[MS-PPPI]:
PPP Over IrDA Dialup Protocol Specification

Revision Summary

Date	Revision History	Revision Class	Comments
05/11/2007	0.1		MCPP Milestone 4 Initial Availability
08/10/2007	1.0	Major	Updated and revised the technical content.
09/28/2007	1.0.1	Editorial	Revised and edited the technical content.
10/23/2007	1.0.2	Editorial	Revised and edited the technical content.
11/30/2007	1.0.3	Editorial	Revised and edited the technical content.

Date	Revision History	Revision Class	Comments
01/25/2008	1.0.4	Editorial	Revised and edited the technical content.

Table of Contents

1 Introduction

This document defines the PPP Over IrDA Dialup Protocol, and includes contributions from Microsoft, Ericsson, and Nokia. The PPP Over IrDA Dialup Protocol is a deprecated protocol that specifies how to initialize and use a modem over an infrared link. This protocol enables the scenario in which a computer with infrared capabilities has network access by using a modem by way of the infrared link.

1.1 Glossary

The following terms are defined in [MS-GLOS]:

Client
Server

The following terms are specific to this document:

<CR>: ASCII character "Enter" (decimal symbol # 13), as defined in [RFC1345].

<LF>: ASCII character "Line feed" (decimal symbol # 10), as defined in [RFC1345].

AT Command Set: The Attention Code (AT) command set is used to issue one or more commands to a modem device. The AT prefix signals the modem that one or more commands are to follow as specified in [V25TER].

Data-Carrying TinyTP PDU: The **TinyTP** Protocol distinguishes between **PDUs** that carry data and **PDUs** that are used during **TinyTP** connection establishment. The 'Data-Carrying' prefix denotes that the **PDU** being discussed is used for data exchange and not **TinyTP** connection establishment.

Device Address: The **IrLAP device address** of a station. This is a 32-bit identifier that is randomly selected by a station. It is expected to be relatively static between successive initializations of the **IrLAP** communication services. See [IRLMP] for more details.

Escape sequence: A series of three consecutive characters (+++) sent to the infrared modem causing it to exit online data mode and enter online command mode.

Information Access Service (IAS): Each device that implements the set of infrared protocols, specifically [IRLMP], maintains an information base so that one **IrDA** device can discover what services another IrDA-compliant device offers, as well as gain information about the device itself. This information is held in a number of objects in the information base and is accessed by communicating with the **IAS**.

IrDA: The Infrared Data Association, often referred to as **IrDA**, is a nonprofit organization whose goal is to develop globally adopted specifications for infrared wireless communication.

IrDial: A component of the PPP Over IrDA Dialup Protocol that implements modem commands.

IrLAP: An acronym for the IrDA-defined data-link layer protocol: Infrared Link Access Protocol. See [IRLAP] for more details.

IrNet: A component of the PPP Over IrDA Dialup Protocol that implements a data path pass-through between the PPP over HDLC-like framing and **TinyTP**.

Link Service Access Point Selector (LSAP-SEL): A selector that distinguishes between LSAPs within a station. Legal values for an **LSAP-SEL** lie in the range 0x00-0x7F. With the exception of the special **LSAP-SEL** values 0x00 (LM-IAS), 0x70 (Connectionless Data service), 0x71-

0x7E (reserved), and 0x7F (reserved for broadcast and currently not implemented), the assignment of **LSAP-SEL** values is arbitrary. See section 3.1.2 in [IRLMP] for more details.

PDU: A **TinyTP** Protocol Data Unit as specified in [IRTTP] section 2.3.1. The maximum size of the **PDU** must be in the range of 64 Bytes and 2048 Bytes. See section 6.6.5 in [IRLAP] and [IRTTP] section 2.3.1 for more details.

SDU: A **TinyTP** Client Service Data Unit as defined in [IRTTP] section 2.2.3. The **SDU** specifies the maximum size of a data block that can be exchanged between two **TinyTP clients**. For example, the IrDial Protocol, which is implemented as **TinyTP client**, MAY negotiate an **SDU** size of 10 KB. The **TinyTP** Protocol segments and reassembles this **SDU** in multiple **PDUs** which are transmitted over the infrared link.

TinyTP: The Infrared Data Association Tiny Transport Protocol.

TTPSAP: A **TinyTP** Service Access Point address is a selector that distinguishes between different **TinyTP client** connections. **TTPSAP** is always equal to the <Device Address><LSAP-SEL>. See section 2.1 in [IRTTP] for more details.

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as described in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

[IRLAP] Infrared Data Association, "IrDA Link Access Protocol v.1.1", June 1996, http://irda.org/displaycommon.cfm?an=1&subarticlenbr=7

Note There is a charge to download the specification.

[IRLMP] Infrared Data Association, "IrDA Link Management Protocol v1.1", January 1996, http://irda.org/displaycommon.cfm?an=1&subarticlenbr=7

Note There is a charge to download the specification.

[IRTTP] Infrared Data Association, "IrDA Tiny TP v1.1", October 1996, http://irda.org/displaycommon.cfm?an=1&subarticlenbr=7

Note There is a charge to download the specification.

[MS-GLOS] Microsoft Corporation, "Windows Protocols Master Glossary", March 2007.

[RFC1345] Simonsen, K., "Character Mnemonics and Character Sets", RFC 1345, June 1992, http://www.ietf.org/rfc/rfc1345.txt

[RFC1661] Simpson, W., Ed., "The Point-to-Point Protocol (PPP)", RFC 1661, July 1994, http://www.ietf.org/rfc/rfc1661.txt

[RFC1662] Simpson, W., Ed., "PPP in HDLC-like Framing", RFC 1662, July 1994, http://www.ietf.org/rfc/rfc1662.txt

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.ietf.org/rfc/rfc2119.txt

[V25TER] ITU-T, "Serial Asynchronous Automatic Dialling and Control", Recommendation V.25ter, July 1997, http://www.itu.int/rec/T-REC-V.25ter-199707-S/en

Note There is a charge to download the specification.

[V42BIS] ITU-T, "Data Compression Procedures for Data Circuit-Terminating Equipment (DCE) Using Error Correction Procedures", 1990, http://www.itu.int/rec/T-REC-V.42bis-199001-I/en

Note There is a charge to download the specification.

1.2.2 Informative References

[IRPLS] Infrared Data Association, "IrDA Physical Layer Specification v.1.4", May 2001, http://irda.org/displaycommon.cfm?an=1&subarticlenbr=7

Note There is a charge to download the specification.

[IRCOM] Infrared Data Association, "'IrCOMM': Serial and Parallel Port Emulation over IR (Wire Replacement) v.1.0", November 1995, http://irda.org/displaycommon.cfm?an=1&subarticlenbr=7

Note There is a charge to download the specification.

[MSDN-IRNET] Microsoft Corporation, "Infrared Network (IrNET)", http://msdn2.microsoft.com/en-us/library/ms817914.aspx

1.3 Protocol Overview (Synopsis)

The PPP Over IrDA Dialup Protocol specifies how to initialize and exchange data with a modem over an infrared link. The PPP Over IrDA Dialup Protocol has a **client** and a **server** role:

- The server is the modem that has an infrared device and implements the PPP Over IrDA Dialup Protocol. In addition, the server MUST implement the family of **IrDA** protocols, specifically: IrDA TinyTP v1.1 Protocol (as specified in [IRTTP]), IrDA LinkManagement Protocol v1.1 (as specified in [IRLMP]), IrDA LinkAccess Protocol v1.1 (as specified in [IRLAP]), and IrDA Physical Layer Specification (as specified in [IRPLS]).
- The client is the computer that implements the PPP Over IrDA Dialup Protocol. In addition the client MUST implement the family of IrDA protocols, specifically: IrDA TinyTP v1.1 Protocol (as specified in [IRTTP]), IrDA LinkManagement Protocol v1.1 (as specified in [IRLMP]), IrDA LinkAccess Protocol v1.1 (as specified in [IRLAP]), and IrDA Physical Layer Specification (as specified in [IRPLS]).

To initialize the modem, the PPP Over IrDA Dialup Protocol uses **IrDial** messages as described in section 2.2. Briefly, the IrDial messages are AT dial commands as specified in [V25TER], which are exchanged between the computer and the modem via the infrared link.

Each IrDial message is echoed back to the client by the server. Note that the echoing of messages back to the client is applicable only while the PPP Over IrDA Dialup Protocol is initializing the modem. When the modem is used for data exchange there are no message echoes.

After initializing the modem, the PPP Over IrDA Dialup Protocol can exchange data with it. To exchange data with the modem the PPP Over IrDA Dialup Protocol uses the Point-to-Point Protocol (PPP) in HDLC-like framing to encapsulate the higher-layer network data as specified in [RFC1662]. The PPP in HDLC-like framing frames are segmented (if necessary) and framed in **TinyTP** Service Data Units (**SDUs**) as specified in [IRTTP].

1.4 Relationship to Other Protocols

A preferred alternative to the PPP Over IrDA Dialup Protocol is the IrDA Infrared Communications Protocol (IrCOMM), as specified in [IRCOM], which provides emulation of serial and parallel ports over the IrDA TinyTP v1.1 Protocol (as specified in [IRTTP]), IrDA LinkManagement Protocol v1.1 (as specified in [IRLMP]), and IrDA LinkAccess Protocol v1.1 (as specified in [IRLAP]) protocol stack. By providing serial port emulation, the computer can initialize and use the modem over an infrared link by simply reading and writing to a serial port.

The PPP Over IrDA Dialup Protocol depends on the Infrared Network (IrNET) Protocol, as specified in [MSDN-IRNET], for the data path pass-through of PPP in HDLC-like framing frames to the TinyTP layer. Figure 1 depicts the PPP Over IrDA Dialup Protocol and its relationship to the other higher-layer and lower-layer protocols mentioned in this document.

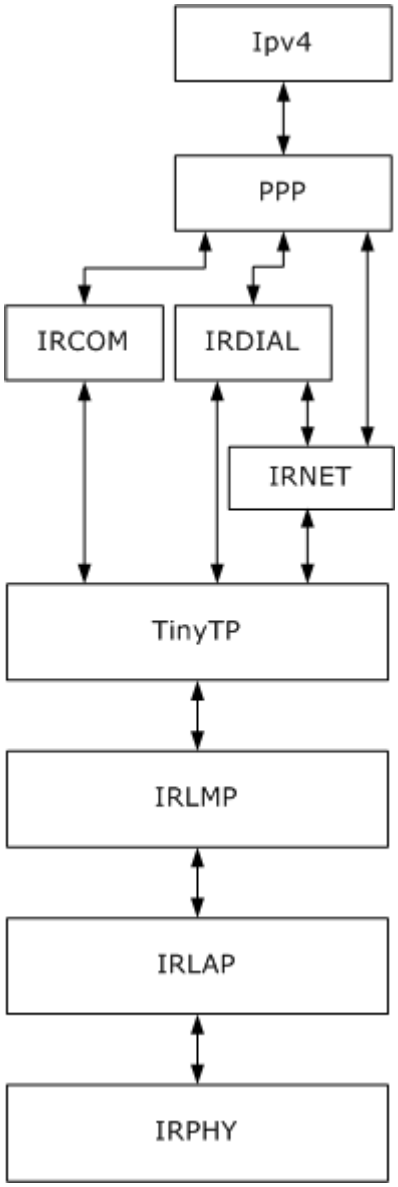


Figure 1: Protocol stack diagram showing PPP Over IrDA Dialup Protocol, which consists of the IRDIAL and IRNET components and their relationship to other protocols. Both IRDIAL and IRNET use TinyTP as their transport protocol. Note: the PPP component includes both [RFC1661] and [RFC1662] encapsulations.

1.5 Prerequisites/Preconditions

Prior to using the PPP Over IrDA Dialup Protocol, the client and server must have an established TinyTP connection. See sections 3.2.3 and 3.3.3 for further details.

1.6 Applicability Statement

The PPP Over IrDA Dialup Protocol is deprecated and should not be used. Instead, the IrDA Infrared Communications Protocol (IrCOMM), as specified in [IRCOM], should be used for modem initialization and data transfer over an infrared link.

1.7 Versioning and Capability Negotiation

The PPP Over IrDA Dialup Protocol does not implement any versioning or capability negotiations.

1.8 Vendor-Extensible Fields

The PPP Over IrDA Dialup Protocol does not introduce or rely on any vendor-extensible fields.

1.9 Standards Assignments

The PPP Over IrDA Dialup Protocol does not introduce any new standard fields.

2 Messages

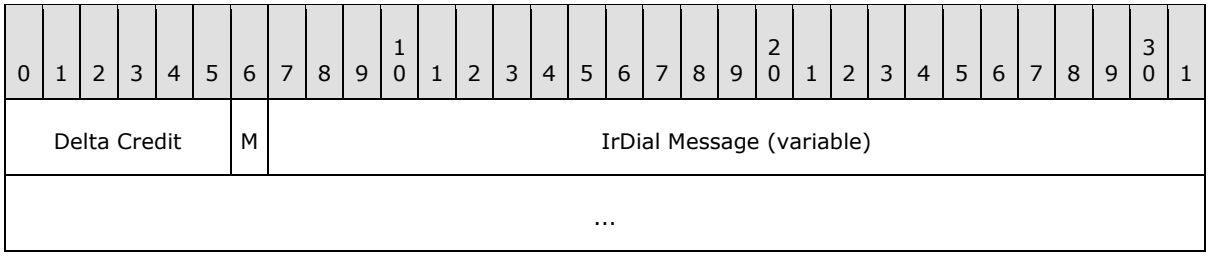
The following sections specify how PPP Over IrDA Dialup Protocol messages are transported and message syntax.

2.1 Transport

The PPP Over IrDA Dialup Protocol uses IrDA Tiny TP, as specified in [IRTTP], as a transport protocol for initialization and data exchange with the modem. Note that for data exchange with the modem, [RFC1662] frames are transported directly via IrDA Tiny TP, as specified in [IRTTP].

2.2 Message Syntax

As specified in [IRTTP] section 2.2.1, the maximum size of a single Data-Carrying TinyTP PDU packet for a given connection is negotiated by the IrDA Link Access Protocol, as specified in [IRLAP].



Delta Credit (6 bits): The **Delta Credit** field MUST specify the number (0-127) of additional **Data-Carrying TinyTP PDUs** that may be sent in the reverse direction, as specified in section 2.3.1 of [IRTTP].

M (1 bit): The **M** field MUST be set to zero, as specified in section 2.3.1 of [IRTTP], to denote that this is a single TinyTP **PDU**.

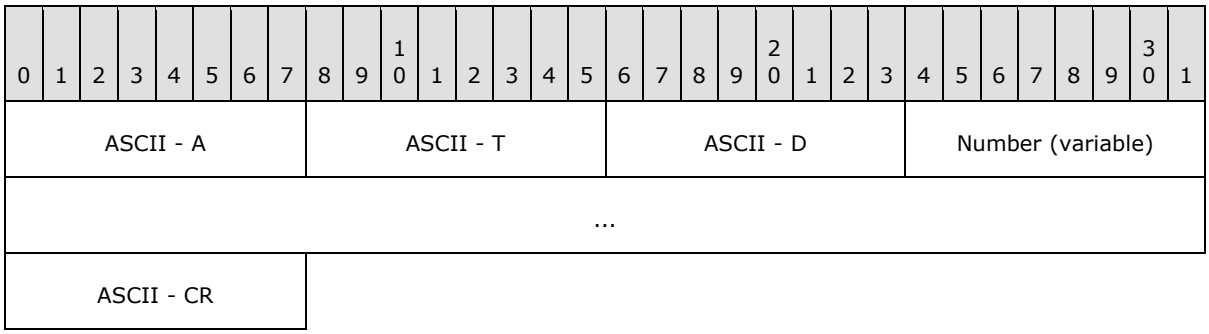
IrDial Message (variable): Each IrDial message is a sequence of ASCII characters and MUST conform to the definitions in section 2.2.1. Messages of this form are often referred to as **AT Commands**. The IrDial message MUST NOT exceed the maximum size of a TinyTP PDU, which is negotiated by the IrDA Link Access Protocol (as specified in [IRLAP]). See [IRTTP] section 2.2.1 for more details.

2.2.1 IrDial Message Formats

2.2.1.1 General Formatting Rules

- Each IrDial message MUST end with **<CR>**.
- Each IrDial message that is a response to a previous IrDial message MUST be in the format <CR><LF>Response<CR><LF>.

2.2.1.2 Dial Message



ASCII - A (1 byte): The ASCII encoding for the 'A' character, 0x41, as specified in [RFC1345].

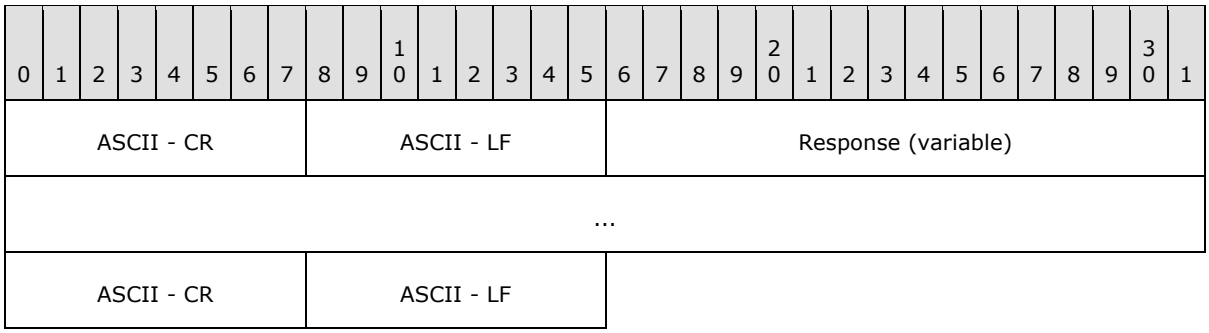
ASCII - T (1 byte): The ASCII encoding for the 'T' character, 0x54, as specified in [RFC1345].

ASCII - D (1 byte): The ASCII encoding for the 'D' character, 0x44, as specified in [RFC1345].

Number (variable): A sequence of ASCII encoded numbers, as specified in [RFC1345]. This number denotes the phone number that the modem will dial.

ASCII - CR (1 byte): The ASCII encoding for the carriage return character, 0x0d, as specified in [RFC1345].

2.2.1.3 Dial Response Message



ASCII - CR (1 byte): The ASCII encoding for the carriage return character, 0x0d, as specified in [RFC1345].

ASCII - LF (1 byte): The ASCII encoding for the line feed character, 0x0a, as specified in [RFC1345].

Response (variable): The **Response** field MUST be ASCII encoded string. The **Response** field MUST have one of the following values:

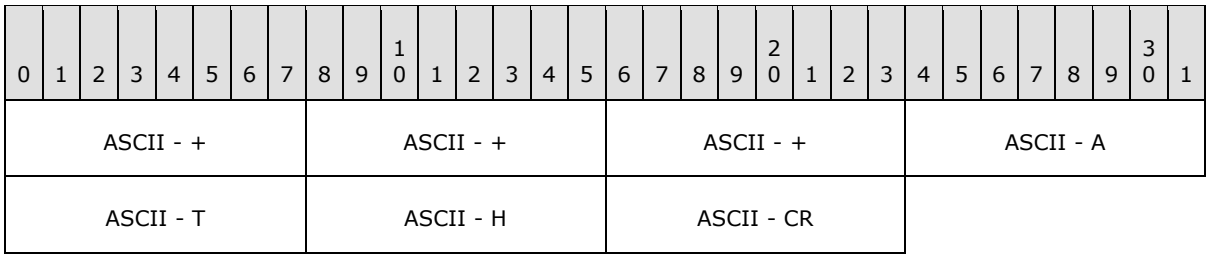
Value	Meaning
"CONNECT <speed>"	Data connection established at the rate given in <speed>.
"NO CARRIER"	Unable to establish a connection, or the connection attempt was aborted.

Value	Meaning
"ERROR"	An unexpected error occurred while trying to establish the connection.
"NO DIALTONE"	The mobile phone is being used for a voice call or is not within coverage of the network.
"BUSY"	The phone number called is engaged.

ASCII - CR (1 byte): The ASCII encoding for the carriage return character, 0x0d, as specified in [RFC1345].

ASCII - LF (1 byte): The ASCII encoding for the line feed character, 0x0a, as specified in [RFC1345].

2.2.1.4 Hook Message



ASCII - + (1 byte): The first three fields specify three occurrences of the ASCII encoding for the '+' character, 0x2b, as specified in [RFC1345]. This string of 3 '+' characters is often referred to as an **escape sequence**.

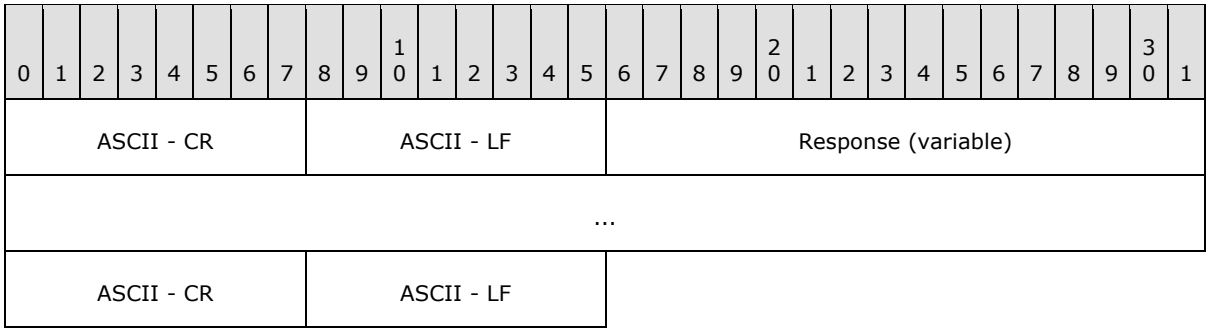
ASCII - A (1 byte): The ASCII encoding for the 'A' character, 0x41, as specified in [RFC1345].

ASCII - T (1 byte): The ASCII encoding for the 'T' character, 0x54, as specified in [RFC1345].

ASCII - H (1 byte): The ASCII encoding for the 'H' character, 0x48, as specified in [RFC1345].

ASCII - CR (1 byte): The ASCII encoding for the carriage return character, 0x0d, as specified in [RFC1345].

2.2.1.5 Hook Response Message



ASCII - CR (1 byte): The ASCII encoding for the carriage return character, 0x0d, as specified in [RFC1345].

ASCII - LF (1 byte): The ASCII encoding for the line feed character, 0x0a, as specified in [RFC1345].

Response (variable): The **Response** field must be an ASCII-encoded string. The **Response** field MUST have one of the following values:

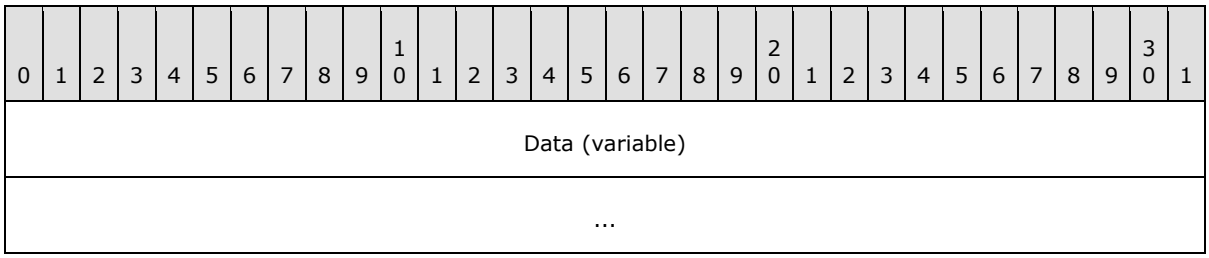
Value	Meaning
"OK"	Modem is already in 'Offline Command Mode' state.
"NO CARRIER"	Connection terminated.
"ERROR"	Unexpected error.

ASCII - CR (1 byte): The ASCII encoding for the carriage return character, 0x0d, as specified in [RFC1345].

ASCII - LF (1 byte): The ASCII encoding for the carriage return character, 0x0a, as specified in [RFC1345].

2.2.1.6 Data Message

The PPP Over IrDA Dialup Protocol relies on the Infrared Network (IrNET) Protocol (for more information, see [MSDN-IRNET]) for the pass-through of data messages between the PPP over HDLC-like framing (as specified in [RFC1662]) and Tiny TP (as specified in [IRTTP]) for the client and the server.



Data (variable): The **Data** field MUST be ASCII encoded. If the length of the **Data** field exceeds the maximum size of a single Data-Carrying TinyTP PDU packet, the TinyTP Protocol performs segmentation and reassembly (SAR) of the **Data** field.

3 Protocol Details

3.1 Common Details

The following sections specify details of the PPP Over IrDA Dialup Protocol including abstract data models and message processing rules.

3.1.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

The following data model and state are common for the client and the server role of the protocol.

Modem state: Data element that describes the current state of the modem device. The modem device can be in one of two states: offline command mode or online data mode.

Offline command mode state: In this state there is no call up and the modem is accepting AT commands via IrDial messages. This is the state in which all sessions are started.

Online data mode state: There is a call up and the modem can receive and transmit data.

IrDial-TTP connection state: Data element that describes the current state of the TinyTP connection that is used for PPP Over IrDA Dialup Protocol communication. The server and the client can either be in an IrDial-TTP connection established state or in an IrDial-TTP connection not-established state.

3.1.2 Timers

This protocol includes the following timer:

IdleTimer: 12 seconds expiration timer

Besides the aforementioned timer, the IrDA Tiny TP (as specified in [IRTTP]) transport over which PPP Over IrDA Dialup Protocol messages are conveyed may have timers associated with it to achieve guaranteed and in-order delivery.

3.1.3 Initialization

The initially assumed values for the data elements defined in section 3.1.1 are:

- Modem state: Offline command mode.
- IrDial-TTP connection state: IrDial-TTP connection not-established.

For more information, see sections 3.2.3 and 3.3.3.

3.1.4 Higher-Layer Triggered Events

For more information, see sections 3.2.4 and 3.3.4.

3.1.5 Message Processing Events and Sequencing Rules

For more information, see sections 3.2.5 and 3.3.5.

3.1.6 Timer Events

After the PPP Over IrDA Dialup Protocol is initialized as specified in sections 3.2.3 and 3.3.3, both the client and the server set the IdleTimer. The timer is reset by the client and the server any time they send or receive a message.

When the timer fires, the PPP Over IrDA Dialup Protocol is moved to the uninitialized state.

3.1.7 Other Local Events

There are no local events common to both server and client.

3.2 Server-Specific Details

The server role is always assumed by the modem that implements the PPP Over IrDA Dialup Protocol on an infrared device.

3.2.1 Abstract Data Model

There is no additional server-specific data model. The data model in section 3.1.1 is assumed.

3.2.2 Timers

No timers, beyond that defined in section 3.1.2, are defined for the server role.

3.2.3 Initialization

The initialization of the server is a two-step sequence:

1. The server MUST register a service access point (SAP) with the **IAS**, as specified in [IRLMP] section 3.1.2, and wait for any client to connect to it. The service access point object contained in the IAS MUST have class name **IrModem** as defined in [IRLMP] section 4.2.4. This allows other PPP Over IrDA Dialup Protocol clients to discover the PPP Over IrDA Dialup Protocol service on the server.

2. The modem device on the server is put in the offline command mode state.

3.2.4 Higher-Layer Triggered Events

A user can configure the device to become a PPP Over IrDA Dialup Protocol server. In the case of this event, the sequence described in section 3.2.3 is initiated.

3.2.5 Message Processing Events and Sequencing Rules

3.2.5.1 Receiving Dial Message

When a dial message arrives, the server can be in one of two modes: offline command mode or online data mode.

If the server is in offline command mode when a dial message is received, the server responds by sending the same dial message. This response is called "echoing the message." Upon echoing the message, the server executes the Dial command sequence and sends a dial response message with the appropriate **Response** field string.

If the server is in online data mode when a dial message is received, the server treats it as a regular dial message (that is, it does not execute the Dial command sequence and takes no action).

3.2.5.2 Receiving Hook Message

When a hook message arrives, the server can be in one of two modes: offline command mode or online data mode.

If a server is in offline command mode when a hook message is received, it responds by echoing the hook message followed by a hook response message with the **Response** field set to "OK".

If a server is in online data mode when a hook message is received, it echoes the message and sends a hook response message with the **Response** field set to "NO CARRIER". As a result of this message being sent, the server moves from the online data mode state to offline command mode state.

3.2.6 Timer Events

There are no timer triggered events beyond those described in section 3.1.6.

3.2.7 Other Local Events

No local events are defined.

3.3 Client-Specific Details

3.3.1 Abstract Data Model

In addition to the data model specified in section 3.1.1, the client has the following data element:

Phone number for modem to dial: A data element that contains the phone number that a client sends in the dial message (section 2.2.1.2).

3.3.2 Timers

No timers, beyond that defined in section 3.1.2, are defined for the client role.

3.3.3 Initialization

A client that wants to establish a TinyTP connection to be used by the PPP Over IrDA Dialup Protocol MUST perform an IAS GetValueByClass on the class name **IrModem**, attribute **IrDA:TinyTP:LsapSel**, as specified in [IRLMP] section 4.2.4. The client MUST initiate the TinyTP connection to the **LSAP-SEL** value returned by the server, as specified in [IRTTP] section 2.2.1.

3.3.4 Higher-Layer Triggered Events

The user initializes the PPP Over IrDA Dialup Protocol as specified in section 3.3.3.

The user initiates a dialing sequence by passing the phone number for modem to dial to the PPP Over IrDA Dialup Protocol.

3.3.5 Message Processing Events and Sequencing Rules

3.3.5.1 Sending Dial Message

The dial message is sent to the server as specified in section 2.2.1.2.

3.3.5.2 Receiving Dial Response Message

If the client is in online data mode when the dial response message arrives, the client treats it as a regular data message (that is, it does not parse the Dial Response Message Request field and takes no action).

If the client is in offline command mode when the dial response message arrives, the client extracts the connect parameters and takes action based on the **Response** field.

If the **Response** field contains 'CONNECT <speed>' the client changes its modem state from offline command mode to online data mode. The protocol passes the **Response** field to the higher-layer protocol, which results in the client starting to send data messages to the server as specified in [RFC1661] to perform PPP layer link establishment.

In the case when the **Response** field is "NO CARRIER", "ERROR", "NO DIALTONE", or "BUSY", the client changes its modem state to offline command mode and passes the **Response** field to the higher-layer protocol.

3.3.5.3 Sending Hook Message

The client MAY send a hook message to the server as specified in section 2.2.1.4.

3.3.5.4 Receiving Hook Response Message

When the hook response message arrives, the client extracts the **Response** field and passes it to a higher-layer protocol.

3.3.5.5 Receiving Echo Message

When the client receives the echo of its own message from the server, it discards it and waits for the response message. For example, when the client sends a dial message, it receives an echo of the dial message, which it discards, and continues to wait for the arrival of a dial response message.

3.3.5.6 Receiving Data Message

When a client receives a data message, it is processed as specified in [RFC1662] section 4.

3.3.5.7 Sending Data Message

A client sends a data message, as specified in [RFC1662] section 4, encapsulated in one or more Tiny TP (as specified in [IRTTP]) SDUs.

3.3.6 Timer Events

There are no timer-triggered events beyond those discussed in section 3.1.6.

3.3.7 Other Local Events

Upon a successful dial/dial response message exchange from the Tiny TP (as specified in [IRTTP]) layer and a dial response message with a "Connect" Response, the PPP Over IrDA Dialup Protocol indicates, by passing the "Connect" Response to the PPP layer that the PPP Over IrDA Dialup Protocol link is ready for data message exchange.

4 Protocol Examples

The following sections describe several operations as used in common scenarios to illustrate the function of the PPP Over IrDA Dialup Protocol.

4.1 Connection Setup and Data Exchange

In this example, the user initiates the PPP Over IrDA Dialup Protocol connection by initializing the client-side protocol as specified in section 3.3.3. The user then passes the phone number to the client-side PPP Over IrDA Dialup Protocol as specified in section 3.3.4. It is assumed that server-side initialization, as specified in section 3.2.3, has already taken place.

1. The client sends a dial message as specified in section 2.2.1.2, with the **Number** field containing 8001231234.
2. The server echoes the dial message back to the client (not shown in Figure 2).
3. The server sends the dial response message, which contains the connection speed at which it can operate in the form <CR><LF>CONNECT 9600<CR><LF>, where 9600 is the speed negotiated between the two modems (one modem being the PPP Over IrDA Dialup Protocol server and the other modem being the device at number 800-123-1234 from step 1).
4. Once the connection is established and the client-side of the PPP Over IrDA Dialup Protocol indicates to the PPP layer to send data, the client sends a data message to the server as specified in section 2.2.1.6.
5. The server sends a data message to the client, which is delivered to the PPP Protocol as specified in [RFC1662].
6. The client sends a hook message to the server.
7. The server sends a hook response message to the client with the **Response** field of 'NO CARRIER'.

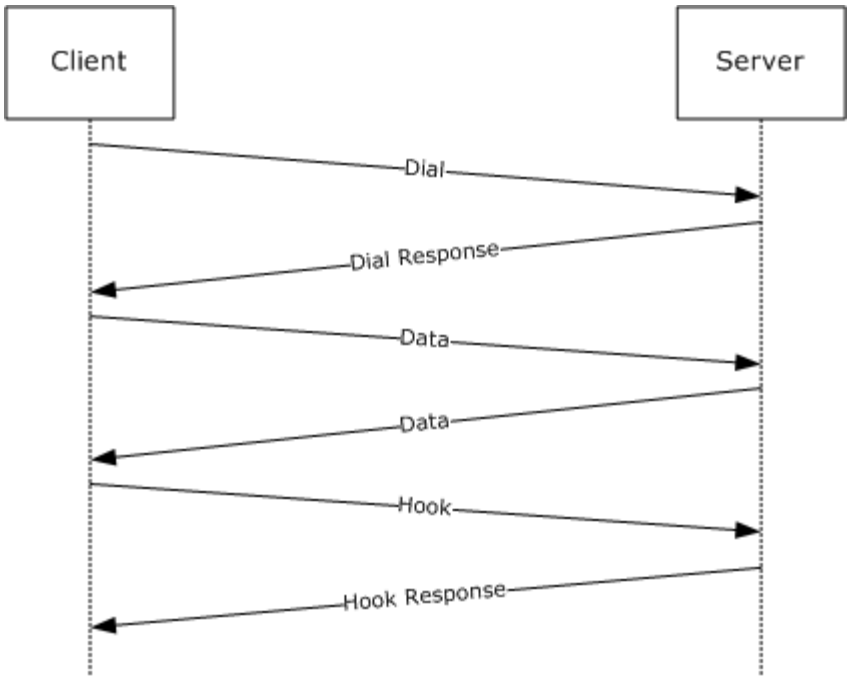


Figure 2: Example of PPP Over IrDA Dialup Protocol connection setup, data exchange, and connection teardown.

5 Security

The following sections specify security considerations for implementers of the PPP Over IrDA Dialup Protocol.

5.1 Security Considerations for Implementers

The PPP Over IrDA Dialup Protocol does not provide any security mechanisms.

5.2 Index of Security Parameters

The PPP Over IrDA Dialup Protocol does not introduce any security parameters.

6 Appendix A: Windows Behavior

The information in this specification is applicable to the following versions of Windows:

- Windows Server 2003

Exceptions, if any, are noted below. Unless otherwise specified, any statement of optional behavior in this specification prescribed using the terms SHOULD or SHOULD NOT implies Windows behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that Windows does not follow the prescription.

7 Index

S

T

V

W