UPJV Students B.Nader
T.Ata

Request for Comments : 80000
Category : Standards Track

TFTP Bigfile Option

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Abstract

The Trivial File Transfer Protocol [1] is a simple, lock-step, file transfer protocol which allows a client to get or put a file onto a remote host. One of its primary uses is the booting of diskless nodes on a Local Area Network. TFTP is used because it is very simple to implement in a small node's limited ROM space. However, the choice of a 512-octet blocksize is not the most efficient for other uses such as transferring large multimedia files, such as videos or high-resolution images.

The problem is solved using the blocksize option [2] nevertheless this option is limited by the number of blocks (2 bytes) and the maximum size of block described in the blocksize option RFC (65464 bytes).

Using the blocksize option will allow us to transfer data more efficiently with a maximum file size 4290 mb which is also limited.

This document describes a TFTP option which allows the client and server to transfer data with unlimited size.
The TFTP Option Extension mechanism is described in [3]

Bigfile Option Specification

The TFTP Read Request or Write Request packet is modified to include the bigfile option as follows. Note that all fields except "opc" are NULL-terminated.

Read Request packet

```
+-------+---~~---+---+---~~---+---+---~~---+---+---~~----+---+
|  opc  |filename| 0 |  mode  | 0 | bigfile| 0 | maxsize | 0 |
+-------+---~~---+---+---~~---+---+---~~---+---+---~~----+---+
```

Write Request packet

```
+-------+--~~----+---+---~~---+---+---~~---+---+----~~----+---+
|  opc  |filename| 0 |  mode  | 0 | bigfile| 0 | filesize | 0 |
+-------+---~~---+---+---~~---+---+---~~---+---+---~~-----+---+
```

opc

The opcode field contains either a 1, for Read Requests, or 2,for Write Requests, as defined in [1].

filename

The name of the file to be read or written, as defined in [1].

mode

The mode of the file transfer: "netascii", "octet", or "mail",as defined in [1].

bigfile

The Bigfile option, "bigfile" (case in-sensitive).

maxsize

The maximum size in MB that the client is ready to get from the server,this is useful because he may not be aware of any changes to the file since the last time it was retrieved from the server.

filesize

The size of the file in MB to put onto the server.This useful to make sure that the server is capable of receiving a file with such size.

Bigfile Data Packet

The Bigfile option will use a large data packet size 65464 bytes matching the maximum block size defined in blocksize option [2].

The goal here is to have a better transmission speed for large files. Given that the standard TFTP protocol supports file transfers up to 32 MB, it is advisable to not use this option for files equal to or less than 32 MB in size, effectively reducing memory usage.
The client and server must implement these changes.

Bigfile Block Number

The Bigfile option will allow to go beyond the maximum number of blocks (65535 bytes) by simply returning to block# 1 after sending / receiving the block# 65535, the client and server must implement this change.

For example:

```
+-------+------------+---+--------+---+---------+---+-------+---+
|   1   | titanic.mp4| 0 | octet  | 0 | bigfile | 0 | 30000 | 0 |
+-------+------------+---+--------+---+---------+---+-------+---+
```

is a Read Request, for the file named «titanic,mp4», in octet (binary) transfer mode, with maximumsize of 30000 MB (30 GB).

If the server is willing to accept the bigfile option, it sends an Option Acknowledgment (OACK) to the client. The specified value must be less or equal to the value specified by the client (the value is the size of the file requested by the client). The client must then either accept the OACK or send an ERROR packet, with error code 8, to terminate the transfer.

The rules for determining the final packet are unchanged from [1]. The reception of a data packet with a data length less than 65464 bytes is the final packet. If the blocksize is greater than the amount of data to be transfered, the first packet is the final packet. If the amount of data to be transfered is an integral multiple of the blocksize, both the sender and the receiver will eventually time out, leading to the termination of the transfer process.

Proof of Concept

Performance tests were conducted on a prototype implementation on a TFTP server and TFTP client hosted on the same machine. The tests aimed to evaluate the efficiency and effectiveness of transferring files with big size on a local environment.
These tests also where used in a multi user environment (2 clients reading the same file from the TFTP server).

```
             Seconds
                80 +
                   |
                   |                                      x
                   |
                70 +
                   |                                  x
                   |
                   |
                60 +
                   |                           filesize (GB) | time (s)
                   |                           -------------   ------
                   |                               1,203         26
                50 +                               1,587         30
                   |                               2,975         68
                   |                               3,192         75
                   |
                40 +
                   |
                   |
                   |
                30 +                    x
                   |
                   |                 x
                   |
                20 +
                   |
                   |
                   |
                10 +
                   |
                   |
                   |
                 0 +-//--+-----+-----+-----+-----+-----+-----+-→GB
                        0,5    1    1,5    2    2,5    3    3,5
```

As was anticipated, the transfer time increases with an increase in filesize. The reason for the increase in time is the increase in the number of packets sent.

Files Concurrency

If the TFTP Server has the capabilities to manage multiple clients concurrently, it must respect the principles of the Writer-Reader problem [3], meaning to deny access to clients when needed, by sending a ACCESS DENIED error packet (error code 2).

References

[1] Sollins, K., "The TFTP Protocol (Revision 2)", STD 33, RFC 783 , October 1992.

[2] Malkin, G., and A. Harkin, "TFTP Option Extension", RFC 2347,May 1998.
[3] Writers-Readers problem, Wikipedia.

Authors' Addresses

BEN AMMAR Nader
UPJV
33 Rue St Leu, 80000 Amiens
Phone: +33766549265
EMail: nader.ben.ammar@etud.u-picardie.fr

TOZ Ata
33 Rue St Leu, 80000 Amiens
Phone: +33766549265
EMail: ata.toz@etud.u-picardie.fr

Full Copyright Statement