

TFTP “bigfile” option

1. Introduction

The Trivial File Transfer Protocol (TFTP) is commonly used for file transfer between a client and a server in network environments. However, the size of files that can be transferred is limited by the standard specification of the TFTP protocol. To allow the transfer of large files without limitation, this RFC introduces a new option, called "bigfile," for the TFTP protocol.

2. Description of the "bigfile" Option

2.1 Syntax of the Option

The "bigfile" option is specified in the Read Request (RRQ) and Write Request (WRQ) packets as follows:

```
+-----+---~~---+---+---~~---+---+---~~---+---+---~~---+---+-->
| opc |filename| 0 | mode | 0 | opt1 | 0 | value1 | 0 | <
+-----+---~~---+---+---~~---+---+---~~---+---+---~~---+---+-->

>-----+---+---~~---+---+
< optN | 0 | valueN | 0 |
>-----+---+---~~---+---+
```

opc

The opcode field contains either a 1, for Read Requests, or 2, for Write Requests.

filename

The name of the file to be read or written.
This is a NULL-terminated field.

mode

The mode of the file transfer: "netascii", "octet", or "mail".

This is a NULL-terminated field.

opt1

The first option, in case-insensitive ASCII (e.g., blksize).

In this case for 'bigfile' option opt1 contain 1

value1

The value associated with 'big file' option.

optN, valueN

The final option/value pair. Each NULL-terminated field is specified in case-insensitive ASCII.

Upon receiving a RRQ or WRQ packet containing the "bigfile" option, the server should examine the requested options. If the server supports the "bigfile" option, it responds with an Option Acknowledgment (OACK) packet to confirm acceptance of the option. If the server does not support the "bigfile" option, it sends an Error packet with error code 8 (Option Negotiation Failure) to indicate that the requested option is not available.

2.2 Server Response

Upon receiving an RRQ or WRQ with the "bigfile" option, the server responds with an Option Acknowledgment (OACK) packet if it supports large file transfers. The OACK packet includes the "bigfile" option to confirm its support.

```
+-----+---~---+---+---~---+---+---~---+---+---~---+---+
| opc | opt1 | 0 | value1 | 0 | optN | 0 | valueN | 0 |
+-----+---~---+---+---~---+---+---~---+---+---~---+---+
```

opc

The opcode field contains a 6, for Option Acknowledgment.

opt1

The first option acknowledgment, copied from the original

request.

value1

The acknowledged value associated with the first option. If and how this value may differ from the original request is detailed in the specification for the option.

optN, valueN

The final option/value acknowledgment pair.

If the client receives an OACK packet confirming the "bigfile" option, it proceeds with the file transfer operation as per the standard TFTP protocol, with the ability to handle files of large size.

3. File Transfer Management

The "bigfile" option facilitates the transfer of large files. When a file to transfer exceeds the standard size that TFTP can handle, the client and server use this option to signal their capability to handle such files. This allows the file to be fragmented into multiple DATA packets, each followed by an ACK packet, until the transfer is complete.

Each data block is assigned a unique block number, starting from 1 and incrementing for each successive block. This numbering plays a key role in the transmission order and verification of received data integrity.

In case of transmission errors or packet loss, the block number allows the server and client to identify and retransmit specific missing or erroneous blocks, without requiring the retransmission of the entire file.

The transmission of data blocks follows a strict protocol where each block sent by the sender (server or client) is acknowledged by an ACK packet from the receiver, indicating successful reception:

Transmission Sequence: Data blocks are sent sequentially. The sender waits for an acknowledgment for each block before proceeding to send the next block.

Transfer Integrity: ACKs serve as confirmation that each block has been correctly received and recorded by the client, thus ensuring the integrity of the transferred file.

4. Security Considerations

The basic TFTP protocol has no security mechanism. This is why it has no rename, delete, or file overwrite capabilities. This document does not add any security to TFTP; however, the specified extensions do not add any additional security risks.

5. References

RFC 1350, The TFTP Protocol (Revision 2)

RFC 2347, TFTP Option Extension