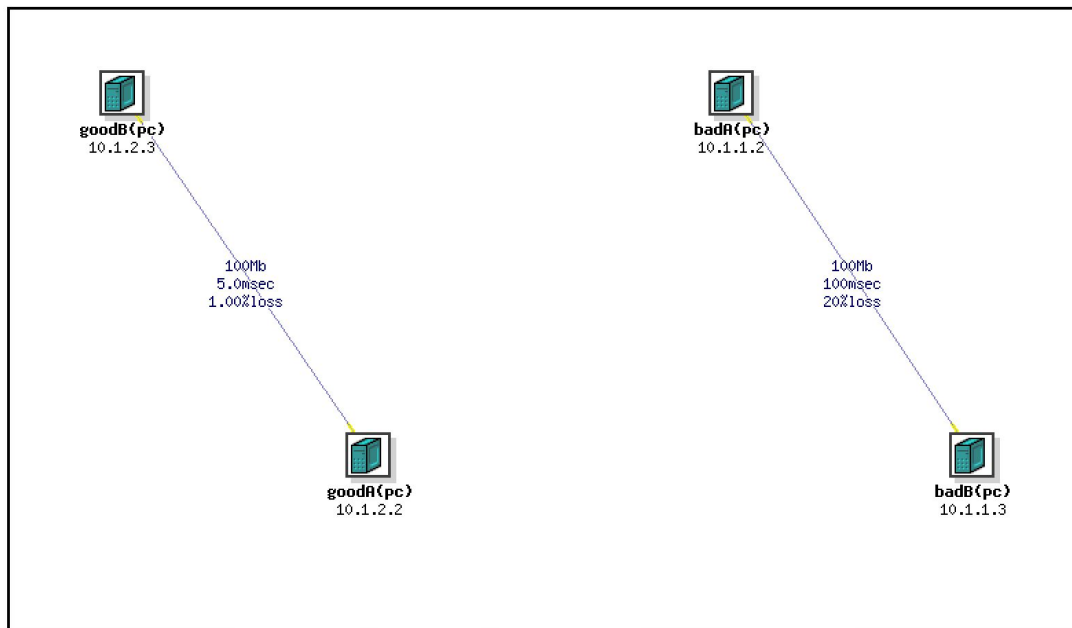# Fast, Reliable File Transfer

EE 542: Cloud Computing
Professor Cho
Lab Assignment 5
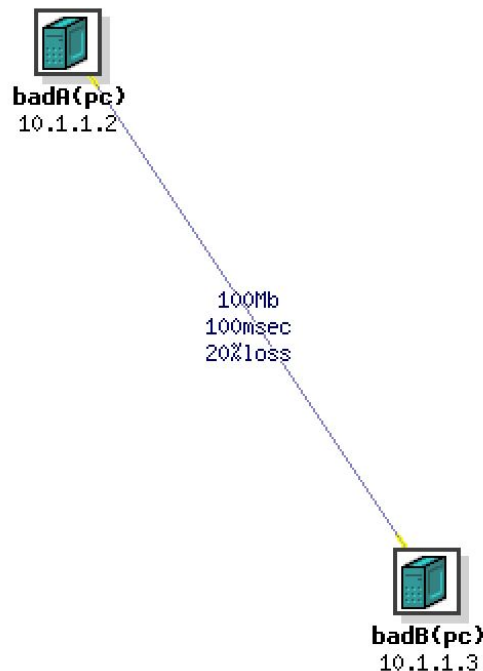
Ali Afzal
Spencer McDonough
Zaine Pazooki

Introduction

File transfer protocols which primarily use TCP as their transport protocol are extremely unreliable over lossy links.

For example, on our emulated link (with 200 ms RTT and 20% loss) scp was not able to transfer files around one gigabytes


```
usc542ee@bada:/tmp$ sudo scp data1G.bin usc542ee@10.1.1.3:/tmp
usc542ee@10.1.1.3's password:
data1G.bin                               0% 2208KB    8.9KB/s - stalled -
```

File Transfer Utility

TCP file transfers are inherently encumbered on lossy links (because of TCP's over emphasis on the reliability of transferring each packet and the exponential window back-off in event of packet loss which has a 1-in-5 probability in our link). Therefore, we chose to only rely on UDP for our protocol.

UDP in itself also cannot transfer files in a reliable fashion. We experienced irretrievable packet losses without building in our own lightweight reliability mechanism. For the socket code, we incorporated the UDP packet code from Lab 2 exercises. To ensure connection and delivery, we defined different structures and built-in checks, expanded on in the next slide.

data_size specifies the size of each chunk (the file is broken into chunks to be sent more efficiently, more on this later). Since Ethernet breaks down each packet into 1500-byte datagrams, we chose to make the process more efficient by preemptively dividing the file ourselves.
- The last data packet is normally smaller, so a different size is allocated

Init packet is used to perform a simple handshake to acknowledge the existence of the receiver.

ack_packet is an int-array which acts as an accumulated ack packet for ensuring reliable delivery of all packets.
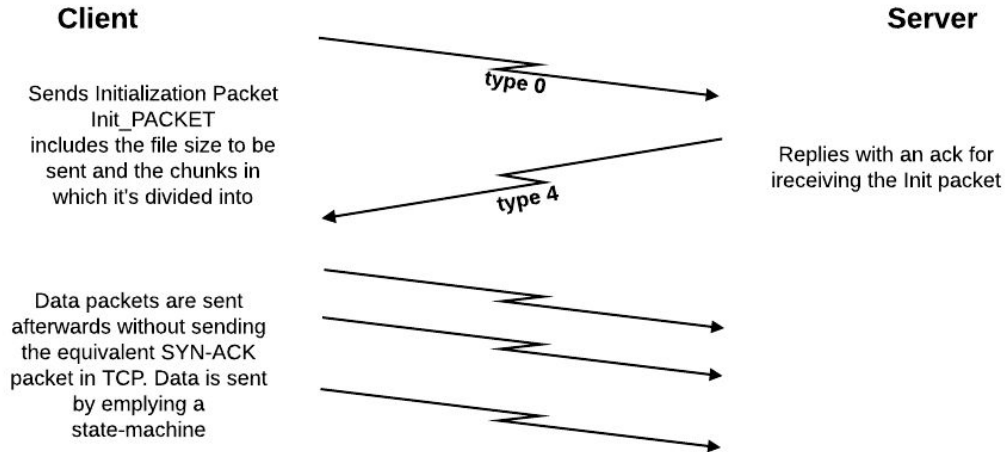
```
#define data_size 1400

struct packet{
        uint8_t type;
        int sequence_number;
        char data[data_size];
};


struct Init_PACKET{
        uint8_t type;
        u_int file_size;
        u_int chunk_size;
};


struct ack_packet{
        uint8_t type;
        uint8_t packet_tracker[20000];
};
```

Client   **Server**

Sends Initialization Packet
Init_PACKET
includes the file size to be
sent and the chunks in
which it's divided into

type 0

type 4

Replies with an ack for
ireceiving the Init packet

Data packets are sent
afterwards without sending
the equivalent SYN-ACK
packet in TCP. Data is sent
by emplying a
state-machine

Packet Types

To make distinctions between packets easier, we specified each packet structure with a type, which is enumerated below

```
type 0: client --> server: init_packet
type 1: client --> server: normal-sized data packets
type 2: client --> server: client request server's missing packet (ack) sequence
type 2: server --> client: server sends updated missing packet (ack) sequence
type 3: server --> client: receiver ACKs everything except last packet
type 4: server --> client: ACK - init_packet receipt
type 5: server --> client: entire transaction done
type 6: client --> server: final data packet
```

Reliable UDP File Transfer State
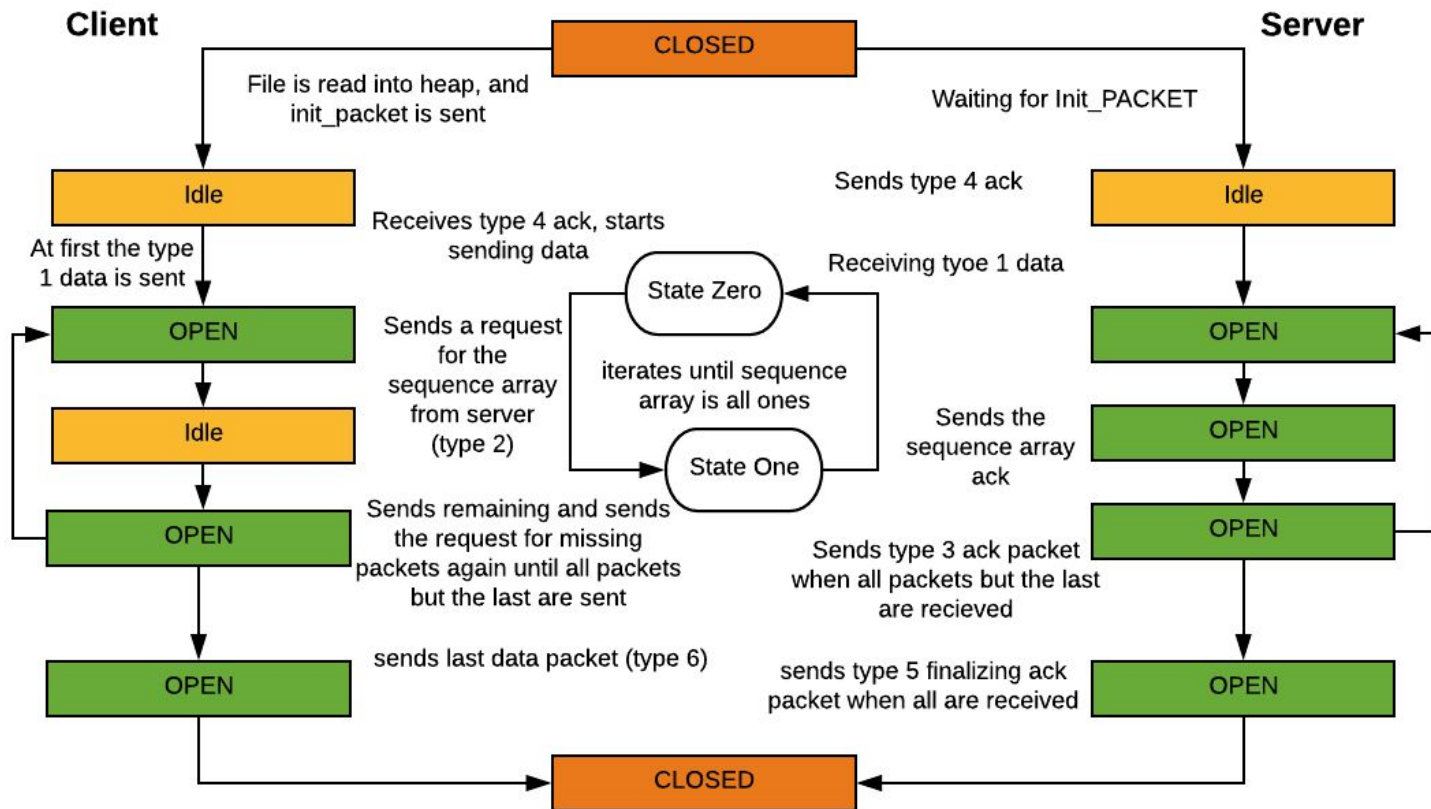
Results - 65MB Good Link

Client.c output:

65MB over 1% loss link

Delay = 0.118s

→ Throughput = 65/0.118s = 550MBps

```
usc542ee@goodb:/tmp$ ls -la
total 206156
drwxrwxrwxt  7 root     root         4096 Sep 13 23:30 .
drwxr-xr-x 27 root     root         4096 Jul 20  2017 ..
-rwxr-xr-x  1 usc542ee EE542        18304 Sep 13 23:30 client
-rw-r--r--  1 usc542ee EE542        12016 Sep 13 23:30 client.c
-rw-r--r--  1 usc542ee EE542         9668 Sep 13 13:46 client_udp.c
-rw-r--r--  1 usc542ee EE542        20480 Sep 13 13:54 .client_udp.c.swn
-rw-r--r--  1 usc542ee EE542        20480 Sep 13 13:52 .client_udp.c.swo
-rw-r--r--  1 usc542ee EE542         4096 Sep 13 13:50 .client_udp.c.swp
-rw-r--r--  1 usc542ee EE542    104857600 Sep 13 23:29 data100M.bin
-rw-r--r--  1 usc542ee EE542      1048576 Sep 13 14:11 data65.bin
-rw-r--r--  1 usc542ee EE542    104857600 Sep 13 13:49 data.bin
drwxrwxrwt  2 root     root         4096 Sep 13 13:36 .font-unix
drwxrwxrwt  2 root     root         4096 Sep 13 13:36 .ICE-unix
-rw-r--r--  1 usc542ee EE542         3806 Sep 13 23:31 log_client.txt
drwxrwxrwt  2 root     root         4096 Sep 13 13:36 .Test-unix
drwxrwxrwt  2 root     root         4096 Sep 13 13:36 .X11-unix
drwxrwxrwt  2 root     root         4096 Sep 13 13:36 .XIM-unix
usc542ee@goodb:/tmp$ ./client
Usage: no filename provided
usc542ee@goodb:/tmp$ ./client data65.bin
File Size: 1048576
No of shunks : 749
Max file size : 1048576
Memory read : 1047200
Memory left to read : 1376
count value  : 748
File Size read: 1376, count : 748
Init packet sent
Init packet no : 0

Ack of init received
Started timer!
type 4
type 4
type 2
type 2
type 3
all packets received except last
Packet Type 5 (cummulative ack) received :
type last 3
all packets sent
Total Transfer Time: 0.117655 seconds
usc542ee@goodb:/tmp$
```

Results - 1GB Good Link

Client.c output:
1000MB over 1% loss link
Protocol failure - Segmentation fault

```
usc542ee@goodb:/tmp$ ls -la
total 1231224
drwxrwxrwt  7 root      root         4096 Sep 13 23:43 .
drwxr-xr-x 27 root      root         4096 Jul 20  2017 ..
-rwxr-xr-x  1 usc542ee EE542        18304 Sep 13 23:43 client
-rw-r--r--  1 usc542ee EE542        12018 Sep 13 23:43 client.c
-rw-r--r--  1 usc542ee EE542         9668 Sep 13 13:46 client_udp.c
-rw-r--r--  1 usc542ee EE542        20480 Sep 13 13:54 .client_udp.c.swn
-rw-r--r--  1 usc542ee EE542        20480 Sep 13 13:52 .client_udp.c.swo
-rw-r--r--  1 usc542ee EE542         4096 Sep 13 13:50 .client_udp.c.swp
-rw-r--r--  1 usc542ee EE542    104857600 Sep 13 23:29 data100M.bin
-rw-r--r--  1 usc542ee EE542   1048576000 Sep 13 23:35 data1G.bin
-rw-r--r--  1 usc542ee EE542      1048576 Sep 13 14:11 data65.bin
-rw-r--r--  1 usc542ee EE542    104857600 Sep 13 13:49 data.bin
drwxrwxrwt  2 root      root         4096 Sep 13 13:36 .font-unix
drwxrwxrwt  2 root      root         4096 Sep 13 13:36 .ICE-unix
-rw-r--r--  1 usc542ee EE542        65536 Sep 13 23:40 log_client.txt
drwxrwxrwt  2 root      root         4096 Sep 13 13:36 .Test-unix
drwxrwxrwt  2 root      root         4096 Sep 13 13:36 .X11-unix
drwxrwxrwt  2 root      root         4096 Sep 13 13:36 .XIM-unix
usc542ee@goodb:/tmp$ ./client data1G.bin
File Size: 1048576000
No of shunks : 748983
Max file size : 1048576000
Memory read : 1048574800
Memory left to read : 1200
count value  : 748982
File Size read: 1200, count : 748982
Init packet sent

Ack of init received
Started timer!
Segmentation fault (core dumped)
usc542ee@goodb:/tmp$ 
```

Results - 65MB Bad Link

Client.c output:
65MB over 20% loss link
Delay = 0.2.564s
→ Throughput = 65/2.564s = 25.35 MBps

```
usc542ee@badb:/tmp$ gcc client.c -lm -o client
usc542ee@badb:/tmp$ ./client data65.bin
File Size: 1048576
No of shunks : 749
Max file size : 1048576
Memory read : 1047200
Memory left to read : 1376
count value  : 748
File Size read: 1376, count : 748
Init packet sent

Ack of init received
Started timer!
all packets received except last
Packet Type 5 (cummulative ack) received :
type last 5
all packets sent
Total Transfer Time: 2.563930 seconds
usc542ee@badb:/tmp$
```

Results - 1G Bad Link

Client.c output:
1000MB over 10% loss link
Protocol failure - Segmentation fault

```
usc542ee@badb:/tmp$ gcc client.c -lm -o client
usc542ee@badb:/tmp$ ./client data65.bin
File Size: 1048576
No of shunks : 749
Max file size : 1048576
Memory read : 1047200
Memory left to read : 1376
count value  : 748
File Size read: 1376, count : 748
Init packet sent

Ack of init received
Started timer!
all packets received except last
Packet Type 5 (cummulative ack) received :
type last 5
all packets sent
Total Transfer Time: 2.563930 seconds
usc542ee@badb:/tmp$
```