

Telecommunication II

Assignment 1 : Report

The assignment was to design a gateway that would receive and send back packets between a client and a server. The client is to send a packet to the gateway. And the gateway needs to send that packet to the server. The server sends back an acknowledgement to the client, passing by the gateway.

In my implementation, the gateway uses the port number 12345. While the client uses 50000 and the server 50001 as default.

The client packetizes the string he wants to send to the server with a sequence number. That packet is then encapsulated into another packet and the whole is sent to the gateway.

Packets consist of a payload and a header. The payload has the information to be transmitted while the header contains the information about the transmission.

My header are 10 bytes long and contain information about the source and the destination port number,

In my headers, the first two bytes, 0 and 1, are for the sender port number.

Byte 5 and 6 are for the destination port number.

Byte 7 tells the gateway whether or not, the packet received contains another packet inside it. In other words, it tells the gateway if it can just forward the received packet its destination if its 0. Or else, it needs to be unpacked and take the new packet and send that one.

Byte 8 is a flag telling if a packet in an NACK or not.

Byte 9 contains the sequence number, or the acknowledgement number.

The client first sets up the packet for the server, with the string to be sent and the corresponding header, and makes a new header for the gateway and takes the original packet as a payload for the new one.

```
> Frame 89: 110 bytes on wire (880 bits), 57 bytes captured (456 bits) on interface 0
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▼ User Datagram Protocol, Src Port: 50000, Dst Port: 12345
    Source Port: 50000
    Destination Port: 12345
    Length: 33
    Checksum: 0xb8b5 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
▼ Data (25 bytes)
    Data: c3500000003039010000c350000000c35100000268656c6c...
    [Length: 25]
```

```
0000  02 00 00 00 45 00 00 35 0a 53 00 00 80 11 00 00  ....E..5 .S.....
0010  7f 00 00 01 7f 00 00 01 c3 50 30 39 00 21 b8 b5  ....P09.!..
0020  c3 50 00 00 00 30 39 01 00 00 c3 50 00 00 00 c3  .P...09. ...P....
0030  51 00 00 02 68 65 6c 6c 6f                Q...hell o
```

As we can see in the screenshot, the data is 25 bytes long, the first 10 are the header that the gateway reads. It reads byte 7, or 0x27, and understand that it needs to throw this header to make the new packet to be sent to the server. Then, it reads the last 15 bytes, which are the original packet, destined for the server. And takes the destination port from bytes 0x2f and 0x30, which are c3 and 51 respectively.

The gateway doesn't need to know the port of the server in advances, it just learn from the original packet. With this design, the gateway could be connected to different ports of the same server. As long as it is the local host. But it makes for a bigger packet.

The server receives the original packet and check the sequence number, the bytes at location 0x33. He compares it to the sequence number he expects. If they are the same, the server sends back an acknowledgement for the next packet. Otherwise he sends back a negative acknowledgement, although this feature isn't implemented in my program.

The server creates a new packet for the acknowledgement, with its port as the source and the client port for the destination although it is first sent to the gateway. The gateway reads 0 for byte 7, 0x27 so it just forwards the packet to the client. And the client reads the packet as an acknowledgement for the next packet and so updates his sequence number.

```
> Frame 877: 90 bytes on wire (720 bits), 47 bytes captured (376 bits) on interface 0
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
  User Datagram Protocol, Src Port: 12345, Dst Port: 50000
    Source Port: 12345
    Destination Port: 50000
    Length: 23
    Checksum: 0x3bb9 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
  Data (15 bytes)
    Data: c351000000c35000000241434b2032
    [Length: 15]
```

0000	02 00 00 00 45 00 00 2b	6e bb 00 00 80 11 00 00E..+ n.....
0010	7f 00 00 01 7f 00 00 01	30 39 c3 50 00 17 3b b9 09.P..j.
0020	c3 51 00 00 00 c3 50 00	00 02 41 43 4b 20 32	.Q....P. ..ACK 2

In my design, the gateway only needs to know the IP address, which for this assignment is just that of the localhost, so my gateway doesn't need to know any information. It can figure out by itself, who is sending a packet and where to send it under the right form.