Computer Networks　　Jacobs University Bremen　　March 28th, 2020
Arsenij Percov

Homework 3

Problem 3.1

Solution:

1)

h1 to r1 ping:

```
h1 ping -6 2001:638:709:a::f
statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9206ms
rtt min/avg/max/mdev = 0.044/0.082/0.093/0.016 ms
```

h1 to r2 ping:

```
h1 ping -6 2001:638:709:f::2  statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9195ms
rtt min/avg/max/mdev = 0.101/0.112/0.165/0.022 ms
```

h1 to h2 ping:

```
h1 ping -6 -c 10 2001:638:709:b::1
statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9199ms
rtt min/avg/max/mdev = 0.113/0.133/0.159/0.012 ms
```

r1 to h1 ping:

```
r1 ping -6 2001:638:709:a::1 statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 10222ms
rtt min/avg/max/mdev = 0.040/0.084/0.103/0.019 ms
```

r1 to r2 ping:

```
r1 ping -6 -c 10 -i 0
statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.025/0.035/0.100/0.022 ms, ipg/ewma 0.052/0.049 ms
```

r1 to h2 ping:

```
r1 ping -6 -c 10 -i 0 2001:638:709:b::1
statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.035/0.047/0.128/0.028 ms, ipg/ewma 0.063/0.065 ms
```

r2 to h1 ping:

```
r2 ping -6 -c 10 -i 0 2001:638:709:a::1
statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.035/0.048/0.130/0.029 ms, ipg/ewma 0.071/0.066 ms
```

r2 to r1 ping:

```
 r2 ping -6 -c 10 -i 0 2001:638:709:f::1
statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.026/0.037/0.100/0.023 ms, ipg/ewma 0.055/0.051 ms
```

r2 to h2 ping:

```
r2 ping -6 -c 10 -i 0 2001:638:709:b::1
statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.025/0.035/0.099/0.022 ms, ipg/ewma 0.054/0.049 ms
```

h2 to r2 ping:

```
h2 ping -6 -c 10 -i 0 2001:638:709:b::f
statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.024/0.034/0.102/0.024 ms, ipg/ewma 0.052/0.050 ms
```

h2 to r1 ping:

```
h2 ping -6 -c 10 -i 0 2001:638:709:f::1
statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.036/0.046/0.122/0.026 ms, ipg/ewma 0.065/0.063 ms
```

h2 to h1 ping:

```
 h2 ping -6 -c 10 -i 0 2001:638:709:a::1
statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.046/0.060/0.163/0.034 ms, ipg/ewma 0.078/0.083 ms
```

b) h1 to h2:

```
h1 traceroute 2001:638:709:b::1
traceroute to 2001:638:709:b::1 (2001:638:709:b::1), 30 hops max, 80 byte packets
 1  2001:638:709:a::f (2001:638:709:a::f)  0.615 ms  0.541 ms  0.506 ms
 2  * * *
 3  2001:638:709:b::1 (2001:638:709:b::1)  0.391 ms  0.326 ms  0.283 ms
```

h1 to h2:

```
h1 traceroute -6 2001:638:709:b::1
traceroute to 2001:638:709:b::1 (2001:638:709:b::1), 30 hops max, 80 byte packets
 1  2001:638:709:a::f (2001:638:709:a::f)  1.429 ms  1.236 ms  1.182 ms
 2  2001:638:709:f::2 (2001:638:709:f::2)  1.136 ms  0.983 ms  0.922 ms
 3  2001:638:709:b::1 (2001:638:709:b::1)  0.866 ms  0.683 ms  0.607 ms
```

h2 to h1:

```
 h2 traceroute -6 2001:638:709:a::1
traceroute to 2001:638:709:a::1 (2001:638:709:a::1), 30 hops max, 80 byte packets
 1  2001:638:709:b::e (2001:638:709:b::e)  1.263 ms  1.154 ms  1.102 ms
 2  2001:638:709:e::1 (2001:638:709:e::1)  1.055 ms  0.893 ms  0.830 ms
 3  2001:638:709:a::1 (2001:638:709:a::1)  0.774 ms  0.678 ms  0.606 ms
```

c)
Using the following commands to configure the mtu
r1 ip link set dev r1-eth1 mtu 1400
r2 ip link set dev r2-eth0 mtu 1400

Using wireshark we could notice that the package got fragmented. We sent a package of size 2000:
First part was sent via the following IPv6 protocol, and had fragment header for IPv6:

```
 Internet Protocol Version 6, Src: 2001:638:709:a::1, Dst: 2001:638:709:b::1
    0110 .... = Version: 6
    .... 0000 0000 .... .... .... .... .... = Traffic Class: 0x00 (DSCP: CS0, ECN: No
    .... .... .... 1111 1110 1001 0110 1111 = Flow Label: 0xfe96f
    Payload Length: 1360
    Next Header: Fragment Header for IPv6 (44)
    Hop Limit: 64
    Source: 2001:638:709:a::1
    Destination: 2001:638:709:b::1
    Fragment Header for IPv6
        Next header: ICMPv6 (58)
        Reserved octet: 0x00
        0000 0000 0000 0... = Offset: 0 (0 bytes)
        .... .... .... .00. = Reserved bits: 0
        .... .... .... ...1 = More Fragments: Yes
        Identification: 0xde8c4c53
    Reassembled IPv6 in frame: 18
```

The second package was ICMPv6 encapsulating IPv6 with the end of the data: (other header and fragment)

```
Internet Protocol Version 6, Src: 2001:638:709:a::1, Dst: 2001:638:709:b::1
 0110 .... = Version: 6
 .... 0000 0000 .... .... .... .... .... = Traffic Class: 0x00 (DSCP: CS0, ECN: No
 .... .... .... 1111 1110 1001 0110 1111 = Flow Label: 0xfe96f
 Payload Length: 664
 Next Header: Fragment Header for IPv6 (44)
 Hop Limit: 64
 Source: 2001:638:709:a::1
 Destination: 2001:638:709:b::1
 Fragment Header for IPv6
     Next header: ICMPv6 (58)
     Reserved octet: 0x00
     0000 0101 0100 1... = Offset: 169 (1352 bytes)
     .... .... .... .00. = Reserved bits: 0
     .... .... .... ...0 = More Fragments: No
     Identification: 0xde8c4c53
 [2 IPv6 Fragments (2008 bytes): #17(1352), #18(656)]
     [Frame: 17, payload: 0-1351 (1352 bytes)]
     [Frame: 18, payload: 1352-2007 (656 bytes)]
     [Fragment count: 2]
     [Reassembled IPv6 length: 2008]
     [Reassembled IPv6 data: 80004848401400058f04815e0000000074a5040000000000...]
```

Important to notice that routers do not fragment the data, as it was done in the ipv4. In this case nodes are expected to make the discovery of the path MTU, and upper layer protocol is expected to limit the payload size. If it fails to do so (ex. our scenario), then Fragment extension header is used, and the packets are leaving the h1 host in already fragmented way.[source:]

d)
We cannot use same strategy from IPv4, since we cannot send Don't Fragment flag, since routers do no fragment at all. In IPv6, works by initially assuming the path MTU as the one from the link layer interface of the origin host. If the package is bigger than MTU of some device on its path, that device will send back ICMPv6 Packet Too Big message with precise MTU, this will be repeated until MTU is small enough to traverse whole path.[source:]

e)
No, the data is cached in the routing tables entries of the host. It can cache it for every active destination. [source: https://tools.ietf.org/html/rfc1191]