TFTP Exercise

1 TFTP protocol description

TFTP (Trivial File Tranfer Protocol) is a simple file transfer protocol. Its detailed specification can be found in RFC 1350. TFTP is built on top of UDP and uses timers to ensure transmission reliability.

In this exercise we consider a simplified version of TFTP.

- files are sent in fixed length blocks of 512 bytes.
- Each data packet contains one block of data, and must be acknowledged by an acknowledgment packet before the next packet can be sent.
- The first packet of a file transfer is a request to read or write a file. It includes file name and type of tranfer (read or write).
- Subsequent blocks are numbered sequentially starting from 1. Sequence number is inserted in TFTP packet header. Every ACK includes sequence number of the corresponding acknowledged packet.
- A data packet of less than 512 bytes signals termination of a transfer.
- Any error packet causes the instantaneous termination of file transfer (for instance if the target file system is full).

A retransmission timer is set every time a block or acknowledgment is sent. TFTP packet headers are described below.

Connection request packets :

```
2 bytes     string    1 byte     string   1 byte
-------------------------------------------------
| Read   |  Filename  |   0  |    Mode    |   0  |
| Request|            |      |            |      |
-------------------------------------------------
 code 01

2 bytes     string    1 byte     string   1 byte
-------------------------------------------------
| Write  |  Filename  |   0  |    Mode    |   0  |
| Request|            |      |            |      |
-------------------------------------------------
 code 02
```

For the two above packets, filename is an ASCII string. Mode field contains the string “netascii” (another mode is “octet” for binary transfers).

Data packets :

```
 2 bytes     2 bytes      n bytes (up to 512)
-----------------------------------------------
| Data blk |   Block #  |         Data         |
-----------------------------------------------
 code 03
```

ACK packets :

```
2 bytes     2 bytes
---------------------
| ACK    |   Block #  |
---------------------
 code 04
```

Error packets :

```
2 bytes     2 bytes      string    1 byte
-----------------------------------------
| Error  | ErrorCode |   ErrMsg   |   0  |
-----------------------------------------
 code 05
```

2 Study of a file transfer

Let us consider the file transfer described in figure below.

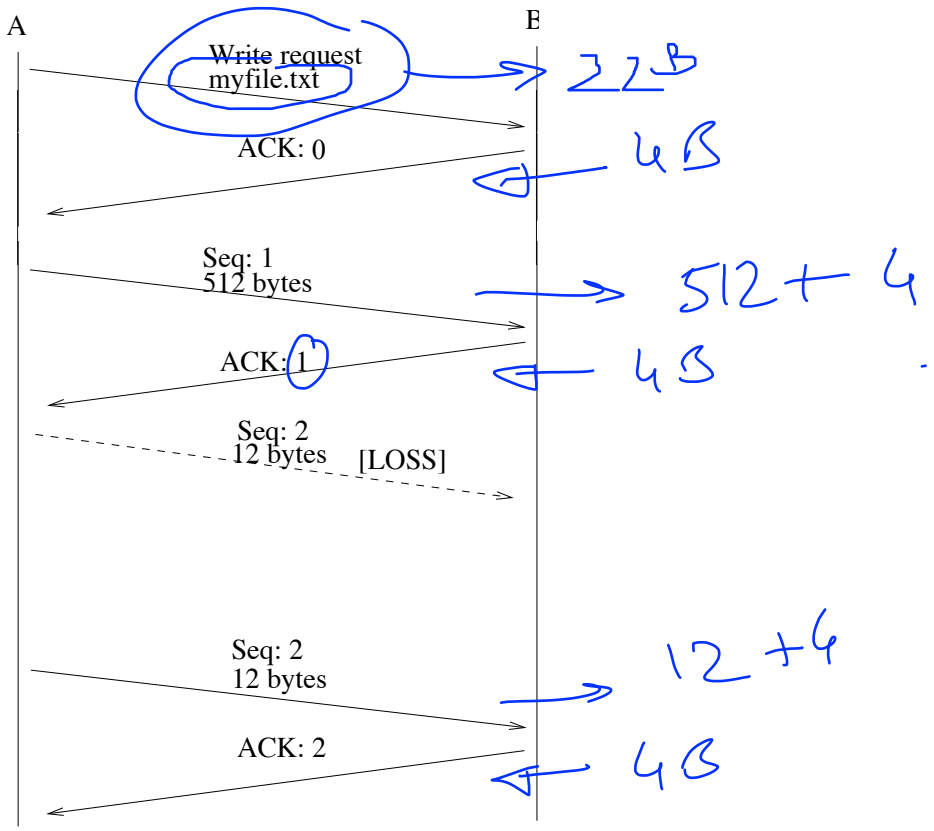


FIGURE 1 –

1. What is the size of the transferred file ?
2. Why does host A re-emit a packet ?
3. What was the exact size of each TFTP packet ?
4. In which case a packet can carry zero-length data ?
5. Why use such a protocol instead of FTP ?

3 Performance

We recall that an IP header is 20 bytes long. A UDP header is 8 bytes. Ethernet preamble 8 bytes. Ethernet frame header 14 bytes with a 32-bits CRC at the end of the frame. Inter-frame gap (IFG) is 9.6μs.

We now assume that A and B are on an 10Mb/s Ethernet LAN with a hub between them. Hub switching time is supposed to be 0.5 μs. Both cables are 50 meters long. Signal propagates at 2.10^8m/s.

6. What is the size of the Ethernet frame for each packet ?
7. Compute the propagation delays.
8. Compute transmission delays for the first data packet.
9. What would be a reasonable timer delay ?

Assume now that no packet is lost ~~in the above transfer~~. and neglect initial handshake + 2 last packets

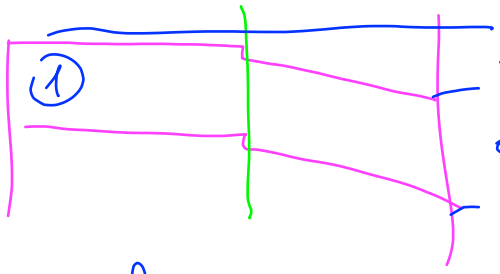
10. What is the average throughput of the file transfer ?
11. What happens in the presence of external traffic on the hub ?
12. What is the throughput if the hub is replaced by a *switch* (*store & forward*) ?

↳ 5125 packets -

6.) 22 + 8 + 20 = 50B (> 46)
TCP UDP IP → 50 + 26 ⇒ 76B
8 + 14 + 4

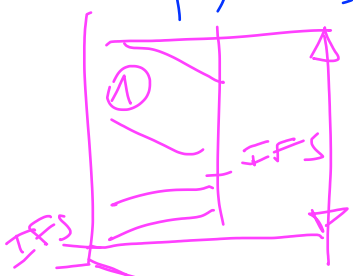
- Ack = 40 + 28 = 32B < 46B
padded ⟹ 46 + 26 = 72B
- Data = 512 + 4 + 28 + 26 = 570B

7.) 0,5µs + 0,5µs = 1µs
↳ cable ↳ hub

8.)


570 × 8 / 10Mb/s = trans time = 456µs

9.) timer delay ? time to receive Ack

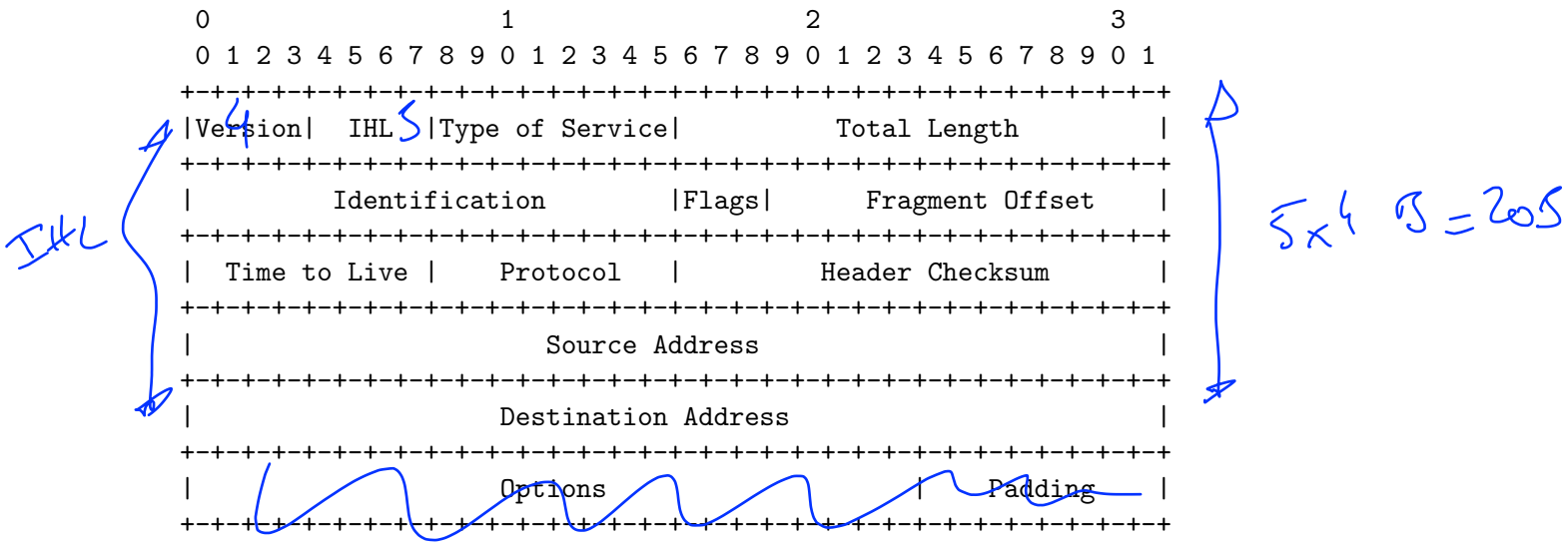


456µs + 1 + 9.6 + 57.6 + 1 = 525.2µs

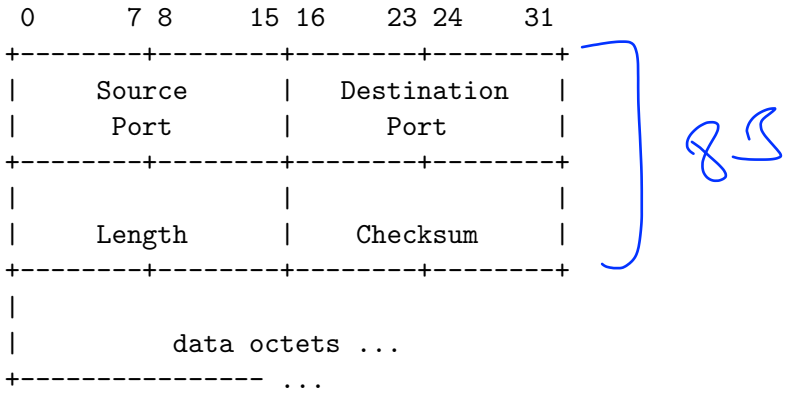
propag in real world ≈ 200 ms (round trip) ≈ 1 ms - 2

⇒ Timer more like 1s

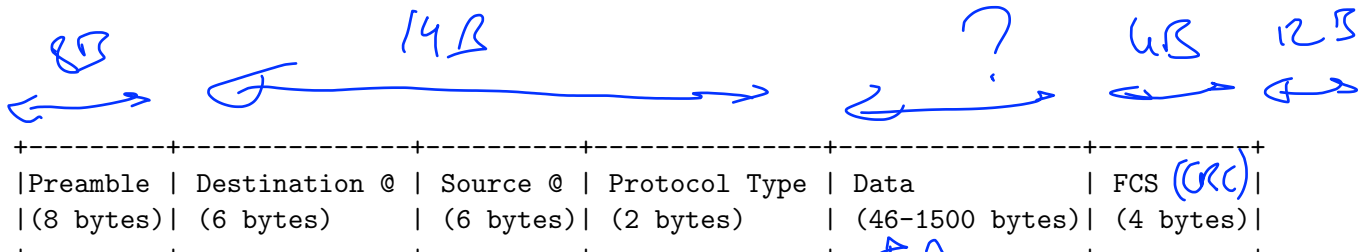
Avg th = data sent / time to send data = 512×8 bits / 525.2+9.6 = 512×8 bits / 534.8 = 7.66 Mb/s

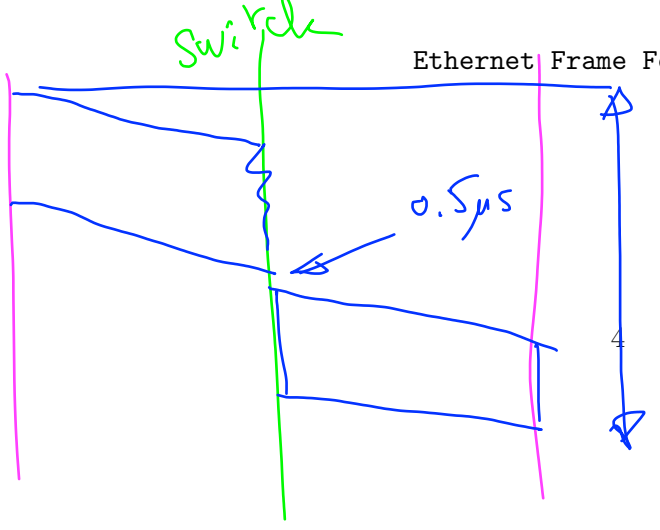


Example Internet Datagram Header



User Datagram Header Format





0,5µs

Propag + 2 trans times

⇒ 1µs + 2 × 570×8 / 10Mb/s

no throughput > half of that with hub !!