged set of users.

Command Requests and Command Responses are of interest to all receivers of the data, hence they must be reflected in the data stream.

Authentication of Data

The policy allows compression of data to be applied either before or after authentication signatures are calculated. The uncompressed data format used for signature calculation must be the data type specified in the Channel Subframe.

Protocol Version

The protocol version to be used shall be the highest version jointly supported by the dp and dc. Minor versions of the protocol must be compatible. If a new version is incompatible with previous version, then it must be assigned a new major version number.

OPTION REQUEST OPTIONS

Options of the protocol should be defined by a managed change process as needs are identified. This change process has yet to be implemented. The current list of options is as follows:

- Connection establishment

Option 1. Connection

This option supports establishing an initial connection and identifies the site requesting the connection in the *request* field. This option must be sent to create a connection and may not accompany any other options. The sequence of frames for each successful connection would be:

- dp: Connection Request Frame
- dc: Connection Response Frame
- dp: Option Request Frame - with option 1
- dc: Option Response Frame - with option 1

The location requesting the connection identifies itself by providing its 8 byte ASCII string name in the option 1 request field, i.e., the same string provided in the Connection Request Frame for the station name field. Given the transitory nature of the option 1, Option Request Frame, sequence numbers are not assigned to these frames.

CHANNEL STATUS DATA FIELD OF CHANNEL SUBFRAME

A variety of structures have been proposed for the *channel status data* field of the Channel Subframe (Table 10 on page 23). To allow several definitions of this field, the definition of its format does not include type or structure.

The policy of this protocol is to support the structure defined in Table 22. Other *channel status data* fields may be added through a managed change process.

First Permitted Status Field

Table 22 indicates a valid structure for the 32-byte *channel status data* field, which is an element of every Channel Subframe. All eight flags defined in the CD-1 protocol are represented.

TABLE 22: CHANNEL STATUS FIELD

Field	Format	Description
channel status	8-byte ASCII	data and station status as follows:
	byte 1	format of channel status field. 1 = this format
	byte 2	data status byte: bit 1 1 = dead sensor channel bit 2 1 = zeroed data bit 3 1 = clipped bit 4 1 = calibration underway bits 5–8 future use
	byte 3	channel security byte: bit 1 1 = equipment housing open bit 2 1 = digitizing equipment open bit 3 1 = vault door opened bit 4 1 = authentication seal broken bit 5 1 = equipment moved bits 6–8 future use
	byte 4	miscellaneous status byte: bit 1 1 = clock differential too large bit 2 1 = GPS receiver off bit 3 1 = GPS receiver unlocked bit 4 1 = digitizer analog input shorted bit 5 1 = digitizer calibration loop back bits 6–8 future use
	byte 5	voltage indicator byte bit 1 1 = main power failure bit 2 1 = backup power unstable bits 3–8 future use
	bytes 6–8	undefined
	20-byte ASCII	time of last GPS synchronization
	IEEE integer	clock differential in microseconds

References

The following sources are referenced in or supplement this document:

[Bow98] Bowman, R., W. Fox, *Proposed Changes to Formats and Protocols for Continuous Data*, CMR-98/01, January 1998.

[GSE95a] Group of Scientific Experts, *Formats and Protocols,* in GSETT 3 Documentation, Volume Two: Operations, Annex III, Chapter 7, CRP/243, 1995.

[IDC3.4.2] Science Applications International Corporation, *Formats and Protocols for Continuous Data,* SAIC-97/3022, 1997.

[IDC3.4.3] Science Applications International Corporation, Formats and Protocols for Continuous Data CD-1.1, SAIC-00/3026, 2000.

[IDC3.4.3Rev0.1] Science Applications International Corporation, Formats and Protocols for Continuous Data CD-1.1 Revision 0.1, SAIC-01/3027, 2001.

[IMS01] International Monitoring System Division, Operations and Maintenance Group, IMS-OM-2001-05, 2001.

[IRI93] Incorporated Research Institutes for Seismology, *Standards for the Exchange of Earthquake Data (SEED)*, Reference Manual SEED Format Version 2.3, February, 1993.

[ISO88] International Standards Organization, *ISO 8601:1988, Date/Time Representations*, Geneva, 1988.

[NIS98] National Institute of Standards and Technology, *Digital Signature Standard (DSS)*, Federal Information Processing Standard Publication FIPS 186-1, U.S. Department of Commerce, December 30, 1996.

[SAI98a] Science Applications International Corporation, *Minutes of an Informal Workshop on Formats and Protocols for Continuous Data*, CMR-98/08, February 1998.

[SAI98b] Science Applications International Corporation, *Proposed New Formats and Protocols for Continuous Data*, SAIC-98/3029, July 1998.

[SAI00] Science Applications International Corporation, *Continuous Data Subsystem for CD-1.1: Requirements and Architectural Description*, SAIC and SAIC/Telcordia, February 2000.

[Sar88] Sarwate, D. V., "Computation of Cyclic Redundancy Checks via Table Look-up," *Communication of the ACM*, Volume 31, No. 8, pp. 1008–1013, August 1988.

[USG85] U.S. Geological Survey, *Seismograph Codes and Coordinates*, Open File Report 85-719, 1984
online: `http://wwwneic.cr.usgs.gov/neis/station_book/background.html`

Glossary

A

ASCII

American Standard Code for Information Interchange. Standard, unformatted 256-character set of letters and numbers. (Eight-bit code representing alphanumeric characters.)

authentication signature

A series of bytes that are unique to a set of data and that are used to verify the authenticity of the data.

C

CMR

Center for Monitoring Research.

CPU

Central Processing Unit.

CSP

Conference of States Parties; the principal body of the CTBTO consisting of one representative from each State Party accompanied by alternate representatives and advisers. The CSP is responsible for implementing, executing, and verifying compliance with the Treaty.

CSS 3.0

Center for Seismic Studies (CSS) version 3 database schema, including a format for storing time-series data in disk files and database descriptors of that data.

CTBT

Comprehensive Nuclear-Test-Ban Treaty (the Treaty).

CTBTO

Comprehensive Nuclear-Test-Ban Treaty Organization; Treaty User group that consists of the Conference of States Parties (CSP), the Executive Council, and the Technical Secretariat.

D

DSA

Digital Signature Algorithm.

DSS

Digital Signature Standard.

E

email

Electronic mail.

Executive Council

Executive body of the CTBTO responsible for supervising the activities of the Technical Secretariat.

F

FDSN

Federation of Digital Seismic Networks.

FIFO

First In, First Out.

Frame Set

Data store defined for each source and destination pair for CD-1.1 data. Each frame set has an associated frame log.

G

GPS

Global Positioning System.

GSETT-3

Group of Scientific Experts Third Technical Test.

H

Hz

Hertz.

I

ID

Identification.

IDC

International Data Centre.

IEEE

Institute for Electrical and Electronic Engineers.

IMS

International Monitoring System.

IP

Internet protocol.

K

K

Kilobyte. 1,024 bytes.

L

LIFO

Last In, First Out.

M

MB

Megabyte. 1,024 kilobytes.

N

N/A

Not Applicable.

NDC

National Data Center.

NEIC

National Earthquake Information Center.

nm

Nanometers.

P

PIDC

Prototype International Data Centre.

PTS

Provisional Technical Secretariat

Provisional Technical Secretariat

Part of the Comprehensive Nuclear-Test-Ban-Treaty Organization, consisting of divisions responsible for the International Data Centre, International Monitoring System, On-site Inspection, Legal and External Relations, and Administration.

S

SAIC

Science Applications International Corporation.

T

TCP/IP

Transmission Control Protocol/Internet Protocol.

Treaty

Comprehensive Nuclear-Test-Ban Treaty (CTBT).

U

UDP

User Datagram Protocol.

UTC

Universal Coordinated Time.

Index

V