分散システム特論レポート

石坂 寿朗

平成 11 年 3 月 10 日

1 TFTP

1.1 特徴

TFTP(trivial file transfer protocol) は、RFC1350 で規定される、非常に単純なファイル転送プロトコルである。

TFTP の特徴として、次のものが挙げられる。

- UDP
- 実装が小さく、容易
- ファイル読み書きのみ、ディレクトリ表示やユーザ認証は無い
- 3 つの転送モード: アスキー (netascii), バイナリ (octet), mail(obsolete)

1.2 概要

tftp は、request から転送が開始される。

パケットは 512bytes の固定長で、512bytes 未満は転送完了を意味する。

ack を受信するのを待ってから次のパケットを送信する。

パケットがネットワーク中で紛失した場合は受信側がタイムアウトして、最後のパケット (data,ack) を再送信し、送信側が再度送信し直す。この様子を図 1 に示す。

エラーによって接続を終了し、エラーパケットを生成する。この時、エラーパケットは再送されないので、タイムアウトでエラーを検出する。

tftp では、三種類のエラーがある。

- 要求を満たせない
- パケット異常
- 資源へのアクセス喪失

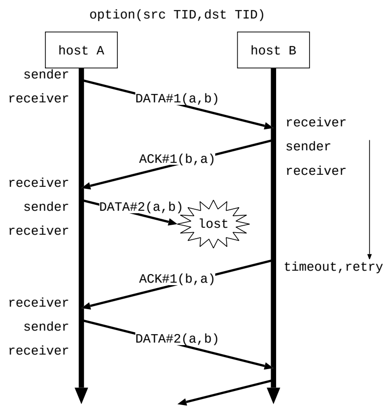

図 1: tftp のタイムアウト時の動作

実装を単純にするため、機能的な制約が多い:

- ブロックを固定長にしてアロケーションを straight forward に、
- lock-step ack によってフローコントロールやデータの並べ替えが不要

1.3 他プロトコルとの関連

tftp は、UDP 上に実装されるよう設計されている。

IP 上に UDP が実装されているので、実際のパケットには IP ヘッダと UDP ヘッダ、そして TFTP ヘッダがつく。また、メディア固有のヘッダが先頭に付く事もある。

IP ヘッダ中の値は規定しないが、UDP ヘッダ中の値は利用される。TID(ポート番号) は initial connection protocol 中で決定される。

TFTP ヘッダ中には 2bytes の opcode があり、packet type を規定している。

1.4 initial connection protocol

読み出し要求の例を図 2 に示す。

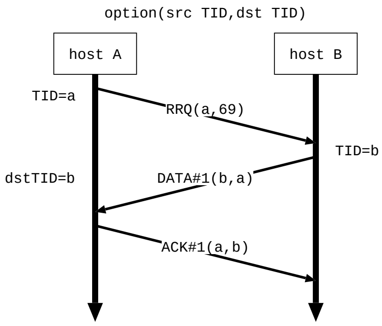

図 2: tftp による読み出し要求の例

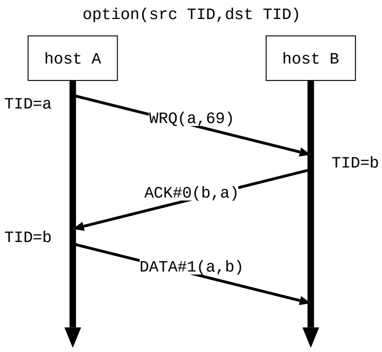

図 3: tftp による書き込み要求の例

次に、書き込み要求の例を図 3 に示す。

この図に示すように、

- 読み: リクエスト送信、最初の読みだし
- 書き: リクエスト送信、write への ack

の順番でパケットが送られる。

ack には、ack の対象となるデータパケットのブロック番号が入っている。データパケットには"1" から始まるブロック番号がつけられる。ただし、write への最初の ack は"0" となる。

最初の応答がエラーパケットなら、リクエストは拒否された事を意味する。

接続を作るために、その接続でのみ使われる自分自身の TID を選択する

応答側は、自分自身の TID をランダムに選択する。こうする事により、同一番号が選択される可能性は低い。

source TID と destin TID は、UDP のポート番号として扱う。ここでは、source TID と destin TID の決定方法について述べる。要求時の source TID は UDP のポート番号、destin TID は初期値 69 を入れて送信する。

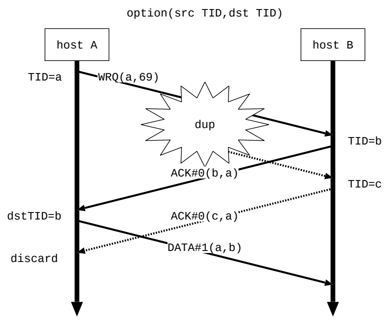

図 4: ネットワーク中での重複

要求に対する ack では、source TID はサーバ側で選択した番号が、destin TID は要求側の番号が格納される。また、これ以降のデータや ack では、ここで選択された TID 番号が利用される。また、src TID が一致しない場合は、単純にその packet を無視する。

例として、ネットワーク中でリクエストが重複した場合を考える。この様子を図 4 に示した。

B はリクエストを 2 つ受信するので、A¡-B に TID の異なる二つの反応が返る。A は、最初の反応に対して接続し、通信を行う。二つ目の反応には拒絶されるが、最初の接続は切断されない。

これより、ネットワーク中でリクエストが重複しても通信は問題なく動作することが分かる。

1.5 TFTP packet

packet の opcode と、その意味の対応は以下の通り。

opcode	意味
1	Read request (RRQ)
2	Write request (WRQ)
3	Data (DATA)
4	Acknowledgment (ACK)
5	Error (ERROR)

1.5.1 RRQ/WRQ request

パケットの書式は以下の通り。

opcode	filename	0	mode	0
2bytes	文字列	1byte	文字列	1byte

opcode は、RRQ 時には"1" の値が、WRQ 時には"2" の値が入る。

mode には、文字列で “NETASCII”, “OCTET”, “MAIL” のいずれかを記述する。大文字、小文字は問わない (case insensitive)。

1.5.2 DATA packet

opcode	block #	data
2bytes	2bytes	n bytes

opcode には、”3” の値が入る。

block # は 1 から始まり、1 ずつ増加する。このように制限することにより、新しいパケットと重複を区別することができる。

data 長は 0 ~ 512bytes で、0 ~ 511bytes なら転送終了を意味する。

1.5.3 ACK packet

opcode	block #
2bytes	2bytes

opcode には、”4” の値が入る。

block #は ack の対象となるブロック番号が入る。また、WRQ に対する応答パケットの場合は block #は”0” となる。

WRQ と DATA パケットに対する応答は ACK 又は ERROR パケットであるが、RRQ と ACK パケットに対する応答は DATA 又は ERROR パケットである。

1.5.4 ERROR packet

opcode	ErrorCode	ErrMsg	0
2bytes	2bytes	string	1byte

ERROR パケットは、他のどのタイプのパケットの応答とすることができる。

opcode には、”5” の値が入る。

ErrorCode はエラーの種類を表す。エラーコードと意味の対応を以下に示す。

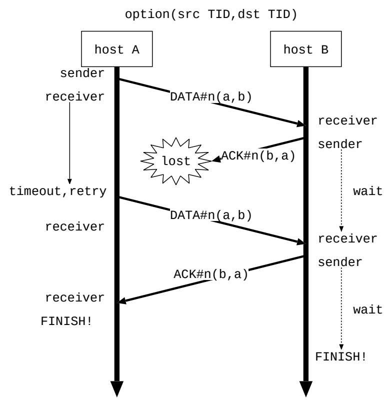

図 5: 最終 ACK の問題

ErrCode	意味
0	Not defined, see error message (if any).
1	File not found.
2	Access violation.
3	Disk full or allocation exceeded.
4	Illegal TFTP operation.
5	Unknown transfer ID.
6	File already exists.
7	No such user.

エラーメッセージは人間が利用するために提供される。

1.5.5 通常終了

データ長が 0 から 511 bytes である DATA パケットにより、転送の終了を通知する。このパケットは他の DATA パケットと同様に ACK パケットにより応答される。しかし、最後の ACK パケットが紛失してしまう可能性があるので、最後の ACK パケットを送出したら直後に終了するのではなく、DATA が再送されるのをしばらく待っておくのが望ましい。

この様子を図 5 に示した。図中に示した右側の点線で待っている部分が必要である。

1.5.6 異常終了

要求が認められなかったり、転送中に何らかのエラーが発生した場合、ERROR パケットが送られる。再送したり応答しないように、単に儀礼的に送信されるものなので、受信される必要はない。タイムアウトもエラー検出に使わなければならない。

2 TFTP 実装の検討

以下では、TFTP を実際に実装したアプリケーションを調査し、正しくプロトコルが実装されているかを検討した。

TFTP は、server-client 型のプロトコルである。よって、検討するアプリケーションは server と client の二つが必要となる。そこで、TFTP の server として FreeBSD-2.2-CURRENT 中の tftpd を選択した[1]。また、同様に TFTP の client として、FreeBSD-2.2-CURRNET 中の netboot を選択した[2]。

[1] ftp://ring.etl.go.jp/pub/FreeBSD/FreeBSD-current/src/libexec/tftpd/
[2] ftp://ring.etl.go.jp/pub/FreeBSD/FreeBSD-current/src/sys/i386/boot/netboot/