Network Simulation

Brandt Elison
Joe Eklund

January 26, 2016

1 Two Nodes

The following experiments were performed on a simulated two-node network with two unidirection links connecting each node to the other. Throughout the rest of this section, we will refer to these two nodes as n_1 and n_2.

This section details three experiments done with this network. Each experiment measures the delay on one or more packets being sent from n_1 to n_2. The experiments each use different values for link speed, propagation delay, and the number and timing of packets sent. The simulated packets in these experiments are all of size 1000 bytes.

Experiment 1

The following parameters were used for this experiment:

- Link speed: 1 Mbps
- Propagation delay: 1000 ms
- Packets sent: 1 packet at $t = 0$

The following code was used to create the network:

```
# The 'path' variable below references the following network configuration:
# n1 n2
# n2 n1
## link configuration
# n1 n2 1Mbps 1000ms
# n2 n1 1Mbps 1000ms

# setup network
net = Network(path)

# setup routes
n1 = net.get_node('n1')
n2 = net.get_node('n2')
n1.add_forwarding_entry(address=n2.get_address('n1'),link=n1.links[0])
n2.add_forwarding_entry(address=n1.get_address('n2'),link=n2.links[0])

# setup app
d = DelayHandler()
net.nodes['n2'].add_protocol(protocol="delay",handler=d)

p = packet.Packet(destination_address=n2.get_address('n1'),ident=1, \
```

```
22                      protocol='delay',length=1000)
23 Sim.scheduler.add(delay=0, event=p, handler=n1.send_packet)
```

The following calculations show that the expected delay for this simulation should be 1.008 seconds:

$delay_{trans} = \frac{8000bits}{10^6 bps} = .008s = 8ms$

$delay_{prop} = 1000ms$

$delay_{total} = delay_{trans} + delay_{prop} = 1008ms = 1.008s$

The simulator output is as follows:

```
1 1.008 1 0 1.008 0.008 1.0 0
```

The first element of the simulator output represents the total delay from transmitting the packet. This value for delay (1.008 seconds) matches the delay that we calculated by hand, which shows that the simulator is accurate.

Experiment 2

The following parameters were used for this experiment:

- Link speed: 100 bps
- Propagation delay: 10 ms
- Packets sent: 1 packet at $t = 0$

The following code was used to create the network:

```
1 # The 'path' variable below references the following network configuration:
2 # n1 n2
3 # n2 n1
4 ## link configuration
5 # n1 n2 100bps 10ms
6 # n2 n1 100bps 10ms
7
8 # setup network
9 net = Network(path)
10
11 # setup routes
12 n1 = net.get_node('n1')
13 n2 = net.get_node('n2')
14 n1.add_forwarding_entry(address=n2.get_address('n1'),link=n1.links[0])
15 n2.add_forwarding_entry(address=n1.get_address('n2'),link=n2.links[0])
16
17 # setup app
18 d = DelayHandler()
19 net.nodes['n2'].add_protocol(protocol="delay",handler=d)
20
21 p = packet.Packet(destination_address=n2.get_address('n1'),ident=1, \
22                      protocol='delay',length=1000)
23 Sim.scheduler.add(delay=0, event=p, handler=n1.send_packet)
```

The following calculations show that the expected delay for this simulation should be 80.01 seconds:

$delay_{trans} = \frac{8000bits}{100bps} = 80s$

$delay_{prop} = 10ms = .01s$

$delay_{total} = delay_{trans} + delay_{prop} = 80.01s$

The simulator output is as follows:

```
80.01 1 0 80.01 80.0 0.01 0
```

The first element of the simulator output represents the total delay from transmitting the packet. This value for delay (80.01 seconds) matches the delay that we calculated by hand, which shows that the simulator is accurate.

Experiment 3

The following parameters were used for this experiment:

- Link speed: 1 Mbps
- Propagation delay: 10 ms
- Packets sent: 3 packets at $t = 0$; 1 packet at $t = 2s$. We will refer to these packets as p_1, p_2, p_3, and p_4 respectively

The following code was used to create the network:

```
# The 'path' variable below references the following network configuration:
# n1 n2
# n2 n1
## link configuration
# n1 n2 1Mbps 10ms
# n2 n1 1Mbps 10ms

# setup network
net = Network(path)

# setup routes
n1 = net.get_node('n1')
n2 = net.get_node('n2')
n1.add_forwarding_entry(address=n2.get_address('n1'),link=n1.links[0])
n2.add_forwarding_entry(address=n1.get_address('n2'),link=n2.links[0])

# setup app
d = DelayHandler()
net.nodes['n2'].add_protocol(protocol="delay",handler=d)

p1 = packet.Packet(destination_address=n2.get_address('n1'),ident=1, \
                   protocol='delay',length=1000)
Sim.scheduler.add(delay=0, event=p1, handler=n1.send_packet)

p2 = packet.Packet(destination_address=n2.get_address('n1'),ident=1, \
                   protocol='delay',length=1000)
Sim.scheduler.add(delay=0, event=p2, handler=n1.send_packet)

p3 = packet.Packet(destination_address=n2.get_address('n1'),ident=1, \
                   protocol='delay',length=1000)
Sim.scheduler.add(delay=0, event=p3, handler=n1.send_packet)

# Late packet
```

```
34 p4 = packet.Packet(destination_address=n2.get_address('n1'),ident=1, \
35                     protocol='delay',length=1000)
36 Sim.scheduler.add(delay=2, event=p4, handler=n1.send_packet)
```

The following calculations show the expected delay for p_1:

$delay_{trans} = \frac{8000bits}{10^6 bps} = .008s = 8ms$

$delay_{prop} = 10ms = .01s$

$delay_{total} = delay_{trans} + delay_{prop} = .018s$

p_2, p_3, and p_4 are the same size as p_1, so each takes the same amount of time to go from n_1 to n_2 once they have begun to transmit. However, each of these packets is delayed for various reasons, so they do not arrive at $t = .018s$ like p_1 does.

p_2 and p_3 are delayed by queing delay as they wait for their turn to begin transmitting. Each must wait for the transmission delay of the previous packet(s) to elapse before it can begin to transmit, so the following delays exist for p_2 and p_3:

$delay_{p_2} = delay_{p_1} + delay_{trans} = .026s$

$delay_{p_3} = delay_{p_2} + delay_{trans} = .034s$

p_4 is delayed purely because it is transmitted 2 seconds later than the other packets. By the time it begins transmitting, the other packets have already arrived at n_2. Therefore,

$delay_{p_4} = 2s + delay_{p_1} = 2.018s$

The simulator output is as follows:

```
1 0.018 1 0 0.018 0.008 0.01 0
2 0.026 1 0 0.026 0.008 0.01 0.008
3 0.034 1 0 0.034 0.008 0.01 0.016
4 2.018 1 2.0 0.018 0.008 0.01 0.0
```

The first element of each line of the simulator output represents the total delay from transmitting each packet in order. The values for delay (.018, .026, .034, and 2.018 seconds) match the delay that we calculated by hand, which shows that the simulator is accurate.

2 Three Nodes

The following experiments were performed on a simulated three-node network with two unidirection links connecting n_1 with n_2 and two unidirection links connecting n_2 with n_3. Throughout the rest of this section, we will refer to these three nodes as n_1, n_2, and n_3.

This section details two experiments done with this network. Each experiment measures the delay on one or more packets being sent from n_1 to n_3. The experiments each use different values for link speeds and propagation delays. The simulated packets in these experiments are all of size 1000 bytes.

Experiment 1

The following parameters were used for this experiment:

- Link speed n_1 to n_2: 1 Mbps
- Link speed n_2 to n_3: 1 Mbps

- Propagation delay n_1 to n_2: 100 ms
- Propagation delay n_2 to n_3: 100 ms
- Packets sent: 1000 packets at $t = 0$

The following code was used to create the network:

```
# The 'path' variable below references the following network configuration:
#n1 n2
#n2 n1 n3
#n3 n2
# link configuration
#n1 n2 1Mbps 100ms
#n2 n1 1Mbps 100ms
#n2 n3 1Mbps 100ms
#n3 n2 1Mbps 100ms


# setup network
net = Network(path)

# setup routes
n1 = net.get_node('n1')
n2 = net.get_node('n2')
n3 = net.get_node('n3')
n1.add_forwarding_entry(address=n2.get_address('n1'),link=n1.links[0])
n1.add_forwarding_entry(address=n3.get_address('n2'),link=n1.links[0])
n2.add_forwarding_entry(address=n1.get_address('n2'),link=n2.links[0])
n2.add_forwarding_entry(address=n3.get_address('n2'),link=n2.links[1])
n3.add_forwarding_entry(address=n1.get_address('n2'),link=n3.links[0])
n3.add_forwarding_entry(address=n2.get_address('n3'),link=n3.links[0])

# setup app
d = DelayHandler()
net.nodes['n3'].add_protocol(protocol="delay",handler=d)

for i in range(1000):
  p = packet.Packet(destination_address=n3.get_address('n2'),ident=1,protocol='delay',len
  Sim.scheduler.add(delay=0, event=p, handler=n1.send_packet)
```

The simulator output is as follows:

```
0.216,1,0,0.216,0.016,0.2,0.0
0.224,1,0,0.224,0.016,0.2,0.008
0.232,1,0,0.232,0.016,0.2,0.016
0.24,1,0,0.24,0.016,0.2,0.024
...
8.192,1,0,8.192,0.016,0.2,7.976
8.2,1,0,8.2,0.016,0.2,7.984
8.208,1,0,8.208,0.016,0.2,7.992
```

If we were to change both link speeds to 1 Gbps, the simulator produces the following:

```
0.200016 1 0 0.200016 1.6e-05 0.2 0.0
0.200024 1 0 0.200024 1.6e-05 0.2 8e-06
0.200032 1 0 0.200032 1.6e-05 0.2 1.6e-05
```

```
0.20004  1  0  0.20004  1.6e-05  0.2  2.4e-05
...
0.207992  1  0  0.207992  1.6e-05  0.2  0.007976
0.208  1  0  0.208  1.6e-05  0.2  0.007984
0.208008  1  0  0.208008  1.6e-05  0.2  0.007992
```