

# Socket Programming Homework 1

21800603 이현승

22000546 이예은

## 1. Protocol 구조

### 1) Header

Header	Data type	Description
type	int	메세지 타입의 종류
seq_no	int	패킷의 sequence number
data	char []	전송할 파일 데이터
data_len	int	데이터의 길이
checksum	int	checksum

### 2) 메세지 타입

header의 type 변수가 메세지 타입을 나타낸다. type은 filename(0), data(1), ACK(2), transfer complete(3)의 메세지 타입을 가진다. filename(0)은 전송되는 데이터가 파일명이라는 것을 알리고, data(1)는 전송되는 데이터가 파일 데이터를 의미한다는 것을 알린다. ACK(2)는 패킷을 정상적으로 수신했다는 표시를 위한 메세지 타입이고, transfer complete(3)는 모든 데이터가 전송이 완료되었다는 의미를 전달한다. client에서 파일의 크기를 읽어 총 몇 개의 패킷을 보낼지 계산하므로 next sequence number를 보면 몇 개의 패킷이 남았는지 알 수 있다. 마지막 데이터 패킷까지 전송이 된 경우에는 파일 전송이 다 되었음을 알리는 type=3의 패킷을 보낸다.

### 3) Timer

타이머는 High Resolution Timer를 이용하였다. interval field는 사용하지 않고 value만 사용해서 one-shot timer를 이용하였다. 데이터를 보내는 과정에서 send\_base와 nextseqnum이 같아지면 타이머를 시작하여 받은 ACK에 의해 send\_base가 nextseqnum만큼 올라와 같아지면 타이머를 종료하였다. 받은 ACK이 send\_base와 같아지 않을 때는 타이머를 다시 세팅하여 시간을 초기화하였다. 10초가 지나 time-out이 발생하면 time-out handler가 작동하는데, time-out handler에서는 send\_base부터 nextseqnum-1까지 지금까지 ACK을 받지 못한 보낸 패킷들을 전부 재전송한다.

## 2. 프로그램 동작 결과

### 1) 프로그램 동작 상태

(1) client 가 우선 파일명을 패킷 데이터에 담아 파일명을 보내는 것을 의미하는 type=0 을 선언하여 server 에 보낸다.

(2) server 는 파일명을 수신하고 이를 이용하여 새로운 파일을 형성한다.

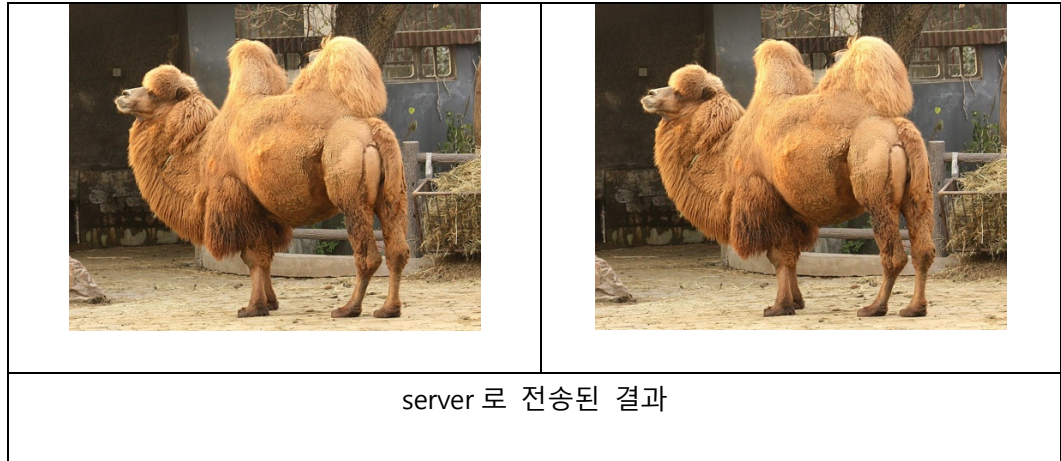
- (3) 새로운 파일이 형성되면 그 이후 client 는 패킷 type 을 data 를 의미하는 2 로 선언하고 버퍼 사이즈에 맞춰서 데이터를 서버에 보낸다.
- (4) 서버는 패킷을 받는 대로 client 에 정상적으로 파일을 수신하였다는 것을 type = 2, sequence number+1 을 보내 정상적으로 수신하였음을 알려준다.
- (5) 설정된 시간 내 ACK 을 받지 못한 경우, client 는 마지막으로 수신한 ACK sequence number 다음 패킷(expectedseqnum)을 다시 발송한다.
- (6) client 가 모든 데이터를 다 읽은 경우, client 는 모든 데이터가 다 전송되었다는 것을 server 에 알리기 위해 transfer complete 를 의미하는 type = 3 을 선언하여 만든 패킷을 server 에 보낸다.
- (7) Packet type 이 3 인 패킷을 받은 서버는 모든 데이터를 수신하였음을 인지하고 expectedseqnum 을 0 으로 초기화한다

## 2) 프로그램 결과

- BactCamel.JPG (binary file, 176KB)



client 에 있는 BactCamel.JPG



```
[s22000546@localhost ~]$ diff hw1/server/BactCamel.JPG ./hw1/files/BactCamel_client.JPG
[s22000546@localhost ~]$
```

육안으로 봤을 때도 동일한 사진이고, diff 명령어를 통해 확인했을 때도 차이가 없다.

- 100kb.txt (txt file, 102KB)

```
[s22000546@localhost ~]$ diff hw1/server/100kb.txt ./hw1/files/100kb.txt
```

<diff 실행 결과>

```
[s22000546@localhost client]$ wc -c ./100kb.txt
102399 ./100kb.txt
```

<client에서 실행한 결과>

```
[s22000546@localhost ~]$ wc -c hw1/server/100kb.txt
102399 hw1/server/100kb.txt
```

<server에서 실행한 결과>

word의 byte수가 같으며, diff 명령어를 이용해 비교했을 때도 차이가 없다는 결과를 얻을 수 있다.