**Computer Network**

#Final Project

201111286 노준호

1. **프로그램 설명**

Design and write an IP datagram transmission program including Dijkstra algorithm for calculating the shortest path

다익스트라를 이용하여 가장짧은길을 계산해서 IP datagram을 두 호스트사이에 전송.

**- The goals of this program are as follows.**  
      1) This program generates ten routers, one client, and three servers for transmitting an IP datagram.  
      2) Each router includes a routing table consisting of router information for object Router, and metric information for adjacent routers, clients, and servers.  
      3) This program calculates the shortest path for transmitting an IP datagram by using Dijkstra algorithm.  
      4) A client transmits an IP datagram to a server by utilizing the shortest path calculated.

**2. 프로그램 결과 설명**

-----------------------------------------------------------------------------  
  
Source IP address for server1 : 10.1.0.12  
\*\*\*\*\*\*\*\*\* Server1- IP: 10.1.0.12, Port: 190\*\*\*\*\*\*\*\*\*  
  
Source IP address for server2 : 10.3.0.9  
\*\*\*\*\*\*\*\*\* Server2- IP: 10.3.0.9, Port: 190\*\*\*\*\*\*\*\*\*  
  
Source IP address for server3 : 10.2.0.3  
\*\*\*\*\*\*\*\*\* Server3- IP: 10.2.0.3, Port: 190\*\*\*\*\*\*\*\*\*  
  
Destination IP Address : 10.3.0.9  
Destination Port : 190  
  
Shortest path for sending IP datagram : [10.0.0.20, 10.0.0.1, 10.1.0.1, 10.1.0.2, 10.1.1.2, 10.3.0.1, 10.3.0.9]  
Distance to destination : 23.0  
  
--- Pass through Router A---  
--- Pass through Router D---  
--- Pass through Router E---  
--- Pass through Router F---  
--- Pass through Router I----

서버 아이피를 지정해준다. Port번호는 모두 190으로 같다. 연결할 서버 IP address와 port를 입력받는다. Destination IP address와 Destination port는 서버의 Source IP ,Source Port가 된다.

다익스트라 알고리즘을 이용하여 가장 짧은 path를 찾고 출력해준다.

----------Handshake in Server----------  
-------------IP Datagram-------------  
Version: 4  
Header Length: 20  
Service Type: 0  
Total Length: 600  
Identification: 0  
Flags: 0  
Fragmentation Offset: 0  
Time to Live: 500  
Protocol: 6  
Header Checksum: 0  
Source IP: 10.0.0.20  
Destination IP: 10.3.0.9  
-------------TCP Segment-------------  
Source Port: 250  
Destination Port: 190  
Sequence Number: 100  
Acknowledgement Number: 0  
Header Length: 8  
Reserved: 0  
TCP Flag  
     - URG: 0  
     - ACK: 0  
     - PSH: 0  
     - RST: 0  
     - SYN: 1  
     - FIN: 0  
Received Window: 3  
Checksum: 3  
Urgent data: 5  
Payload: null

클라이언트에서 다익스트라 알고리즘을 이용하여 서버로 가는 가장 짧은 path를 찾고 세방향 핸드쉐이크를 한다. 두 호스트(클라이언트,서버) 사이에 3개의 데이터그램이 보내지는 세방향 핸드쉐이크가 일어난다. 클라이언트가 자신의 세그먼트에 삽입하는 ACK는 클라이언트가 서버로부터 기대하는 다음바이트의 sequence number 이다. 앞으로 데이터를 보내기 위해 핸드쉐이크 과정이 일어나는 것이다.

Shortest path for sending IP datagram : [10.3.0.9, 10.3.0.1, 10.1.1.2, 10.1.0.2, 10.1.0.1, 10.0.0.1, 10.0.0.20]  
Distance to destination : 23.0  
  
--- Pass through Router I---  
--- Pass through Router F---  
--- Pass through Router E---  
--- Pass through Router D---  
--- Pass through Router A---  
  
  
----------Handshake in Client----------  
  
-------------IP Datagram-------------  
Version: 4  
Header Length: 20  
Service Type: 0  
Total Length: 600  
Identification: 0  
Flags: 0  
Fragmentation Offset: 0  
Time to Live: 500  
Protocol: 6  
Header Checksum: 0  
Source IP: 10.3.0.9  
Destination IP: 10.0.0.20  
-------------TCP Segment-------------  
Source Port: 190  
Destination Port: 250  
Sequence Number: 201  
Acknowledgement Number: 101  
Header Length: 8  
Reserved: 0  
TCP Flag  
     - URG: 0  
     - ACK: 1  
     - PSH: 0  
     - RST: 0  
     - SYN: 1  
     - FIN: 0  
Received Window: 3  
Checksum: 3  
Urgent data: 5  
Payload: null

Hand shake 과정에서 실제 데이터는 전송되지 않으므로 Payload는 null로 표시된다.

---------------------Input Data for Transmission------------------  
>>InputData : konkuk  
  
  
Shortest path for sending IP datagram : [10.0.0.20, 10.0.0.1, 10.1.0.1, 10.1.0.2, 10.1.1.2, 10.3.0.1, 10.3.0.9]  
Distance to destination : 23.0  
  
--- Pass through Router A---  
--- Pass through Router D---  
--- Pass through Router E---  
--- Pass through Router F---  
--- Pass through Router I---  
  
-------------IP Datagram-------------  
Version: 4  
Header Length: 20  
Service Type: 0  
Total Length: 600  
Identification: 0  
Flags: 0  
Fragmentation Offset: 0  
Time to Live: 500  
Protocol: 6  
Header Checksum: 0  
Source IP: 10.0.0.20  
Destination IP: 10.3.0.9  
-------------TCP Segment-------------  
Source Port: 250  
Destination Port: 190  
Sequence Number: 102  
Acknowledgement Number: 203  
Header Length: 8  
Reserved: 0  
TCP Flag  
     - URG: 0  
     - ACK: 0  
     - PSH: 0  
     - RST: 0  
     - SYN: 0  
     - FIN: 0  
Received Window: 3  
Checksum: 3  
Urgent data: 5  
Payload: k

문자열을 입력받으면 핸드쉐이크로 설정한 서버에게 데이터를 전달한다. 이때도 다익스트라 알고리즘을 이용하여 가장 짧은 path 로 데이터를 전송한다. konkuk이라는 문자열을 입력받았을 때 한글자씩 전송하므로 Payload에 k가 먼저 뜬다. 다음엔 o,n,k,u,k순으로 데이터전송이 이루어진다. 코드상에서는 문자열을 한번에 받아서 출력하는식으로 하게된다. 전송될때마다 IP 데이터그램 헤더정보와 TCPSegment 헤더정보가 출력된다. 그리고 seq와 ack값을 1씩 증가시킨다.

>>You would like to continue the program, please enter y.  
X  
  
Shortest path for sending IP datagram : [10.0.0.20, 10.0.0.1, 10.1.0.1, 10.1.0.2, 10.1.1.2, 10.3.0.1, 10.3.0.9]  
Distance to destination : 23.0  
  
--- Pass through Router A---  
--- Pass through Router D---  
--- Pass through Router E---  
--- Pass through Router F---  
--- Pass through Router I---  
  
----- Close socket in server -----  
  
-------------IP Datagram-------------  
Version: 4  
Header Length: 20  
Service Type: 0  
Total Length: 600  
Identification: 0  
Flags: 0  
Fragmentation Offset: 0  
Time to Live: 500  
Protocol: 6  
Header Checksum: 0  
Source IP: 10.0.0.20  
Destination IP: 10.3.0.9  
-------------TCP Segment-------------  
SourcePort : 250  
DestinationPort : 190  
SequenceNumber : 108  
AcknowledgementNumber : 209  
HeaderLength : 8  
Reserverd : 1  
TCPFlags   
     URG : 0  
     ACK : 0  
     PSH : 0  
     RST : 0  
     SYN : 0  
     FIN : 1  
ReceiveWindow : 3  
UrgentData : 5

y이외의 문자를 입력받으면 데이터전송을 종료하고 소켓을 닫고 연결상태를 출력하는 과정을 보여준다.

**3. 참고자료 및 사이트**

https://javabelazy.blogspot.kr/2012/02/dijkstra-algorithm-implementation-in.html