Advanced Lecture on Internet Applications
3. File Transfer: TFTP, FTP, Reliable Multicast

Masataka Ohta

mohta@necom830.hpcl.titech.ac.jp

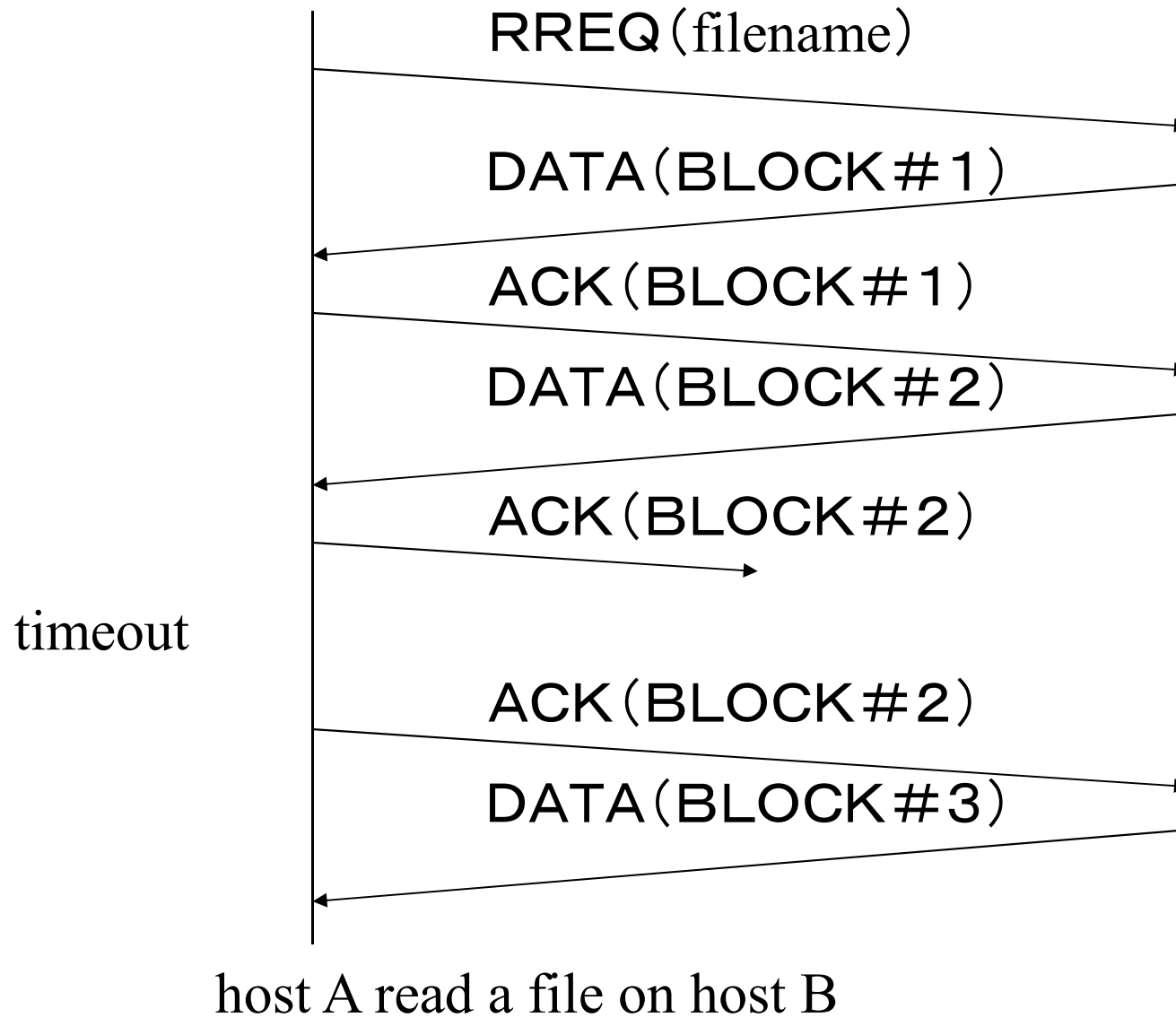ftp://ftp.hpcl.titech.ac.jp/appli3e.ppt

File Transfer

- the most basic application of the Internet
- send all the bits without error
  - retransmission inevitable
    - by application over UDP or by TCP
  - retransmission may be repeated several times
    - certain delay is inevitable
- ACK is useful
  - not usable with multicast

TFTP (Trivial File Transfer Protocol, rfc1350)

- simple file transfer protocol
  - only read/write file
  - no directory listing
  - no user authentication
- constructed over UDP (port 69)
- used for bootstrapping
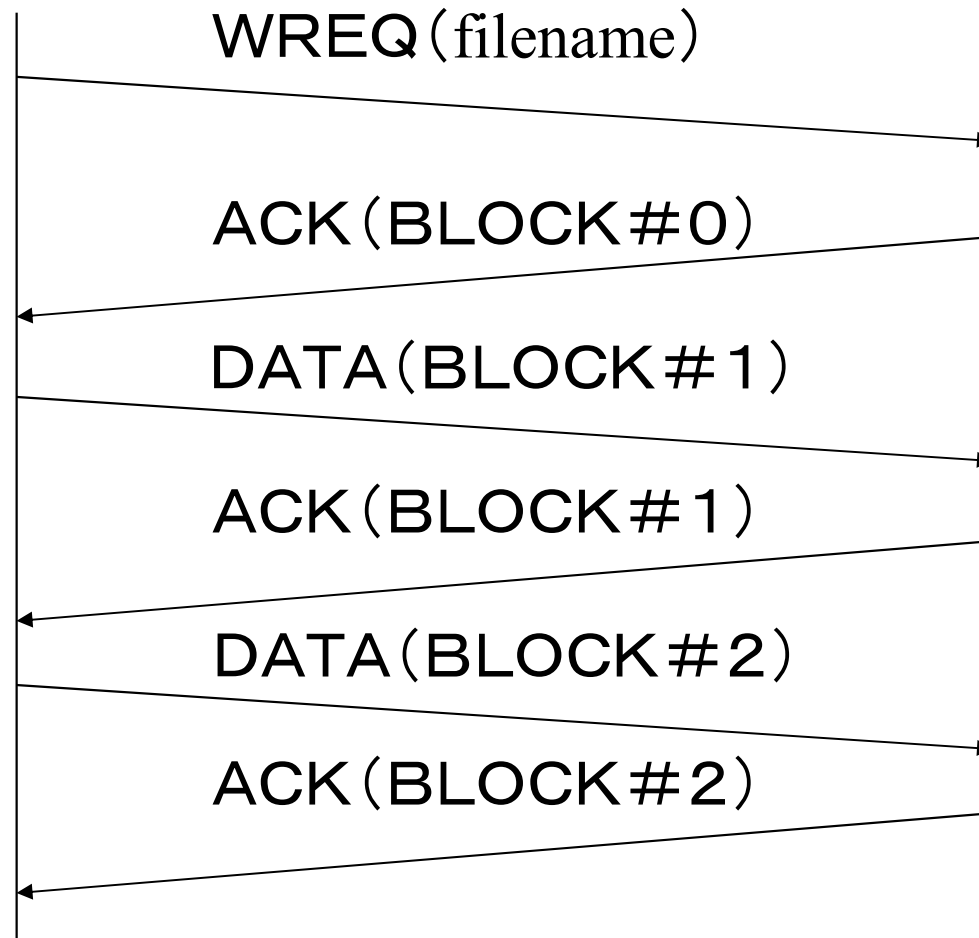- 512B fixed (except for the last) data is sent
  - as ACK is received

host A (at some UDP port)　　　　host B (at UDP port 69)

RREQ（filename）

DATA（BLOCK＃1）

ACK（BLOCK＃1）

DATA（BLOCK＃2）

ACK（BLOCK＃2）

timeout

ACK（BLOCK＃2）

DATA（BLOCK＃3）

host A read a file on host B

host A (at some UDP port)   host B (at UDP port 69)

WREQ（filename）

ACK（BLOCK＃0）

DATA（BLOCK＃1）

ACK（BLOCK＃1）

DATA（BLOCK＃2）

ACK（BLOCK＃2）

host A read a file on host B

Packet Format of TFTP (1)

```
Type    Op #

        2 bytes     string   1 byte     string    1 byte
        ------------------------------------------------------
RRQ/    | 01/02 |  Filename |   0  |     Mode     |   0  |
WRQ     ------------------------------------------------------

        2 bytes     2 bytes        n bytes
        -------------------------------------
DATA    | 03    |  Block # |     Data    |
        -------------------------------------
```

Packet Format of TFTP (2)

```
Type    Op #


        2 bytes     2 bytes
        _____
ACK    | 04     |  Block #  |
        _____
        2 bytes  2 bytes          string      1 byte
        _____
ERROR | 05      |  ErrorCode |   ErrMsg   |   0   |
        _____
```

Errors of TFTP

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.

Retransmission of TFTP

- the current specification retransmit ACK upon timeout

    - timeout period should be determined adaptively

- old specification (rfc783) retransmit by both ends upon timeout

    - packet delay causes duplicated transmission to waste bandwidth

        - wasted bandwidth may result in further packet delays

host A (at some UDP port)   host B (at UDP port 69)

DATA#2

ACK#2   timeout

delay

DATA#2

ACK#2   DATA#3

ACK#3   DATA#3

ACK#3   DATA#4

ACK#4   DATA#4
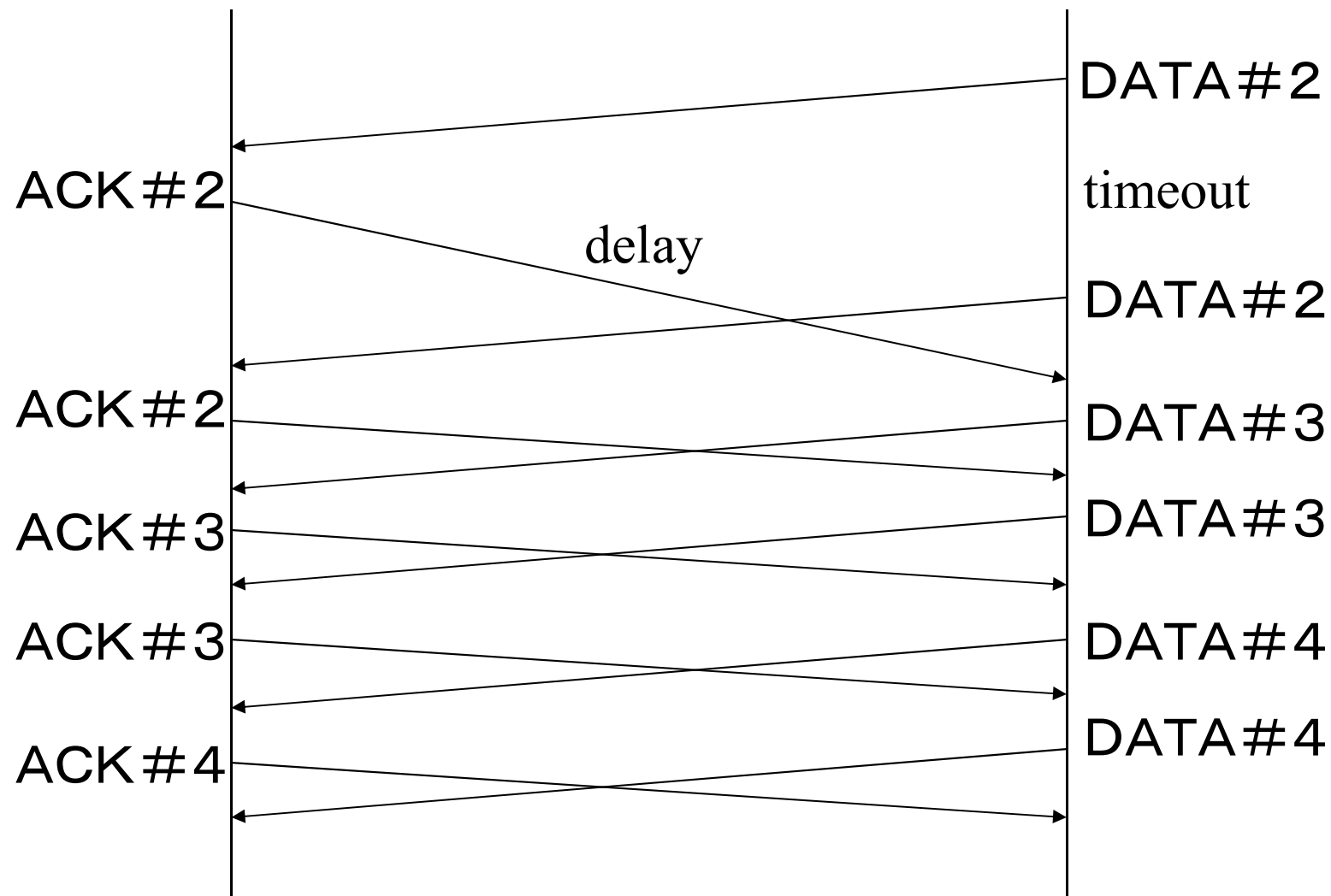
problem caused by retransmission by both ends

FTP (rfc959)

- File Transfer Protocol
- protocol to exchange files over the Internet
- port# 21 is used for command and response
- various file formats are supported

TCP and Command

- commands and replies represented in ASCII strings are exchanged over TCP
  - reply often begins with 3 digits followed by a space and text explaining reply in English

- line is terminated by CR and LF

- data may be sent over the same TCP connection (SMTP) or other TCP connection (FTP)
  - separator for data is necessary for sending over the same TCP

Meaning of 3 Digit Reply Code (1)

- 1yz   Positive Preliminary reply
- 2yz   Positive Completion reply
- 3yz   Positive Intermediate reply
- 4yz   Transient Negative Completion reply
- 5yz   Permanent Negative Completion reply

Meaning of 3 Digit Reply Code (2)

- x0z   Syntax
- x1z   Information
- x2z   Connections
- x3z   Authentication and accounting
- x5z   File system
- "z" gives a finer gradation of meaning in each of the function

Examples of Replies (1)

```
200 Command okay.
500 Syntax error, command unrecognized.
    This may include errors such as command line too long.
501 Syntax error in parameters or arguments.
202 Command not implemented, superfluous at this site.
502 Command not implemented.
503 Bad sequence of commands.
211 System status, or system help reply.
212 Directory status.
213 File status.
214 Help message.
125 Data connection already open; transfer starting.
225 Data connection open; no transfer in progress.
425 Can't open data connection.
226 Closing data connection.
```

Examples of Replies (2)

425 Can't open data connection.

226 Closing data connection.

426 Connection closed; transfer aborted.

227 Entering Passive Mode (h1,h2,h3,h4,p1,p2).

230 User logged in, proceed.

530 Not logged in.

331 User name okay, need password.

332 Need account for login.

532 Need account for storing files.

Examples of Command Sequences (1)

```
LOCAL COMMANDS BY USER              ACTION INVOLVED


ftp (host) multics<CR>          Connect to host S, port L,
                                establishing control connections.
                                <---- 220 Service ready <CRLF>.
username Doe <CR>               USER Doe<CRLF>---->
                                <---- 331 User name ok,
                                          need password<CRLF>.
password mumble <CR>            PASS mumble<CRLF>---->
                                <---- 230 User logged in<CRLF>.
```

Examples of Command Sequences (2)

```
LOCAL COMMANDS BY USER              ACTION INVOLVED


retrieve (local type) ASCII<CR>
(local pathname) test 1 <CR>    User-FTP opens local file in ASCII.
(for. pathname) test.pl1<CR>    RETR test.pl1<CRLF> ---->
                                <---- 150 File status okay;
                                      about to open data
                                      connection<CRLF>.
                                Server makes data connection
                                to port U.


                                <---- 226 Closing data connection,
                                      file transfer successful<CRLF>.
```

Examples of Command Sequences (3)

```
LOCAL COMMANDS BY USER               ACTION INVOLVED


type Image<CR>                       TYPE I<CRLF> ---->
                                     <---- 200 Command OK<CRLF>

store (local type) image<CR>
(local pathname) file dump<CR>       User-FTP opens local file in Image.
(for.pathname) >udd>cn>fd<CR>        STOR >udd>cn>fd<CRLF> ---->
                                     <---- 550 Access denied<CRLF>

terminate                            QUIT <CRLF> ---->
                                     Server closes all
                                     connections.
```
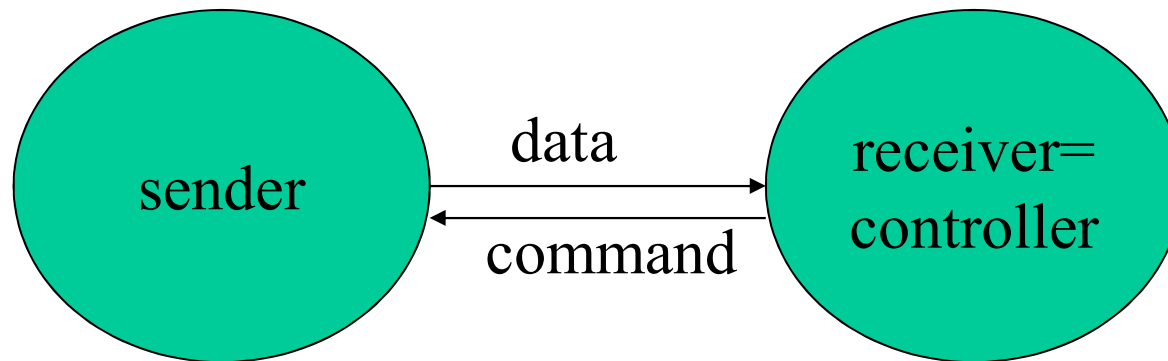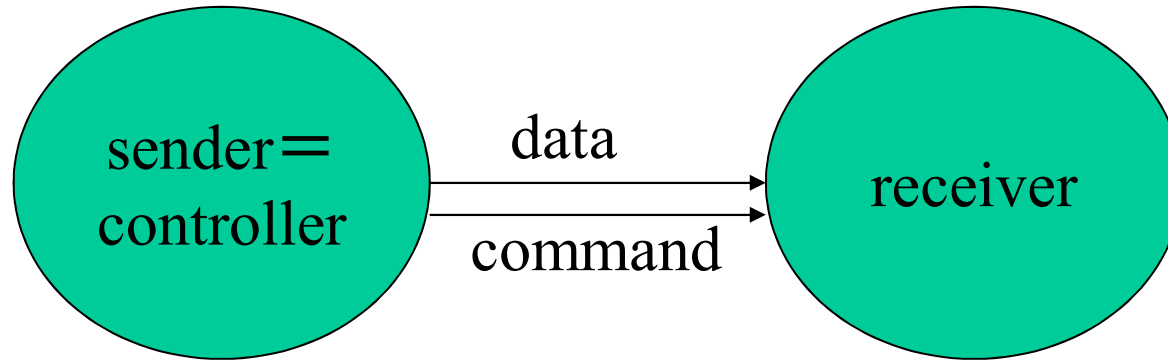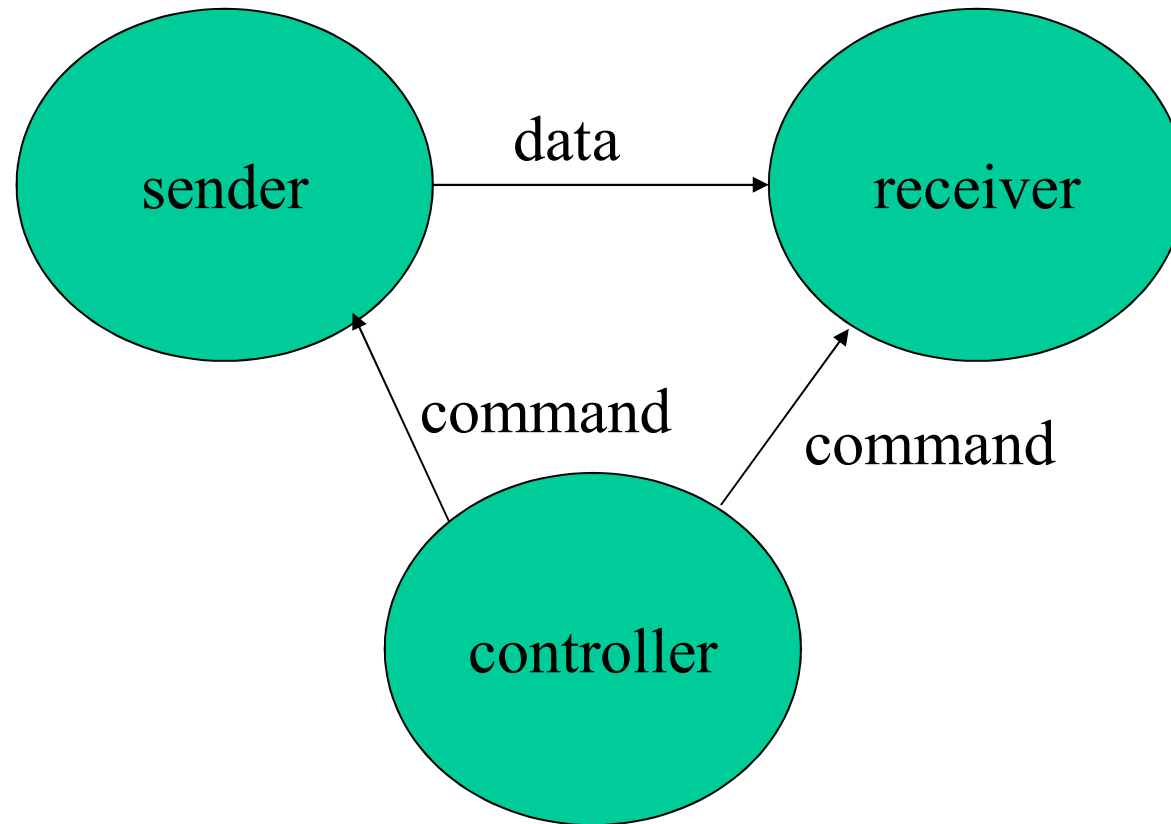
What are Ends of File Transfer?

- sender
- receiver
- controller!!

sender＝
controller   → data →   receiver
              → command →

sender   ← data ←   receiver＝
          ← command ←   controller

usual file transfer

more general form of file transfer

Commands of FTP

- various
  - USER PASS ACCT CWD CDUP SMNT QUIT REIN PORT PASV TYPE STRU MODE RETR STOR STOU APPE ALLO REST RNFR RNTO ABOR DELE RMD MKD PWD LIST NLST SITE SYST STAT HELP NOOP

Some Commands of FTP (1)

- USER
  - user name
- PASS
  - password

Some Commands of FTP (2)

- PORT
  - specify IP address and port number to connect
- PASV
  - ask IP address and port number of server waiting for connection
- TYPE STRU MODE
  - file format

Some Commands of FTP (3)

- RETR
  - read from a file
- STOR
 - write to a file

What is Multicast?

- one to many, many to many communication by copying data in network
    - "broadcast" by network

- necessary in network not possible by end
    - copying end is called reflector or, IMHO improperly, application layer multicast

- intimately related to resource reservation
    - cannot adjust BW according to congestion
    - each multicast address consumes routing table entry

Broadcast Network

- Network to Transfer Voice/Image to Many in Realtime
  - Allocate bandwidth for the transfer
  - Minimize delay
- Wide Area One to Many Communication over Radio Waves
  - Broadcast/Multicast
- Protected by Broadcast Act

Multicast and Broadcast

- broadcast
  - send to all the hosts within a region
  - not realistic over the entire internet
- multicast
  - send to all the members of a group
    - # of members can be arbitrary large
      - member management by network impossible
      - members tell network their existence
  - group is identified by multicast address
    - 224.0.0.0~239.255.255.255
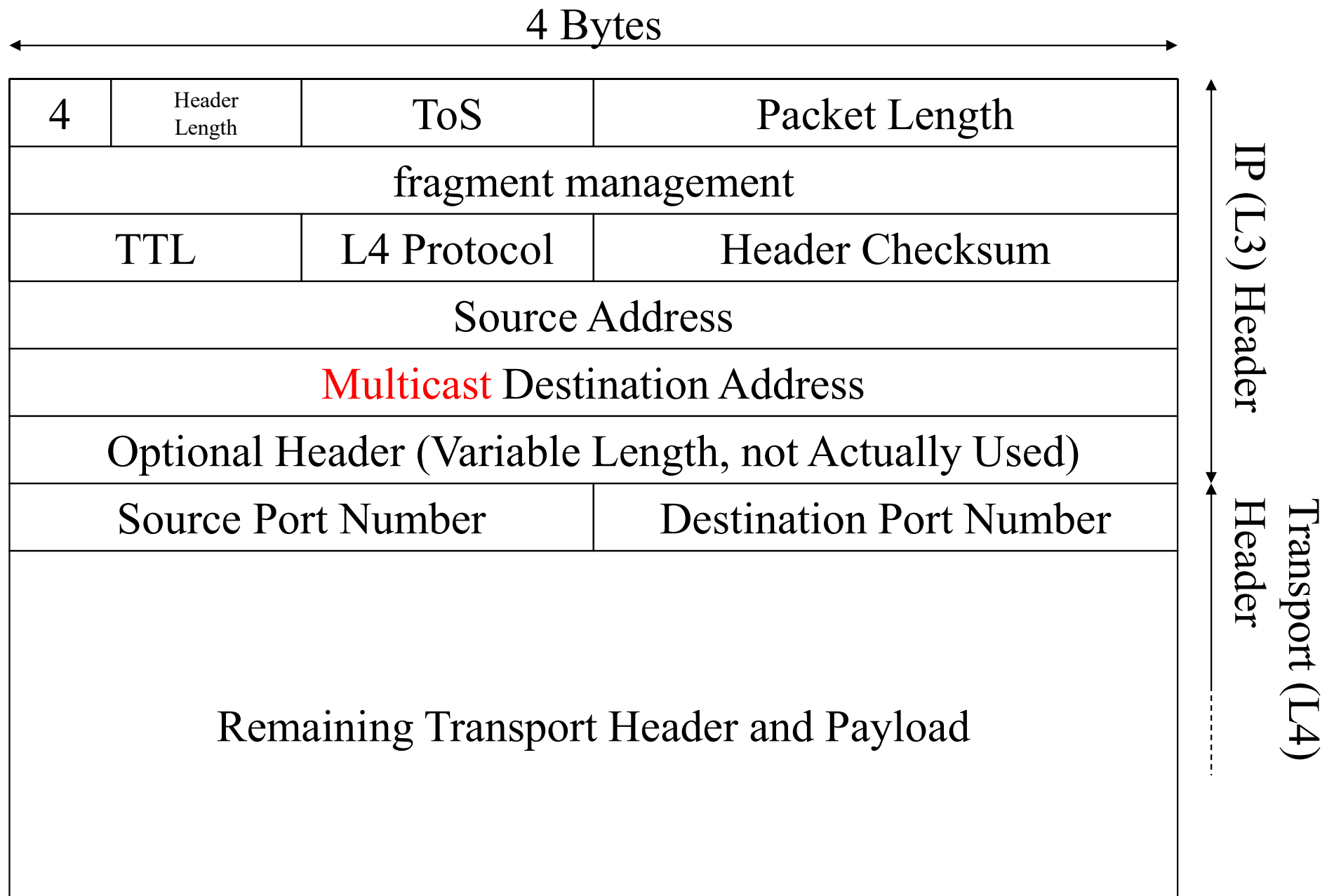
Format of IPv4 Packets (rfc791)

Reliable Multicast

- transfer data without error or loss by multicast
  - in a sense, file transfer
  - can not return ACK by so many receivers
- must repeat transmission without ACK
  - no different from transmitting TV program list
- real time reliable multicast is as meaningless as real time TCP
  - retransmission means delay

Wrap Up

- a bit of error or loss is not allowed in file transfer
  - retransmission with ACK (with UDP/TCP)
- TFTP is light weight
- FTP supports various operations
  - command sequences and replies are sent over TCP
- reliable multicast
  - must repeat transmission of same data