TFTP/FTP

File Transfer and Access: provide computers with ability to access files on remote machines

**TFTP**

TFTP Features
- Read and write files/to remote computers
- Minimal overhead (no security)
- Designed for UDP
- Easy to implement
- Small — possible to include in firmware
- Often uses to bootstrap workstations and network devices
- No Access Control / No Directory Retrieval

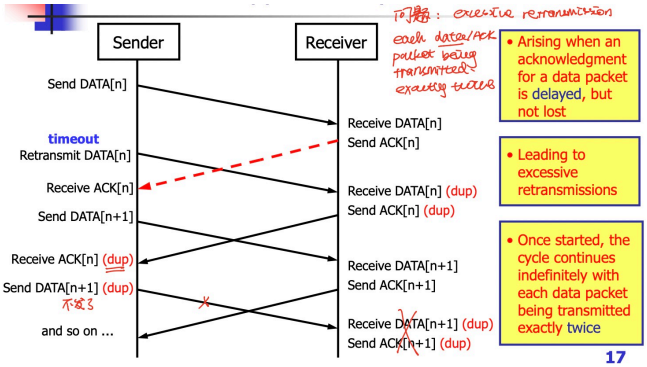
TFTP Transfer Mode
Netascii — for transferring text files
- All lines end with \r\n
- Provides standard format for transferring text files
- Both ends responsible for converting to/from netascii format

Octet — for transferring binary files
- No translation done

TFTP Retransmission
Each side implement the timeout and retransmission:
1. If a data packet get lost in the network, the data sender time out and retransmits the last data packet
2. If an acknowledgement is lost, the acknowledgment sender retransmits the last acknowledgment

- The sender has to keep just one packet on hand for retransmission, since the stop and wait mechanism guarantees that all older packets have been received
- Duplicate data packets must be recognized (ignored) and acknowledgment retransmitted

**SAS (Sorcerer's Apprentice Syndrome)**




SAS产生原因: ACK for a data packet is delayed but not lost
SAS problem: excessive retransmission — each data/ACK packet being transmitted exactly twice after the delay

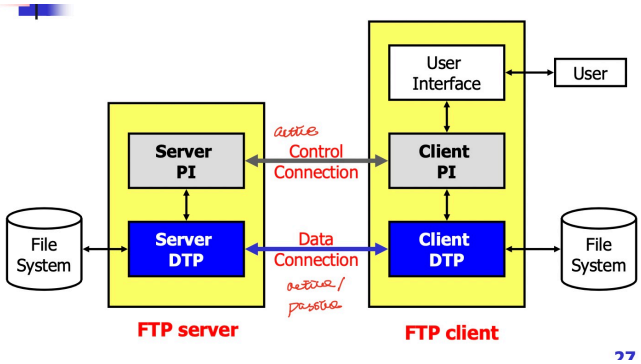
How to fix SAS
Principle: break the retransmission loop
原理: sender should not resend a data packet in response to a duplicate ACK
解决方案: If sender receives ACK[n] — don't send DATA[n+1] is the ACK was a duplicate

TFTP Summary
- UDP **port 69**
- Used with BOOTP and DHCP Configuration applications
- Fixed length blocks —> allocation straight forward (<512 signals the end of the file)
- stop and wait mechanism —> reliable delivery, flow control, no need to reorder data

**FTP**

FTP Features
- Used to transfer files between hosts
- Used to manipulate files: list, delete, rename
- Uses TCP for reliable transfers

FTP model




FTP Client:
- Users interact with FTP client directly
- Active open of control connection, sends command and receives replies
- Control connection uses ASCII plain-test
- Data connection — transfer file data

FTP Server:
Control connection **port 21**
- Receives commands and sends replies

PI (Protocol Interpreter): user and server sides fo the protocol interpretation, interpret's the user's commands

DTP (Data Transfer Process): establishes and manages the data connection (passive or active)

Protocol Interpreter — interprets the user's commands

| User Command | FTP Control Command |
|---|---|
| *cd* mystuff | CWD mystuff |
| *get* dns.pdf | RETR dns.pdf |
| *put* rfc1123.txt | STOR rfc1123.txt |
| *dir* | NLST |

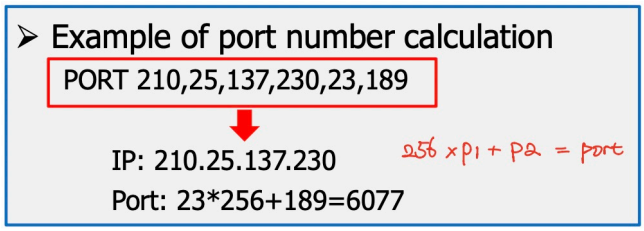
FTP Control Connection & Data Connection
Control Connection
- Remain alive as long as the user keeps the FTP session active
- Passing commands and replies

Data Connection
- Be created dynamically when needed, one data connection persists for one file transfer
- Used for data transmission

**FTP Control Commands**
Three command groups:
- Access control
- Transfer parameter
- Service

Access control group
- USER
- PASS
- ACCT
- CWD
- CDUP — change to parent directory
- REIN — reinitialize
- QUIT

Transfer parameter
Define data connection port:
PORT h1,h2,h3,h4,p1,p2 — (active mode), telling FTP server the port number of client to accept data connection
- FTP server establish data connection




PASV — (passive mode), inform server that client will contact to set up data connections

The 2 systems use different ways to represent text and data
TYPE (type-code)
Typical type-code: A for ASCII — text files, I for image — binary files

The 2 systems may store files in different directory structures
Define file structure: STRU (structure-code)
F — file
R — record
P — page

Define file mode: MODE (mode-code)
S — stream
B — block
C — compressed

Service group
RETR, STOR, APPE, DELE, MKD, RMD, RNFR/RNTO, LIST, ...

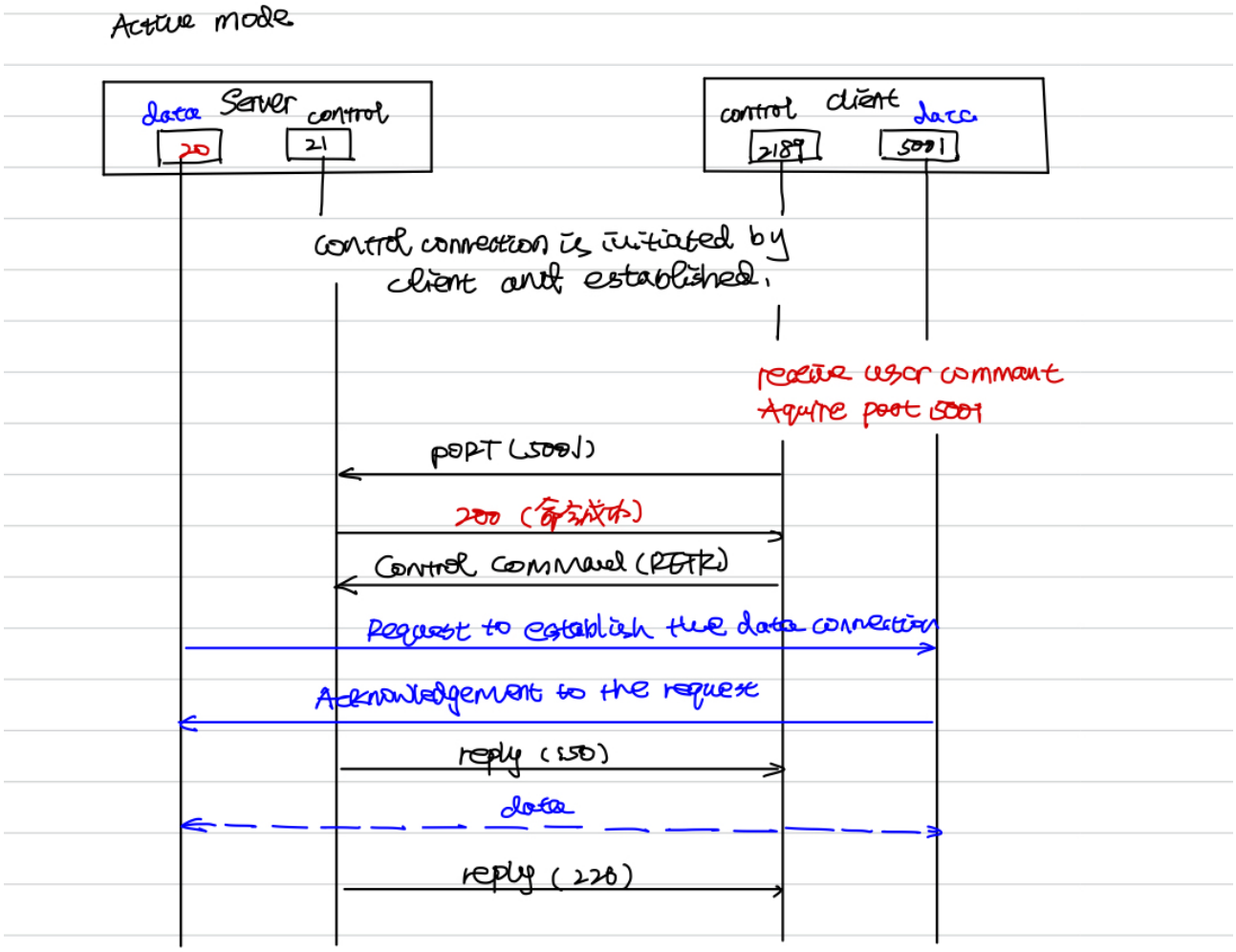
**FTP Control Replies**

- 3 digit code followed by delimiter and text message
- Delimiter: “-“ — not last line ; “ ” — last line
- Numeric code for client program, text for humans

Reply code meanings
1** positive preliminary
2** positive completion
3** positive intermediate
4** transient negative
5** permanent negative

*0* syntax
*1* information
*2* connection information
*3* authentication/accounting
*5* file system

FTP Active Mode
Active mode: server port for data connection = 20



FTP Passive Mode
Passive Mode: server port for data connection > 1024