

CS 536 Spring 2024

Lab 6 (Bonus): Application Layer Routing

Alireza Lotfi

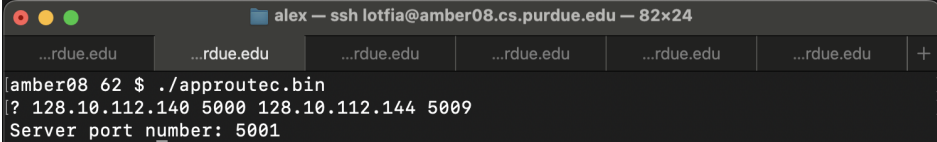
Contents

1	Results	2
1.1	Using 1 Router	2
1.2	Using 3 Routers	3
1.3	Using 5 Routers	4
1.4	RTT values	5

1 Results

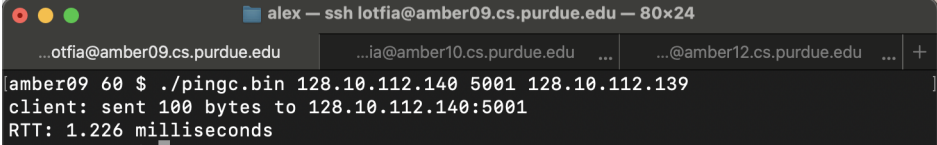
1.1 Using 1 Router

In this scenario, we employ a process akin to tunneling, where data is routed from the client to the server through a single intermediary router. This process can be visualized as follows:



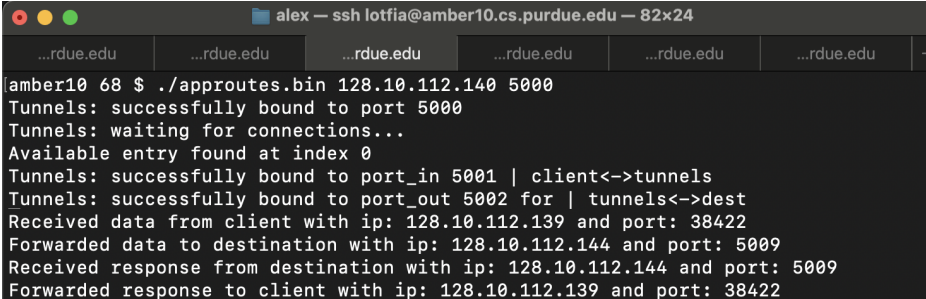
```
alex — ssh lotfia@amber08.cs.purdue.edu — 82x24
...rdue.edu ...rdue.edu ...rdue.edu ...rdue.edu ...rdue.edu ...rdue.edu +
amber08 62 $ ./approutec.bin
? 128.10.112.140 5000 128.10.112.144 5009
Server port number: 5001
```

(a) Configuration



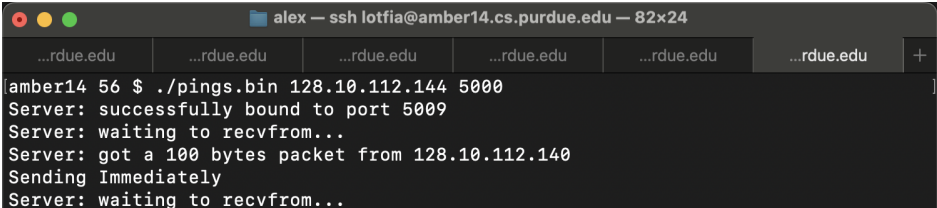
```
alex — ssh lotfia@amber09.cs.purdue.edu — 80x24
...otfia@amber09.cs.purdue.edu ...ia@amber10.cs.purdue.edu ... @amber12.cs.purdue.edu ... +
amber09 60 $ ./pingc.bin 128.10.112.140 5001 128.10.112.139
client: sent 100 bytes to 128.10.112.140:5001
RTT: 1.226 milliseconds
```

(b) Pingc



```
alex — ssh lotfia@amber10.cs.purdue.edu — 82x24
...rdue.edu ...rdue.edu ...rdue.edu ...rdue.edu ...rdue.edu ...rdue.edu +
amber10 68 $ ./approutes.bin 128.10.112.140 5000
Tunnels: successfully bound to port 5000
Tunnels: waiting for connections...
Available entry found at index 0
Tunnels: successfully bound to port_in 5001 | client<->tunnels
Tunnels: successfully bound to port_out 5002 for | tunnels<->dest
Received data from client with ip: 128.10.112.139 and port: 38422
Forwarded data to destination with ip: 128.10.112.144 and port: 5009
Received response from destination with ip: 128.10.112.144 and port: 5009
Forwarded response to client with ip: 128.10.112.139 and port: 38422
```

(c) Router

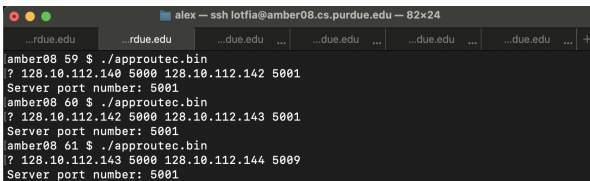


```
alex — ssh lotfia@amber14.cs.purdue.edu — 82x24
...rdue.edu ...rdue.edu ...rdue.edu ...rdue.edu ...rdue.edu ...rdue.edu +
amber14 56 $ ./pings.bin 128.10.112.144 5000
Server: successfully bound to port 5009
Server: waiting to recvfrom...
Server: got a 100 bytes packet from 128.10.112.140
Sending Immediately
Server: waiting to recvfrom...
```

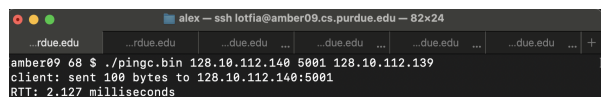
(d) Pings

Figure 1: Process

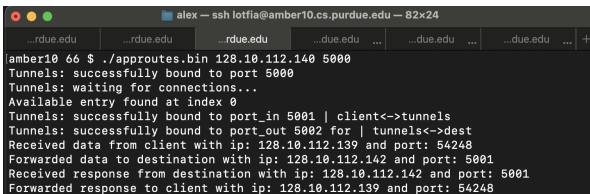
In this segment, we introduce a network configuration utilizing three routers instead of one. Consequently, the data packet originating from the client is relayed through three intermediary routers before reaching its ultimate destination. Upon reaching the destination, any response follows the same path in reverse, returning to the originating device. The process is illustrated below:



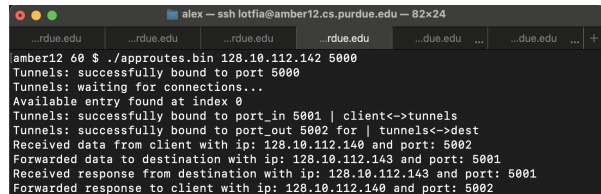
(a) Configuration



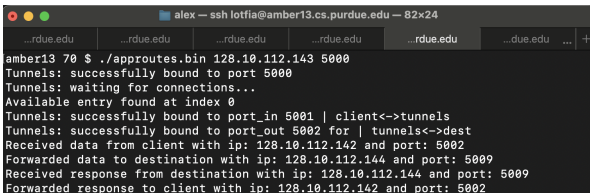
(b) Pingc



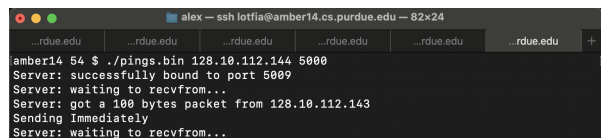
(c) Router 1



(d) Router 2



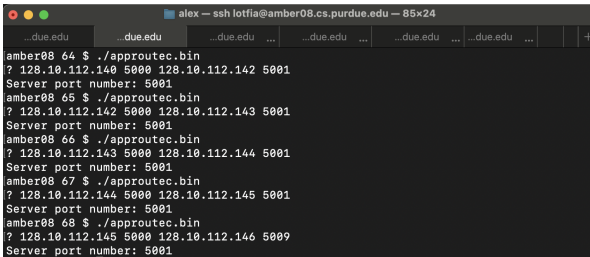
(e) Router 3



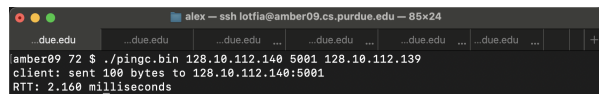
(f) Pings

Figure 2: Process

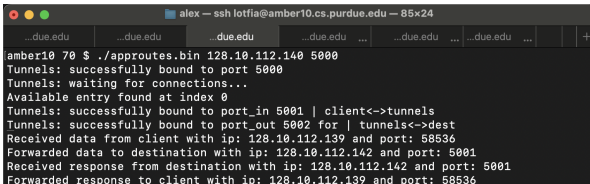
This section, shows the same process with the difference of using five routers instead of three.



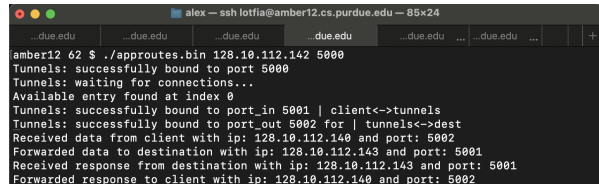
(a) Configuration



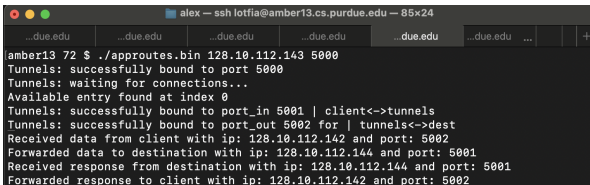
(b) Pingc



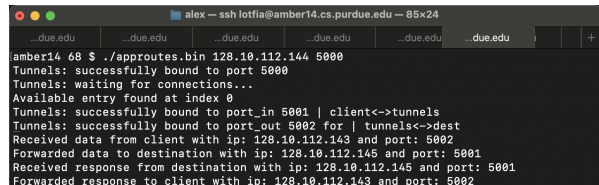
(c) Router 1



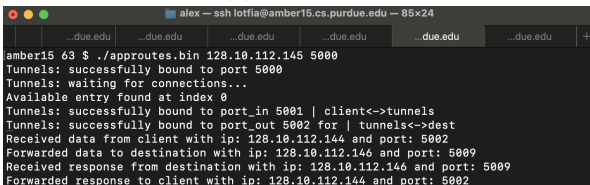
(d) Router 2



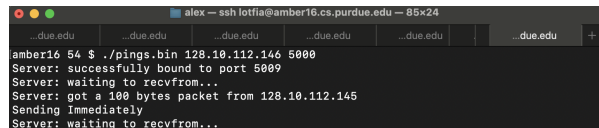
(e) Router 3



(f) Router 4



(g) Router 5



(h) Pings

Figure 3: Process

1.4 RTT values

Table 1: Comparison of RTT values

Number of Routers	RTT (msec)
1	1.226
3	2.127
5	2.160

The provided table compares the Round-Trip Time (RTT) values for different numbers of routers in a network configuration. With just one router, the RTT is recorded at 1.226 milliseconds, indicating a relatively low latency for data transmission. However, as the number of routers increases, there is a notable rise in RTT values. With three routers, the RTT increases to 2.127 milliseconds, showcasing a moderate increase in latency compared to the single-router setup. Interestingly, adding two more routers to the network configuration results in a marginal further increase in RTT to 2.160 milliseconds. This comparison underscores the impact of network topology on data transmission latency, with each additional router introducing a slight but discernible delay in the round-trip communication process.