TDTP – a Trivial Data Transfer Protocol

© 2009, Yuri Tiomkin

A **TDTP** protocol is similar to the **TFTP** protocol, but it has a few remarkable extensions:

- Using Crc32 for the each packet
- Unique Session ID for the each session
- Total data size field as standard part of the protocol
- Variable data portion size per packet

A **TDTP** protocol uses **UDP** protocol as the transport layer.
The UDP ports numbers are not a part of the protocol and the port number value can be any valid UDP port number. In the **TDTP** examples, a host UDP port number was selected as 50000, the peer UDP port number – as 11094.

1. TDTP protocol messages

All 2–bytes and 4-bytes message fields are transferred in the network (big endian) byte order. Address and Data message fields are transferred as sequence of bytes (octets).

1.1 "WR" message

A "WR" message establishes a session to send data to the peer and sends a portion of the data.

1.1.1 "WR" message format

Cmd (1 byte)	**Session ID** (4 bytes)	**Block Num** (4 bytes)	**Tsize** (4 bytes)	**Address Length** (2 bytes)	**Address** (variable)	**Data** (variable)	**CRC32** (4 bytes)

Cmd For the message "WR" this field is 3 (TDTP_CMD_WR)

Session ID This ID will be a current session ID. It should be unique for the each session

Block Num Current block number (for the command "WR" this field is '0' if it is a single block transfer session or '1' if it is a multi-block transfer session).

Tsize Total size of transferred data (like a file length).

Address Length Length of the **Address** field.

Address Address may be a file name/path or device name/id etc.

Data A packet's data. A packet's data length is calculated as
Data length = UDP payload length – (1 + 4 + 4 + 4 + 2 + Address Length + 4)

CRC32 CRC32 for the overall "WR" packet (includes header fields and data)

1.2 "WR_DATA" message

A "WR_DATA" message sends a portion of the data to the peer.

1.2.1 "WR_DATA" message format

Cmd (1 byte)	**Session ID** (4 bytes)	**Block Num** (4 bytes)	**Data** (variable)	**CRC32** (4 bytes)

Cmd For the message "WR_DATA" this field is 4 (TDTP_CMD_WR_DATA)

Session ID A current session ID.

Block Num Current block number.

Data A packet’s data. A packet’s data length is calculated as
Data length = UDP payload length – (1 + 4 + 4 + 4)

CRC32 CRC32 for the overall “WR_DATA” packet (includes header fields and data)

1.3 “RD” message

A “RD” message establishes a session to receive data from the peer.

1.3.1 “RD” message format.

Cmd	**Session ID**	**Address Length**	**Address**	**CRC32**
(1 byte)	(4 bytes)	(2 bytes)	(variable)	(4 bytes)

Cmd For the message “RD” this field is 1 (TDTP_CMD_RD)

Session ID This ID will be a current session ID. It should be unique for the each session.

Address Length Length of the **Address** field.

Address Address may be a file name/path or device name/id etc.

CRC32 CRC32 for the overall “RD” packet (includes header fields and data)

1.4 “RD_DATA” message

A “RD_DATA” message sends a portion of the data from the peer.

1.4.1. “RD_DATA” message format.

Cmd	**Session ID**	**Block Num**	**Tsize**	**Data**	**CRC32**
(1 byte)	(4 bytes)	(4 bytes)	(4 bytes)	(variable)	(4 bytes)

Cmd For the message “RD_DATA” this field is 2 (TDTP_CMD_RD_DATA)

Session ID A current session ID.

Block Num Current block number.

Tsize Total size of transferred data (like a file length) – the same in the all “RD_DATA” blocks

Data A packet’s data. A packet’s data length is calculated as
Data length = UDP payload length – (1 + 4 + 4 + 4 + 4)

CRC32 CRC32 for the overall “RD_DATA” packet (includes header fields and data)

1..5 “ACK” message

An “ACK” is used to send response for other commands. An **ACK message** field contains a positive/ negative response or other information.

1.5.1. “ACK” message format

Cmd	**Session ID**	**Block Num**	**ACK message**	**CRC32**
(1 byte)	(4 bytes)	(4 bytes)	(4 bytes)	(4 bytes)

Cmd For the message “ACK” this field is 5 (TDTP_CMD_ACK_RD)
or 6 (TDTP_CMD_ACK_WR)

Session ID A current session ID.

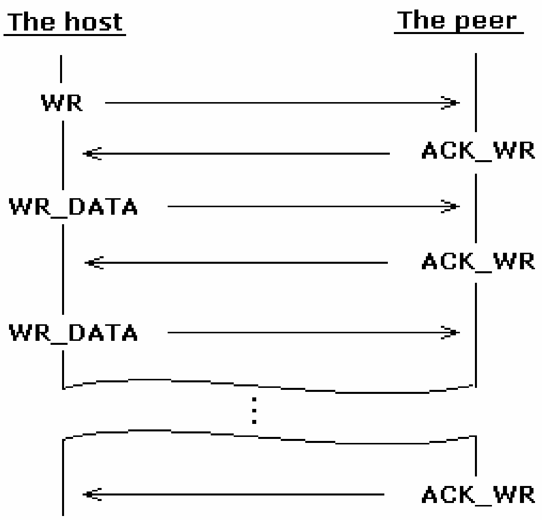
Block Num A block number for the response.

ACK message The “ACK” response info.

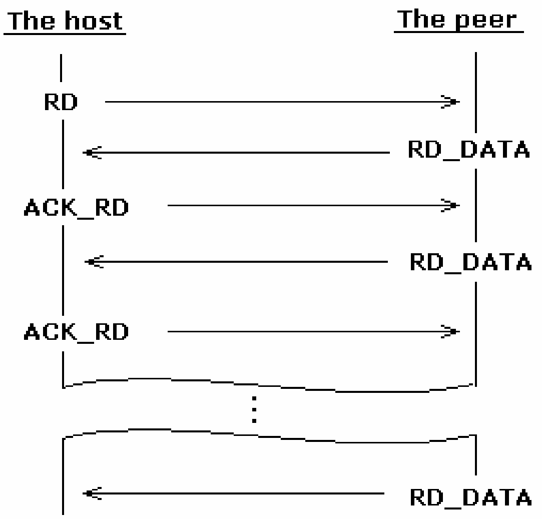
CRC32 CRC32 for the overall “ACK” packet

2. A TDTP data transfer scenarios

The host sends a data to the peer



The host reads a data from the peer



3. An implementation examples

A source code for **TDTP** protocol implementation can be obtained (it's free under BSD-like license) from the www.tnkernel.com (TN NET TCP/IP stack examples)