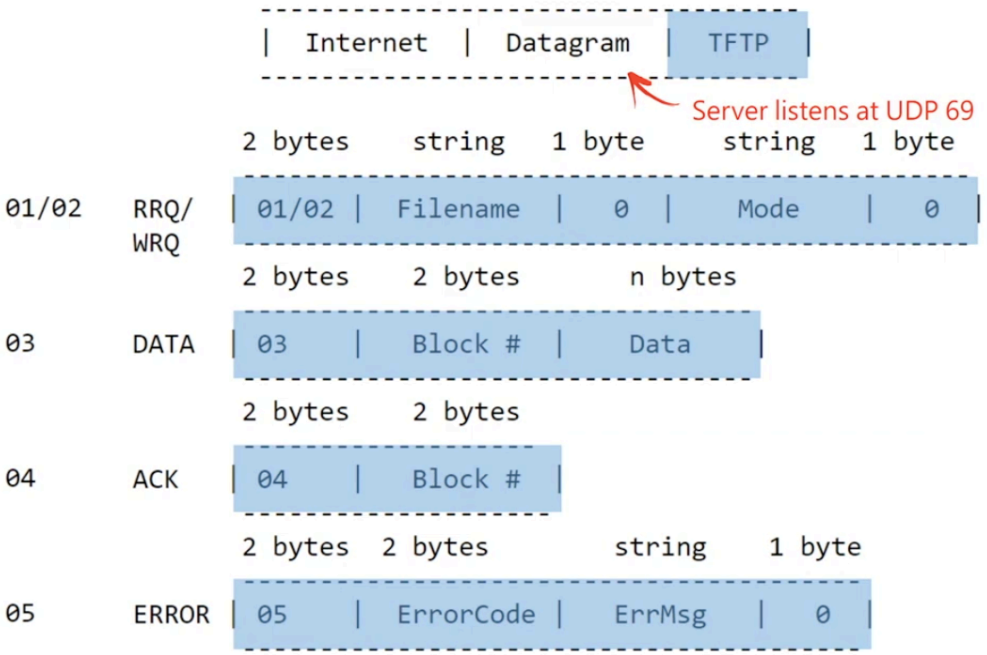
TFTP Introduction

TFTP (Trivial File Transfer Protocol):

簡單檔案傳輸協定
RFC 1350
Layer 7 protocal
Layer 4: UDP（所以不用 handshaking）
TFTP Server: UDP port 69 (default)

TFTP Messages:

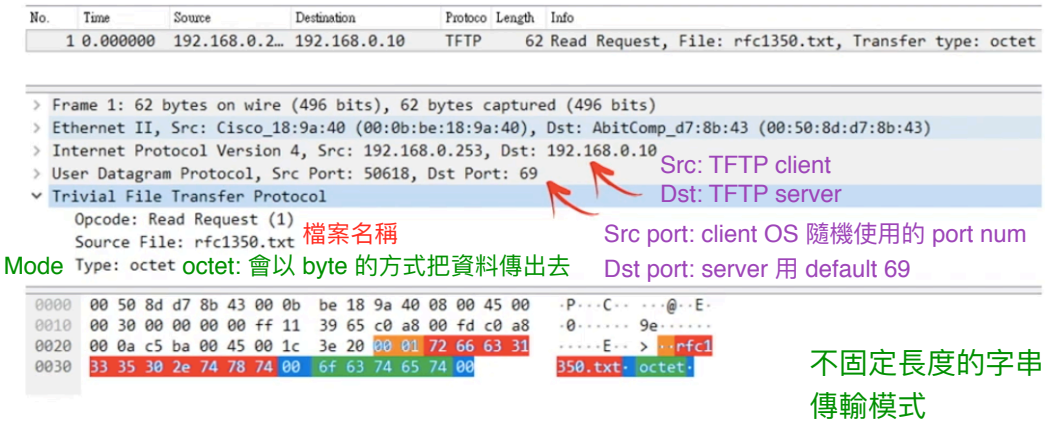


Operation Code	Message
01	RRQ (Read Request) - 從 server 下載檔案
02	WRQ (Write Request) - 上傳檔案至 server
03	DATA - 傳送檔案資料，上傳和下載格式都一樣
04	ACK - acknowledgement
05	ERROR - 回報傳輸錯誤

* TFTP 封包的前 2 個 bytes 是用來辨識封包的種類（operation code）

1. RRQ

當 TFTP client 向 TFTP server 要求下載檔案時，TFTP client 會送出 RRQ 的封包



Src: TFTP client
Dst: TFTP server

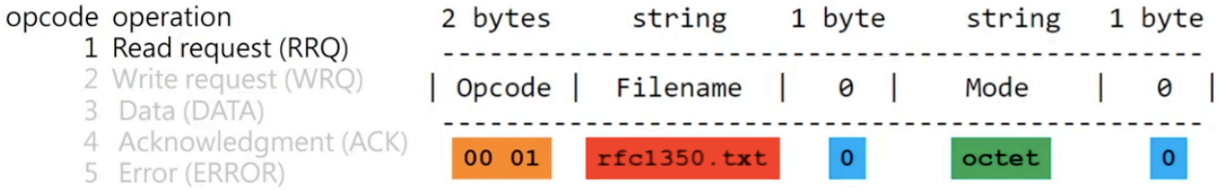
檔案名稱

Src port: client OS 隨機使用的 port num
Dst port: server 用 default 69

Mode 至少有
1. octet
2. netascii
3. mail
但建議用 octet，
因為 octet 可以傳輸任何數位化資料。

Mode octet: 會以 byte 的方式把資料傳出去

不固定長度的字串傳輸模式



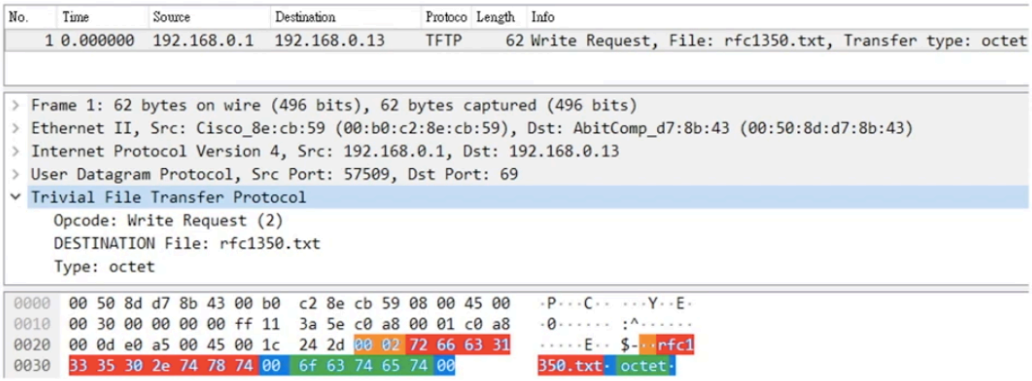
前 2 個 bytes
用來辨識封包的種類
16進位的 0x00 0x01

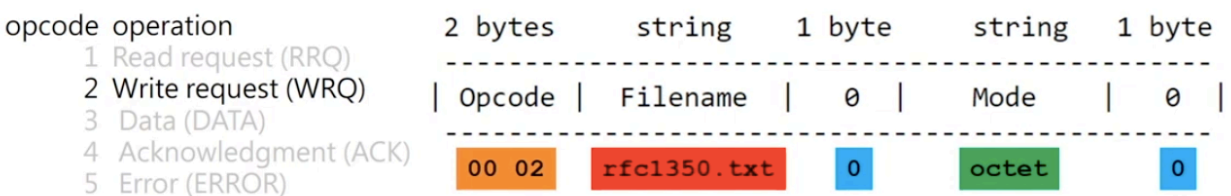
不固定長度的字串
ASCII 編碼
RRQ 下載的檔案名稱

以 1 個 byte 結尾
表示字串結束
0x00

2. WRQ

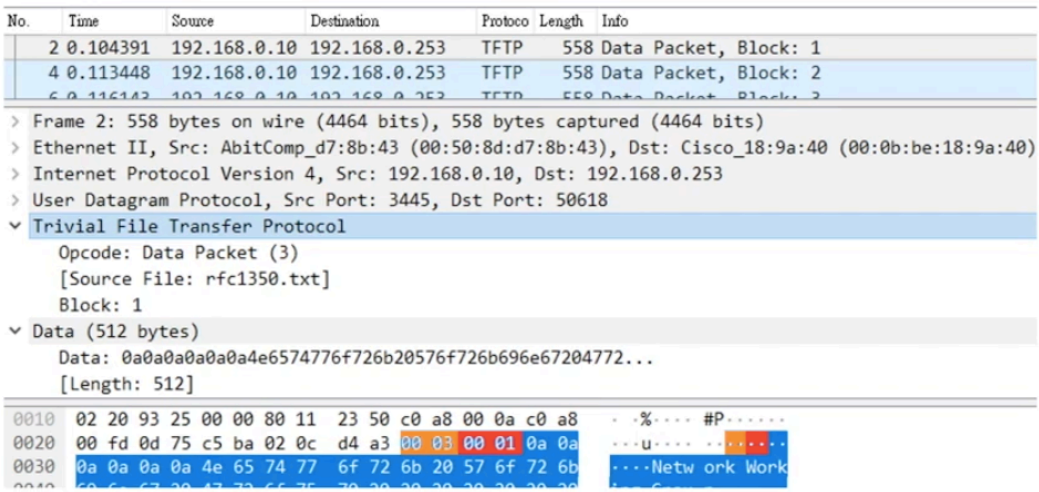
當 TFTP client 要求寫入檔案至 TFTP server，TFTP client 會送出 WRQ 的封包
封包格式和 RRQ 一樣，只差 opcode 為 0x00 0x02

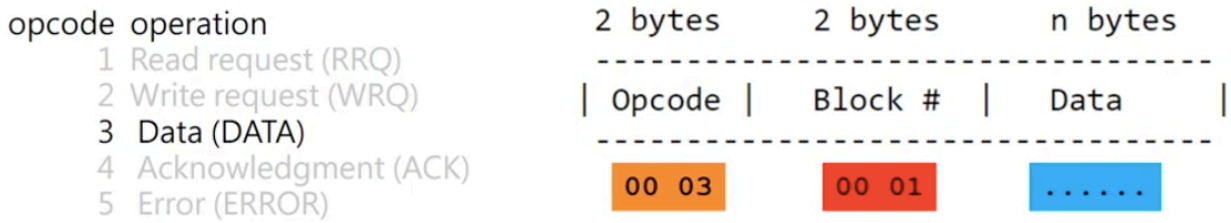




3. DATA

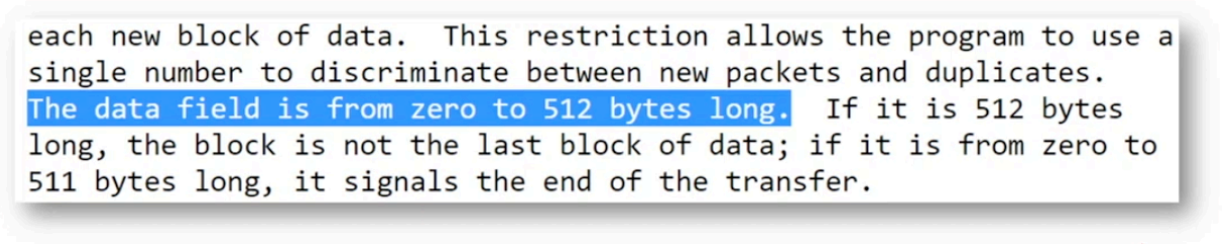
檔案資料如何在 TFTP client 和 server 間傳輸

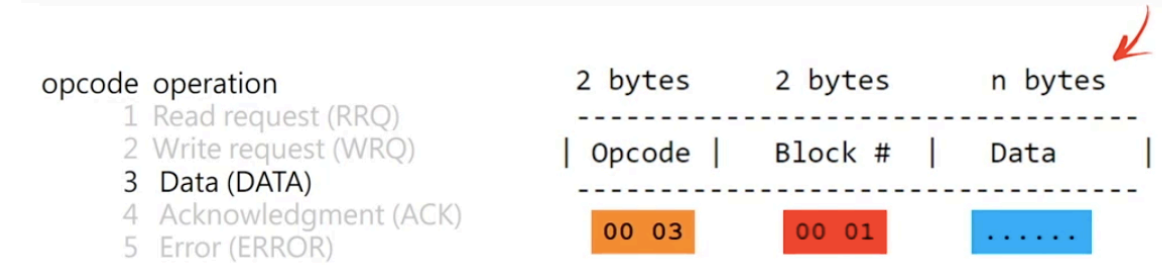




Block:

1. 2 bytes 的整數，表示後面資料區塊的編號（每一個 block 都有自己的編號）
2. 一個大的檔案會被 server 切割成數個 blocks，TFTP client 和 server 會依序傳送每一個 blocks，為了將來能讓 client or server 將多個 blocks 組合回一個大檔案

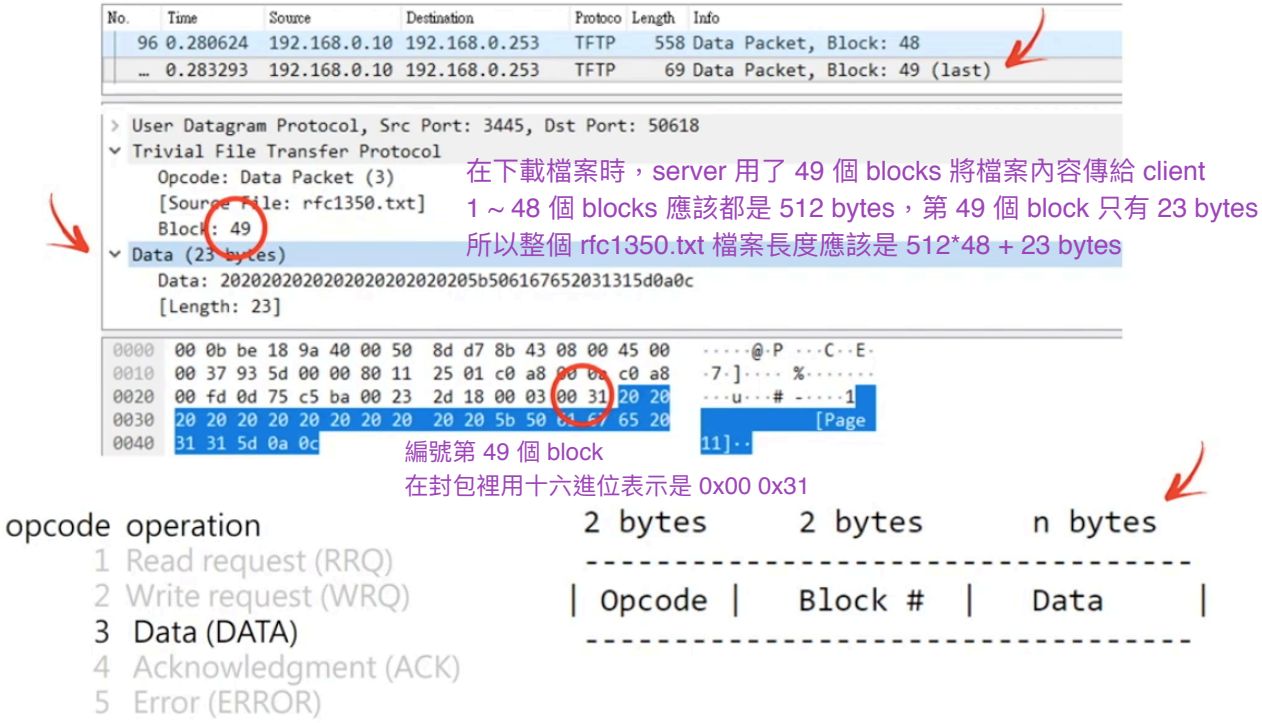




Data:

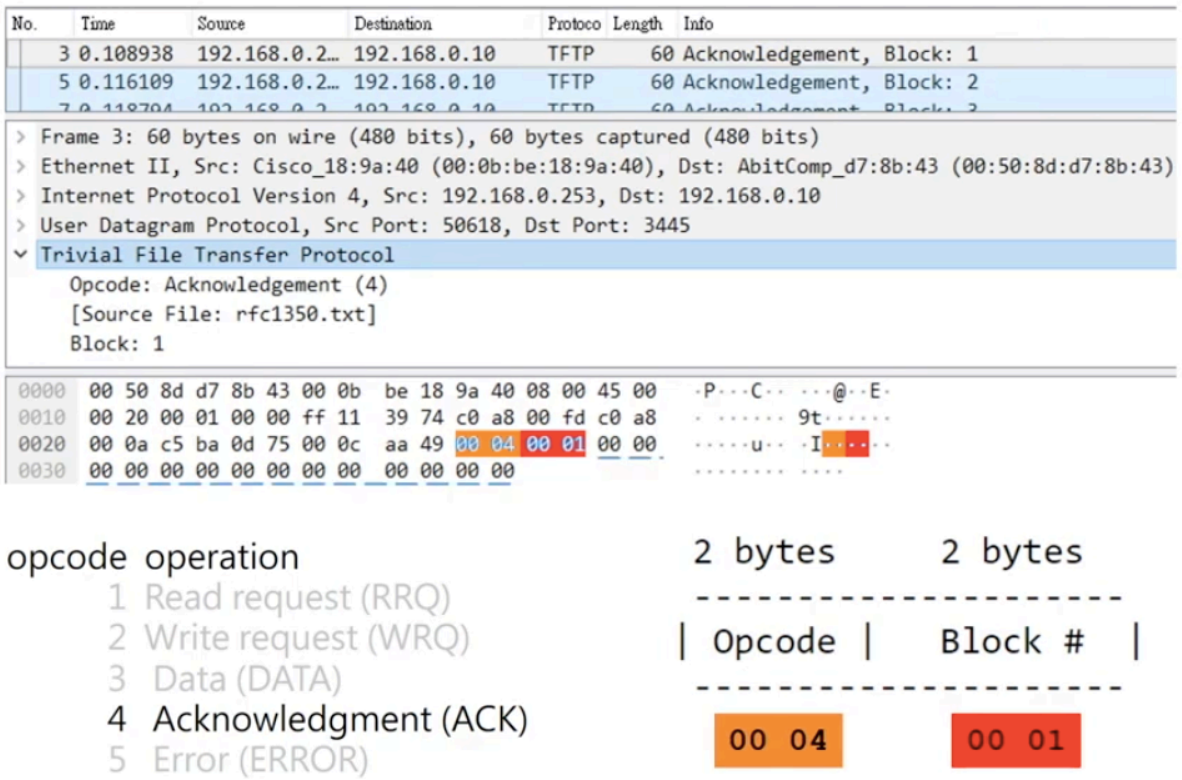
1. 該 block n bytes 的資料內容
2. TFTP 裡沒有 length 來表示 block 的長度，因為 TFTP 的 RFC 裡說這個 data 的長度是 0 ~ 512 bytes
3. 如果一個 DATA 封包後面的 data field 長度是 512 bytes，表示這個 block 後面還會有後續的 block（檔案未傳完）
4. 如果一個 DATA 封包後面的 data field 長度是 0 ~ 511 bytes，表示這個 block 是檔案傳輸的最後一個 block

最後一個 Block:



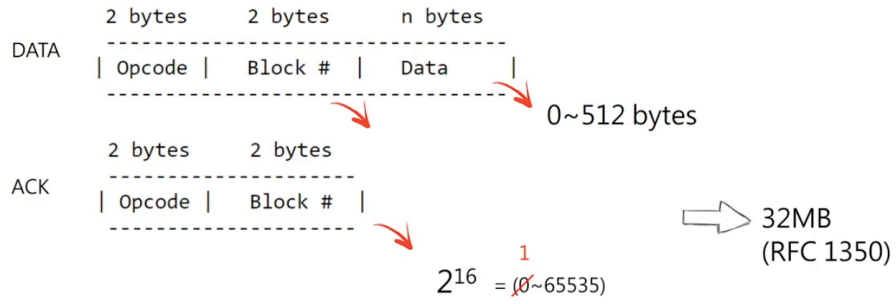
4. ACK

TFTP client 每次從 server 上接收到一個 DATA 時，client 會回傳一個相對應 block number 的 ACK



因為 TFTP 底層是用 UDP，DATA 可能會掉封包，所以 TFTP 就用 block number 來確認資料是否正確、按順序的被傳送和接收。不用 TCP 是因為 TCP 會用比較多記憶體來記錄每個 connection 的狀態，在很小型的嵌入式裝置上 or 在刷新硬體的 firmware 時，OS or bootloader 可能只有幾 MB 的記憶體空間，沒有空間給 TCP 的 implementation。

* TFTP 因為欄位長度而導致的限制



> Data is actually transferred in DATA packets depicted in Figure 5-2. DATA packets (opcode = 3) have a block number and data field. The block numbers on data packets begin with one and increase by one for each new block of data. This restriction allows the program to use a single number to discriminate between new packets and duplicates.

若 data field 最多只能放 512 bytes，且 DATA 和 ACK 的 block number 只有 2 bytes (16 bits)，
也就是說 block number 只能是一個介於 0 ~ (2^16 - 1) 的數字，
但 RFC 1350 說 block number 起始值須為 1，所以實際上只能傳送一個長度約為 32 MB 的檔案。

> The original protocol has a transfer file size limit of 512 bytes/block x 65535 blocks = 32 MB. In 1998 this limit was extended to 65535 bytes/block x 65535 blocks = 4 GB by TFTP Blocksize Option RFC 2348. If the defined blocksize produces an IP packet size that exceeds the minimum MTU at any point of the network path, IP fragmentation and reassembly will occur not only adding more overhead[8] but also leading to total transfer failure when the minimalist IP stack implementation in a host's BOOTP or PXE ROM does not (or fails to properly) implement IP fragmentation and reassembly [9]. If TFTP packets should be kept within the standard Ethernet MTU (1500), the blocksize value is calculated as 1500 minus headers of TFTP (4 bytes), UDP (8 bytes) and IP (20 bytes) = 1468 bytes/block, this gives a limit of 1468 bytes/block x 65535 blocks = 92 MB. Today most servers and clients support block number roll-over (block counter going back to 0 after 65535) which gives an essentially unlimited transfer file size.

RFC 2348 決定把 block size 加大為 65535 bytes，所以一個檔案最多可以傳送到 4 GB，
但 Internet 上的 routers 會因為底層傳輸技術的不同，而對封包進行 fragmentation，
一般來說 MTU 會被設定為乙太網路（Ethernet）的 MTU，也就是 1500 bytes，
小裝置連 TCP 都不想要了，可能也無法處理 IP fragment 的封包，
所以實際上不考慮 fragmentation 的情況下，TFTP 頂多可以傳約 92 MB 的檔案而已，
這種加大 data field size 的策略治標不治本。

所以後來的 RFC option 裡制定了 block number roll-over 的機制：
當 block number 從 1 開始用到 65535 之後，會再從 0 開始繼續使用，就能夠應付大檔案的傳輸了。

5. ERROR

Error Codes

簡易的 TFTP implementation 裡可能只有簡單的列一些錯誤訊息，甚至常常只有使用 0x00 0x00 表示錯誤

Value	Meaning
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.

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

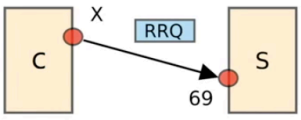
2 bytes	2 bytes	string	1 byte
Opcode	ErrorCode	ErrMsg	0

0x00 0x05　　error message 可有可無　　結尾 0x00

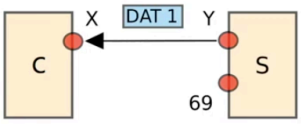
* 下載 (Read)＆上傳 (Write)

Read:

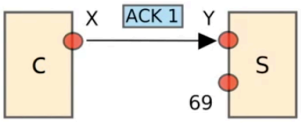
1. client 會隨機使用一個 UDP port number x，向 TFTP server (UDP 69) 傳送 RRQ



2. server 會使用一個新的 random port y，和 x 進行 DATA 和 ACK 的交換
3. 因為 UDP 需要 dest. IP + dest. port，所以 server 要保留 port 69，讓其他 client 聯繫

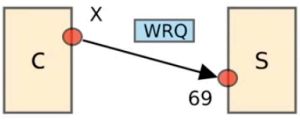


4. 之後 client 收到 DATA 就會回傳相對應的 ACK

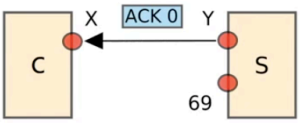


Write:

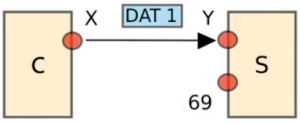
1. client 會隨機使用一個 UDP port number x，向 TFTP server (UDP 69) 傳送 WRQ



2. server 發一個 ACK 0 (block num = 0) 給 client：表示 server 允許 client 開始上傳封包
3. server 和 client 會在 x 和 y 上傳輸



4. client 可以開始上傳第一個 block，server 會回相對應的 ACK
5. block 會不斷接連傳送過去，直到最後一個 data field 小於 512 bytes 的 block 出現，最後一個 ACK 也回成功



Source: https://youtu.be/N9f3WQhf1vQ