Hands-on Session

TCP and UDP over Lossy Channel

Nov 8th, 2012

Transfer “Odyssey by Homer”

- Use TCP and UDP
- Drop packets and check resulting file
- Capture traffic and analyze the pcap trace

TCP Topology

TCP sender:

```
file = open("odyssey.mb.txt", "rb")
        sock = socket.socket()
        sock.connect(("192.168.1.1", 8021))

        while True:
                chunk = file.read(65536)
                if not chunk:
                        break  # EOF
                sock.sendall(chunk)
```

TCP receiver:

>> nc –l 8021

UDP Topology

TCP sender:

```
file = open("odyssey.mb.txt", "rb")
Sock.socket(socket.AF_INET, socket.SOCK_DGRAM)

while True:
        chunk = file.read(1024)
        if not chunk:
                break  # EOF
        sock.sendto(chunk,("192.168.1.1", 8021))
```

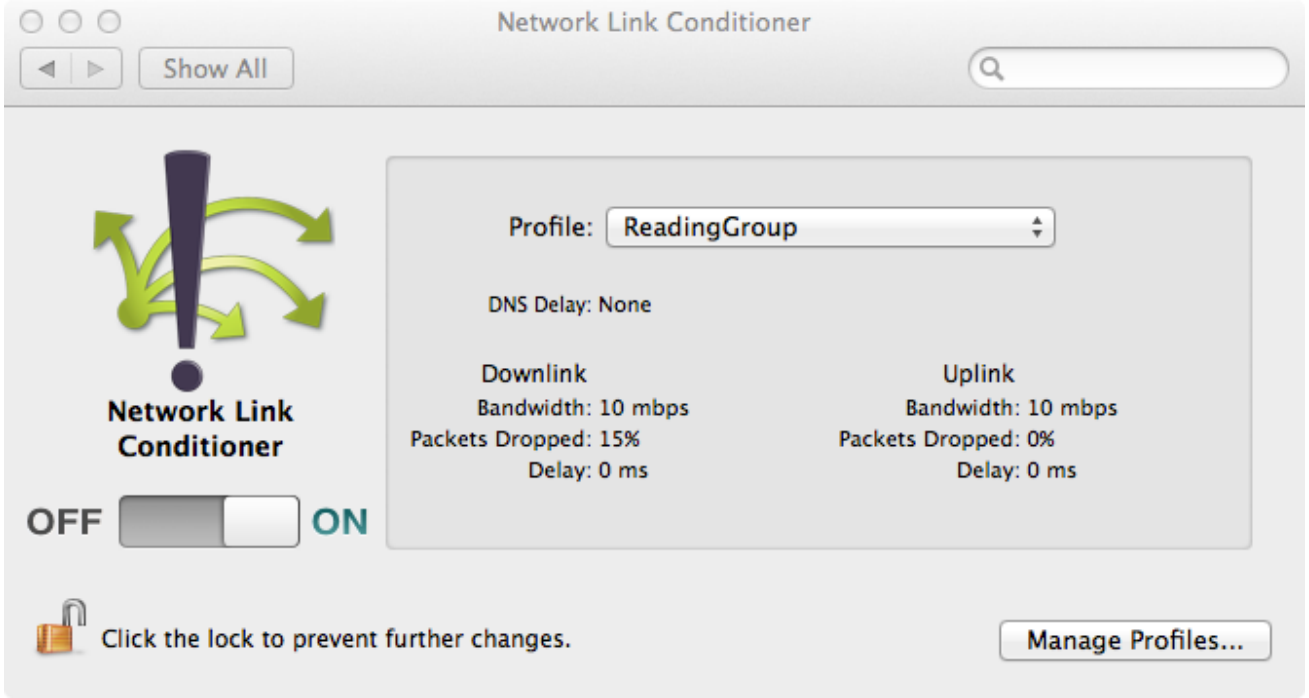
TCP receiver:

```
sock = socket.socket(socket.AF_INET,
socket.SOCK_DGRAM)

sock.bind(('', 8021))

while True:
   print sock.recv(1024)
```

Dropping Packets



Network Link Conditioner Tool (Mac OS X)

- We are **dropping 15%** of the Downlink packets
- Uplink packets are not intentionally dropped

Xcode > Open Developer Tool > More Developer Tools… and get "Hardware IO Tools for Xcode".

Experiment 1

- Send the file using TCP without dropping packets
- Trace file: exp1.pcap

Experiment 2

- Send the file using UDP without dropping packets
- Trace file: exp2.pcap

Experiment 3

- Send the file using TCP while dropping 1% of the downlink packets at the destination
- Trace file:
  - exp3_sender.pcap
    - Check packet number 284
  - Exp3_recv.pcap
    - Check packet number 278

Experiment 4

- Send the file using UDP while dropping 15% of the downlink packets at the destination
- Trace file:
  - Exp4_recv.pcap
  - output.txt