Computer Networks
Jacobs University Bremen
Dr. Jürgen Schönwälder

Course: CO20-320301
Date: 2019-02-15
Due: 2019-02-22

CN 2019 Problem Sheet #1

Problem 1.1: *measurements with ping and traceroute* (1+1 = 2 points)

Do this experiment over IPv4. Make sure you force the usage of IPv4 in case you have both IPv4 and IPv6 connectivity.

a) The `ping` utility can be used to measure the round-trip time between the host running `ping` and remote hosts. Use ping to measure the round-trip time to

   - `amazon.com`,
   - `www.amazon.com`,
   - `www.jacobs-university.de`, and
   - `moodle.jacobs-university.de`

   from a machine connected to the campus network. Calculate the minimum round-trip time observed over 10 samples. What do you observe? (Note: The `fping` implementation can ping multiple hosts concurrently.) Properly record when you did the measurement and which tools were used (including their version).

b) The `traceroute` utility can trace the route packets take on the Internet. The `mtr` utility is a more modern implementation that can trace a route in a faster more aggressive way. Trace the routes to the destinations

   - `amazon.com`,
   - `www.amazon.com`,
   - `www.jacobs-university.de`, and
   - `moodle.jacobs-university.de`

   from a machine connected to the campus network. Enable the command line option of your traceroute utility to lookup the autonomous system (AS) number of the IP addresses (`-z` for `mtr`). For each destination, write down the sequence of autonomous systems crossed (the so-called AS-path) and the number of hops in each autonomous system. Did you observe something interesting? Properly record when you did the measurement and which tools were used (including their version).

Problem 1.2: *ripe stats* (2 points)

The RIPE Network Coordination Centre, the registrar responsible for Europe, is providing services that help troubleshooting Internet routing problems. The `https://stat.ripe.net/` web site provides an easy to use interface to their information resources.

a) Lookup the name of the organizations owning the AS numbers you observed in the previous exercise. Which of the five registries (ARIN, RIPE, APNIC, LACNIC, AFRINIC) did assign the AS numbers?

b) Lookup the IPv6 prefix `2001:638:709::/48`. Who is using this prefix? Is the prefix globally announced? If not, which prefix is announced?

Problem 1.3: *mininet - point-to-point topology* (1+1 = 2 points)

For this problem and subsequent problems you need to install Mininet, which enables you to emulate a complete network on a single computer running Linux. Once you have Mininet installed, run the `point.py` python script (`sudo python point.py`) to create a simple point-to-point topology.

```
#!/usr/bin/env python
""" point.py:

    This mininet script creates a simple point-to-point topology:

            10 mbps
      h1 ------------- h2
"""

from mininet.cli import CLI
from mininet.net import Mininet
from mininet.link import TCLink
from mininet.topo import Topo
from mininet.log import setLogLevel

class PointToPoint(Topo):
    def __init__(self, **opts):
        Topo.__init__(self, **opts)
        h1 = self.addHost('h1')
        h2 = self.addHost('h2')
        self.addLink(h1, h2, bw=10)

if __name__ == '__main__':
    setLogLevel('info')

    topo = PointToPoint()
    net = Mininet(topo=topo, link=TCLink,
                  autoSetMacs=True, autoStaticArp=True)

    net.start()
    CLI(net)
    net.stop()
```

The iperf utility for active data rate measurements between two hosts. On one host, you run an iperf server (`iperf -s`) and on the other host you run an iperf client (`iperf -c <host>`). Additional options control how long `iperf` is measuring the data rate (`-t`) and how frequently results are reported (`-i`).

a) Within Mininet, run an iperf server on h2 (`h2 iperf -s &`) and measure the data rate of the link by running an iperf client on h1 (`h1 iperf -c h2 -i 10 -t 60`). Does the measured data rate match your expectations?

b) Ping h2 from h1 while iperf measures the data rate and while there is no iperf measurement in the background. Is there a difference in the observed round-trip times? What is causing the delays (transmission delay or propagation delay or queuing delay)?

Problem 1.4: *mininet - star topology* (1+1 = 2 points)

Run the `star.py` python script (`sudo python star.py`) to create a simple star topology.

```
#!/usr/bin/env python
""" star.py:

    This mininet script creates a simple star topology with a switch
    in the center:

      h1                h3
         \             /
       10 \         /10 mbps
            \   /
              s1
            /   \
       10 /       \10 mbps
         /           \
      h2              h4
"""

from mininet.cli import CLI
from mininet.net import Mininet
from mininet.link import TCLink
from mininet.topo import Topo
from mininet.log import setLogLevel

class Star(Topo):
    def __init__(self, **opts):
        Topo.__init__(self, **opts)
        h1 = self.addHost('h1')
        h2 = self.addHost('h2')
        h3 = self.addHost('h3')
        h4 = self.addHost('h4')
        s1 = self.addSwitch('s1')
        self.addLink(h1, s1, bw=10)
        self.addLink(h2, s1, bw=10)
        self.addLink(h3, s1, bw=10)
        self.addLink(h4, s1, bw=10)

if __name__ == '__main__':
    setLogLevel('info')

    topo = Star()
    net = Mininet(topo=topo, link=TCLink,
                  autoSetMacs=True, autoStaticArp=True)

    net.start()
    CLI(net)
    net.stop()
```

a) Run an `iperf` measurement between h1 and h2. While iperf is running, run ping between h3 and h4 and observe the round-trip times. Does the `iperf` measurement impact the observed round-trip times?

b) Run an `iperf` measurement between h1 and h2 and another between h3 and h4. Do the two concurrent measurements impact each other?

Problem 1.5: *mininet - network topology* (1+1 = 2 points)

Run the `network.py` python script (`sudo python network.py`) to create a network topology.

```
#!/usr/bin/env python
""" network.py:

    This mininet script creates a switched multi-hop topology:

      h1               h3               h5               h7
       |                |                |                |
      10|              10|              10|              10|
       |     10 mbps    |     15 mbs     |     20 mbps    |
       s1 ------------- s2 ------------- s3 ------------- s4
       |                |     5% loss    |                |
      10|              10|              10|              10|
       |                |                |                |
       h2               h4               h6               h8
"""

from mininet.cli import CLI
from mininet.net import Mininet
from mininet.link import TCLink
from mininet.topo import Topo
from mininet.log import setLogLevel

class Network(Topo):
    def __init__(self, **opts):
        Topo.__init__(self, **opts)
        h1 = self.addHost('h1')
        h2 = self.addHost('h2')
        h3 = self.addHost('h3')
        h4 = self.addHost('h4')
        h5 = self.addHost('h5')
        h6 = self.addHost('h6')
        h7 = self.addHost('h7')
        h8 = self.addHost('h8')
        s1 = self.addSwitch('s1')
        s2 = self.addSwitch('s2')
        s3 = self.addSwitch('s3')
        s4 = self.addSwitch('s4')
        self.addLink(h1, s1, bw=10)
        self.addLink(h2, s1, bw=10)
        self.addLink(h3, s2, bw=10)
        self.addLink(h4, s2, bw=10)
        self.addLink(h5, s3, bw=10)
        self.addLink(h6, s3, bw=10)
        self.addLink(h7, s4, bw=10)
        self.addLink(h8, s4, bw=10)
        self.addLink(s1, s2, bw=10)
        self.addLink(s2, s3, bw=15, loss=5)
        self.addLink(s3, s4, bw=20)

if __name__ == '__main__':
    setLogLevel('info')

    topo = Network()
    net = Mininet(topo=topo, link=TCLink,
                  autoSetMacs=True, autoStaticArp=True)

    net.start()
    CLI(net)
```

```
59        net.stop()
```

a) Run an `iperf` measurement from h1 to h4 and another `iperf` measurement from h3 to h2. What is the measured data rate? Next run an `iperf` measurement from h1 to h3 and another `iperf` measurement from h2 to h4. What is the measured data rate? Explain your observation.

b) Run an `iperf` measurement from h1 to h4 and another `iperf` measurement from h3 to h6. What is the measured data rate? Explain your observation.