1. IP address associated with host-only network: `192.168.56.101` (Figure 1, A)
   The IPv4 address for host-only networks on VirtualBox is set as `192.168.56.1` with a network mask of `255.255.255.0`. This suggests that the IP address of the host-only network must be of the form `192.168.56.x`, which is seen for eth0 below.

```
mininet@mininet-vm:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:b4:52:ec
          inet addr:192.168.56.101  Bcast:192.168.56.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1180 (1.1 KB)  TX bytes:684 (684.0 B)

eth1      Link encap:Ethernet  HWaddr 08:00:27:44:fe:d6
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:282 errors:0 dropped:0 overruns:0 frame:0
          TX packets:284 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:27458 (27.4 KB)  TX bytes:25470 (25.4 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

A

Figure 1

2. The `links` command displays the links between the hosts and switches. There are three connections established: h1-s3, s3-s4 and s4-h2. (Figure 2)

```
mininet> links
h1-eth0<->s3-eth1 (OK OK)
s3-eth2<->s4-eth1 (OK OK)
s4-eth2<->h2-eth0 (OK OK)
```

Figure 2

The `ping` command sends five packets from h1 to h2, then stops. It checks if h1 and h2 are connected and can send and received packs from each other. Finally, it reports the approximate minimum, maximum, average and standard deviation of RTT in milliseconds. (Figure 3)

```
mininet> h1 ping -c 5 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=1225 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=630 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=626 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=625 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=617 ms

--- 10.0.0.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4038ms
rtt min/avg/max/mdev = 617.349/744.885/1225.045/240.118 ms, pipe 2
```

Figure 3

The `iperf` command measures the bandwidth between h1 and h2 to report MSS and TCP window sizes.

```
mininet> iperf h1 h2
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['5.71 Mbits/sec', '12.0 Mbits/sec']
```

Figure 4

3. The links command displays the same information as earlier, since no changes were made to the links between switches and hosts.

```
mininet> links
h1-eth0<->s3-eth1 (OK OK)
s3-eth2<->s4-eth1 (OK OK)
s4-eth2<->h2-eth0 (OK OK)
```

Figure 5

The ping command displays significantly lower transmission and RTT times since a flow entry covering ICMP ping traffic was previously installed in the switch, so no control traffic was generated, and the packets immediately pass through the switch. Since the delay is lower (from 100ms to 5ms, the RTT is also lower by the same factor of 20.

```
mininet> h1 ping -c 5 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=80.5 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=50.5 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=26.5 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=27.2 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=62.1 ms

--- 10.0.0.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4014ms
rtt min/avg/max/mdev = 26.570/49.389/80.515/20.705 ms
```

Figure 6

The iperf command also generated lower values for MSS and TCP window sizes. Since the delay in the h1-h2 link is decreased, the MSS and TCP window sizes also decreased. Packets are coming in faster and therefore, congestion control is activated sooner and window sizes are decreased.

Figure 7