IHDC Protocol

Usage Instruction

InHand Networks

www.inhandnetworks.com

Version: V2.0

Date: Jan,2019

Contents

1. About This Document

1.1. Purpose

The InDTU300 series is a data transmission terminal product based on wireless cellular networks. InDTU300 uses the Data Center (DC) Protocol for communication, which is based on UDP or TCP. This document introduces the DC protocol in detail and provides reference for development when based on DTU300 series.

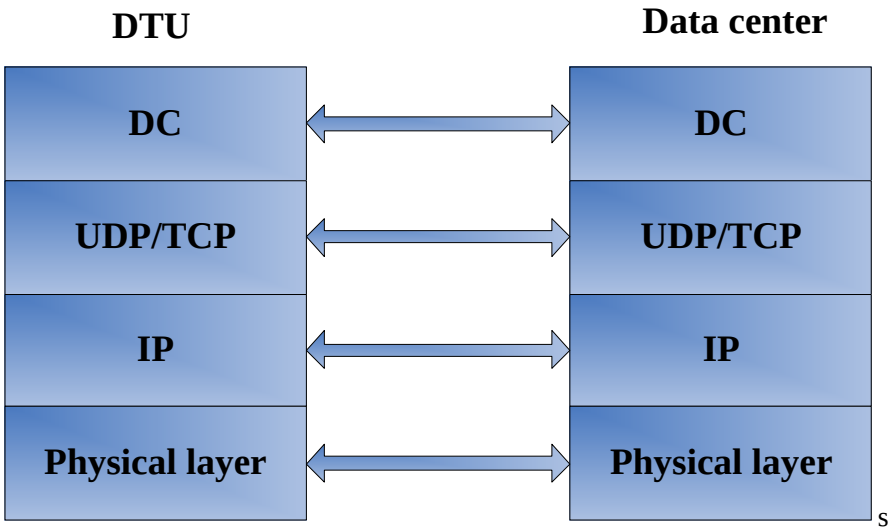
1.2. Definition

The DC protocol is a data communication protocol defined by InHand Networks. It is based on UDP or TCP and provides mechanisms such as login, heartbeat, data transmission, and logout.

2. DC Protocol Content

2.1. DC Protocol Stack Structure

The following figure shows the protocol stack representation of the DC protocol.

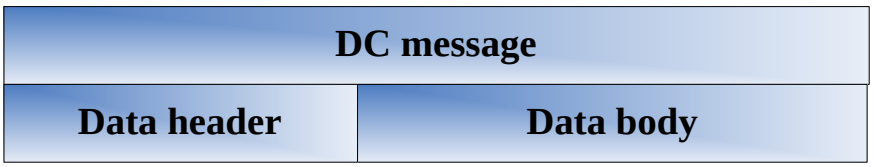


s

For DTU300, its physical layer is wireless cellular network, the dial link for the wireless cellular network (PPP protocol) will act as underlying bearer.

2.2. DC Protocol Packets

The DC protocol defines a protocol data format based on byte streams. The data structure is as follows:



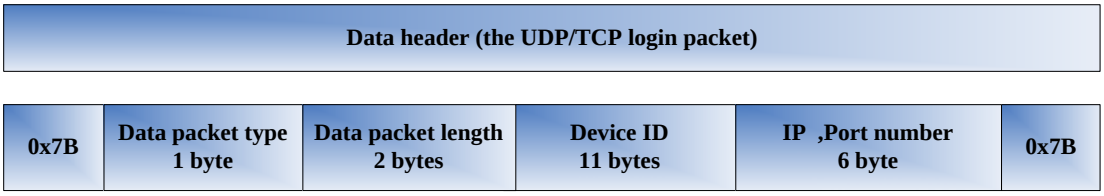
A DC message consists of two parts: data header and data body. The DC protocol defines several packet types, which are indicated by the data packet type identifier (which occupies 1 byte) in the data header. See the following table.

No.	Identifier	Function	Data Flow	Message Length(Byte)	Remarks
1	0x03	Login	DTU -> Data center	22	The data body is empty.
2	0x83	Response to login	DTU <- Data center	16	The data body is empty.
3	0x01	Heart beat	DTU -> Data center	22	The data body is empty.
4	0x81	Response to heartbeat	DTU <- Data center	16	The data body is empty.
5	0x09	Proactive data reporting	DTU -> Data center		The value is invariably 16 if the DC protocol is based on UDP.
6	0x89	Data delivery by the data center	DTU <- Data center		The value is invariably 16 if the DC protocol is based on UDP.
7	0x85	Response to data reporting	DTU <- Data center	16	The data body is empty.
8	0x82	Proactively offline	DTU <-> Data center	22	The data body is empty.
9	0x02	Response to proactively offline	DTU <-> Data center	16	The data body is empty.

Each packet type is defined as follows.

2.2.1. Login Packet

A UDP/TCP login packet adopts the following data header:



The value of the data packet type is 0x03. The IP address assigned by the wireless cellular network occupies 4 bytes. The port number occupies 2 bytes. The length of the data header is 1 + 1 + 2 + 11 + 4 + 2 + 1 = 22 bytes. Network endian is adopted. The data body is empty.

Note:

(1) The device ID is an ASCII-encoded string. If the configured device ID is less than 11 bytes, the DTU fills in 0x00 after the device ID. If the device ID is more than 11 bytes, the DTU keeps the first 11 bytes.

a) For example, if the configured DTU ID is 1234, the device ID in the DC protocol packet header is 0x31, 0x32, 0x33, 0x34, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00.

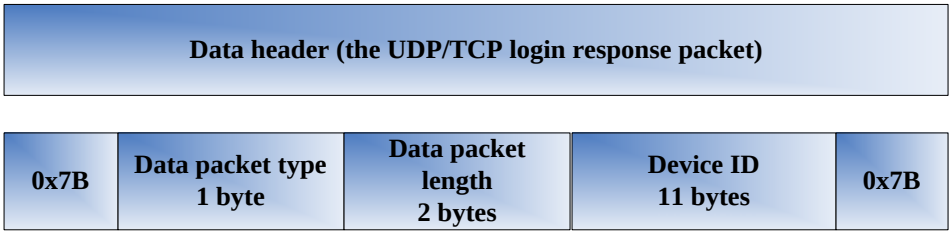
b) If the configured DTU ID is 123456789012, the device ID in the DC protocol packet header is 0x31, 0x32, 0x33, 0x34, 0x35, 0x36, 0x37, 0x38, 0x39, 0x30, 0x31.

(2) Each byte of the IP address assigned by the wireless cellular network is in the hexadecimal format. The two bytes of the port number are also in the hexadecimal format. The IP address and port number adopt network endian. The data header length is indicated by a 2-byte hexadecimal value.

For example, the UDP login packet 7B 03 00 16 3132333400000000000000 0A 0F 07 0C 77 05 7B indicates the device with the ID 1234. Its IP address is 10.15.7.12 and its port is 30496 (0x7705).

2.2.2. Login Response Packet

A UDP/TCP login response packet adopts the following data header:

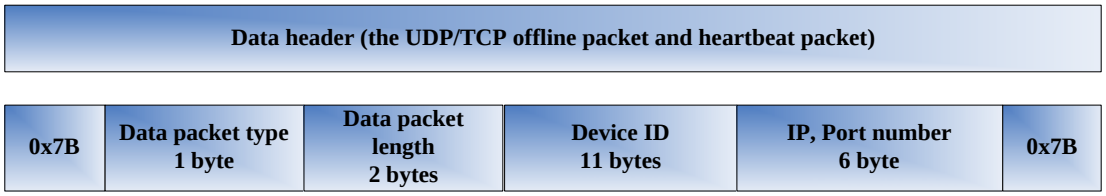


The value of the data packet type is 0x83. The data packet length is 16 bytes. The data body is empty.

For example, a UDP login response packet is 7B 83 00 10 3132333400000000000000 7B.

2.2.3. Offline Packet and Heartbeat Packet

A UDP/TCP offline packet and heartbeat packet adopt the following data header:



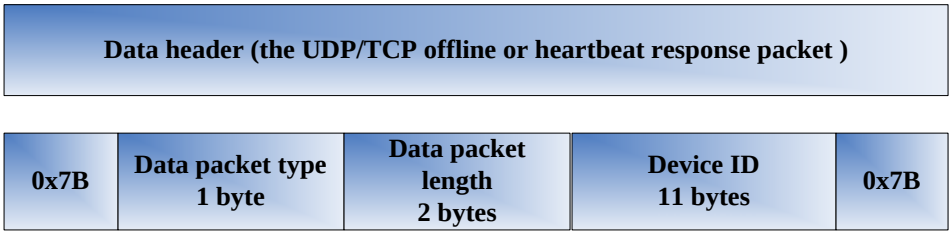
The value of the data packet type is 0x01 (heartbeat packet) or 0x82 (offline packet). The data body is empty.

For example, a UDP heartbeat packet is 7B 01 00 10 3132333400000000000000 C0A80101 1234 7B.

A UDP offline packet is 7B 82 00 10 3132333400000000000000 C0A80101 1234 7B.

2.2.4. Offline Response Packet and Heartbeat Response Packet

A UDP/TCP offline response packet and heartbeat response packet adopt the following data header:

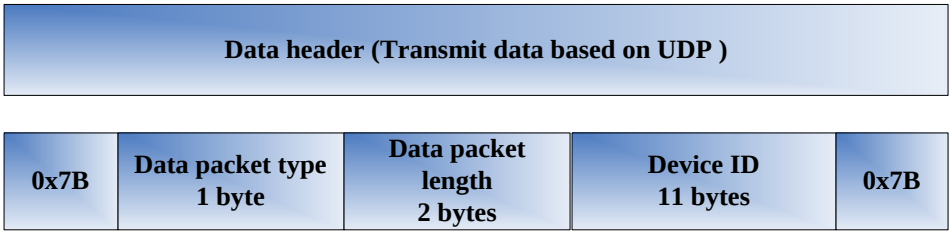


The value of the data packet type is 0x81 (heartbeat response packet) or 0x02 (offline response packet). The data header is empty. The data body is empty.

For example, a UDP heartbeat response packet is 7B 81 00 10 3132333400000000000000 7B.

2.2.5. UDP Data Packet and Data Delivery Packet

A UDP Data Packet and UDP data center-delivered data packet adopt the following data header:



The value of the data packet type is 0x09 (terminal-reported data packet) or 0x89 (data center-delivered data packet).

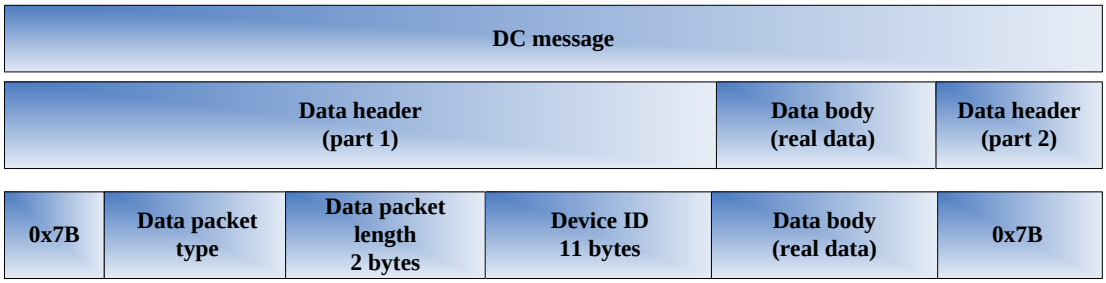
For example, a UDP Data Packet is 7B 09 00 10 3132333400000000000000 7B 31 32 33 34 35 36 37, in which the real data is 31 32 33 34 35 36 37.

Note:

For a UDP Data Packet, the data packet length is invariably 16 bytes (0x0010), excluding the data body length. The data body length is the total length of the received UDP packet subtracted by 16.

2.2.6. TCP Data Packet and Data Delivery Packet

A TCP Data Packet and TCP data center-delivered data packet adopt the following DC message format:



The value of the data packet type is 0x09 (terminal-reported data packet) or 0x89 (data center-delivered data packet).

Note:

A TCP data packet has special combination of the data header and data body, which is different from that of other data packet types. The data body is located in the middle, and the data packet length includes the data body. The length of the data header is 1 + 1 + 2 + 11 + 1 = 16 bytes. The length of the real data body is the data packet length subtracted by 16 bytes.

2.2.7. Data Report Response Packet

The DTU does not return a packet (0x85) in response to the data packet sent by the data center, regardless of whether UDP or TCP is adopted.

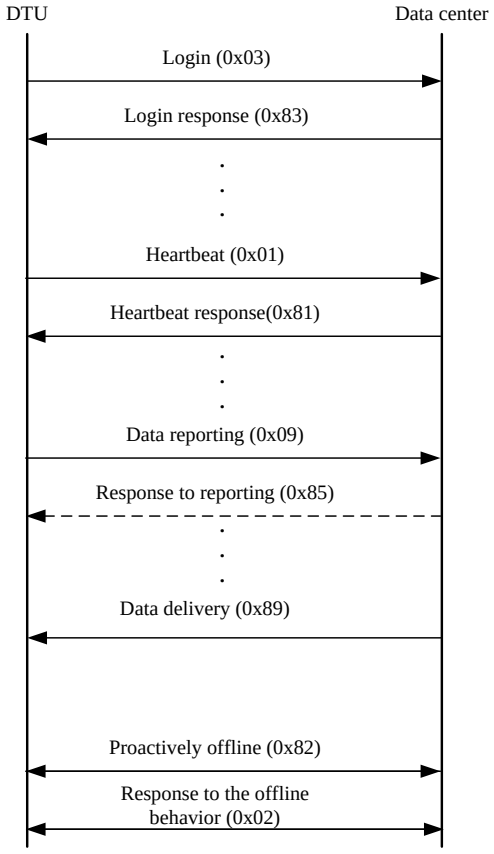
When the DTU reports a data packet to data center, the data center determines whether to return a response packet. The data header of the response packet is as follows:

Data header (Response packet of data center based on TCP/UDP ,after receive data)				
0x7B	**Data packet type 1 byte**	**Data packet length 2 bytes**	**Device ID 11 bytes**	**0x7B**

The data packet type identifier is 0x85. The data body is empty. The length of the entire data packet is 1 + 1 + 2 + 11 + 1 = 16 bytes.

2.3. Data Exchange Process of the DC Protocol

Data exchange of the DC protocol is a synchronous process, as shown in the following figure.



The data exchange process is divided into three phases:

✧ **Login:** DTU must logs in to the data center before sending data, and then sends heartbeats at a specified interval. If the DTU receives no heartbeat response for a specified number of consecutive times, it determines that the connection is incorrect, initiates login again, and restarts when necessary.

Note:

In protocols v1.6 and earlier, the data packet type of login packets and heartbeat packets is 0x01. In protocols later than v1.6, a new login command is added (the data packet type is 0x03). To be

compatible with the center program of earlier versions, the DTU first makes two login attempts at an interval of 10 seconds. If no response is received, the DTU sends three heartbeat packets (previous login packets) at an interval of 10 seconds. If no response is received, the network or data center is faulty.

✧ **Data transmission:** Two modes of data transmission are provided: request-response and proactive reporting. The DTU can report data and the data center can deliver data.

✧ **Offline**: DTU sends a proactive offline data packet before disconnection. The data packet may be lost because the network connection is unreliable. The data center determines the terminal status based on heartbeat timeout rather than the data packet.