CN-Basic
L11/12

Networking tools

Dr. Ram P Rustagi
rprustagi@ksit.edu.in
http://www.rprustagi.com
https://www.youtube.com/rprustagi

Tools - `nc`

- `nc` (netcat)
  - Works as both layer client & server
  - Supports both TCP and UDP
  - Supports both IPv4 and IPv6
  - Common use
    - Simple TCP/UDP based data transfer
    - Shell script based HTTP clients and servers
    - Network daemon testing
    - SOCKS or HTTP ProxyCommand for `ssh`

Tools - nc

- `nc` **usage**
  - `-l` **acting as server**
  - `-u` **use UDP**
  - `-i` **for interval based transmission (lines)**
  - `-k` **for keeping server up after connection close**
- Examples
  - `nc` **<servername> <server port> # client**
  - `nc -l` **port # server (add `-p` on windows)**
  - **To transfer files**
    - **Server:** `nc -l` **<port> >file.dat**
    - **Client:** `cat` **<file> `| nc` <servername> <port>**

Usage: nc

- Terminating connection after idle time

  `nc -w 10` <server> <port> # timeout 10s

  - Don't use with server option

- Accessing a web server

  - nc -C mywww.com 80

  - …

Tools - wget

- **wget:** `wget` [options] <url>
  - Options
  - `-d` # to debug headers
  - `-O` <file> # to save with different name
  - `-i` <urlsfile> # list of urls in a file
  - `-c` # to resume to broken download
  - `-b` # run in background
  - `--limit-rate=`**500k**
  - `--header=`**<headers>**
  - `--http-user=`**user** `--http-password=`**..**
  - `-mk` # mirroring with local link conversion
  - …

Tools - curl

- General invocation, displays content
  `curl` <url>

- To specify specific headers
  `curl -H` <header> <urls>

- To dump HTTP headers
  `curl -D` <file> <url>

Tools - ping

- `ping` (Packet Inter Network Groper)
  - Checking reachability
    - `ping` hostname
  - `-i` changing packet interval
  - `-c` packet count
  - `-f` flooding the network
  - `-q` quite mode
  - `-s` change packet size
  - `-W` response timeout

Usage: ping

```
$ ping -c 10 www.google.com
PING www.google.com (74.125.203.104): 56 data bytes
64 bytes from 74.125.203.104: icmp_seq=0 ttl=47 time=93.346 ms
64 bytes from 74.125.203.104: icmp_seq=1 ttl=47 time=94.300 ms
64 bytes from 74.125.203.104: icmp_seq=2 ttl=47 time=99.392 ms
64 bytes from 74.125.203.104: icmp_seq=3 ttl=47 time=140.457 ms
64 bytes from 74.125.203.104: icmp_seq=4 ttl=47 time=168.882 ms
64 bytes from 74.125.203.104: icmp_seq=5 ttl=47 time=218.813 ms
64 bytes from 74.125.203.104: icmp_seq=6 ttl=47 time=270.833 ms
64 bytes from 74.125.203.104: icmp_seq=7 ttl=47 time=312.594 ms
64 bytes from 74.125.203.104: icmp_seq=8 ttl=47 time=263.280 ms
64 bytes from 74.125.203.104: icmp_seq=9 ttl=47 time=309.129 ms


--- www.google.com ping statistics ---
10 packets transmitted, 10 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 93.346/197.103/312.594/84.298 ms
```
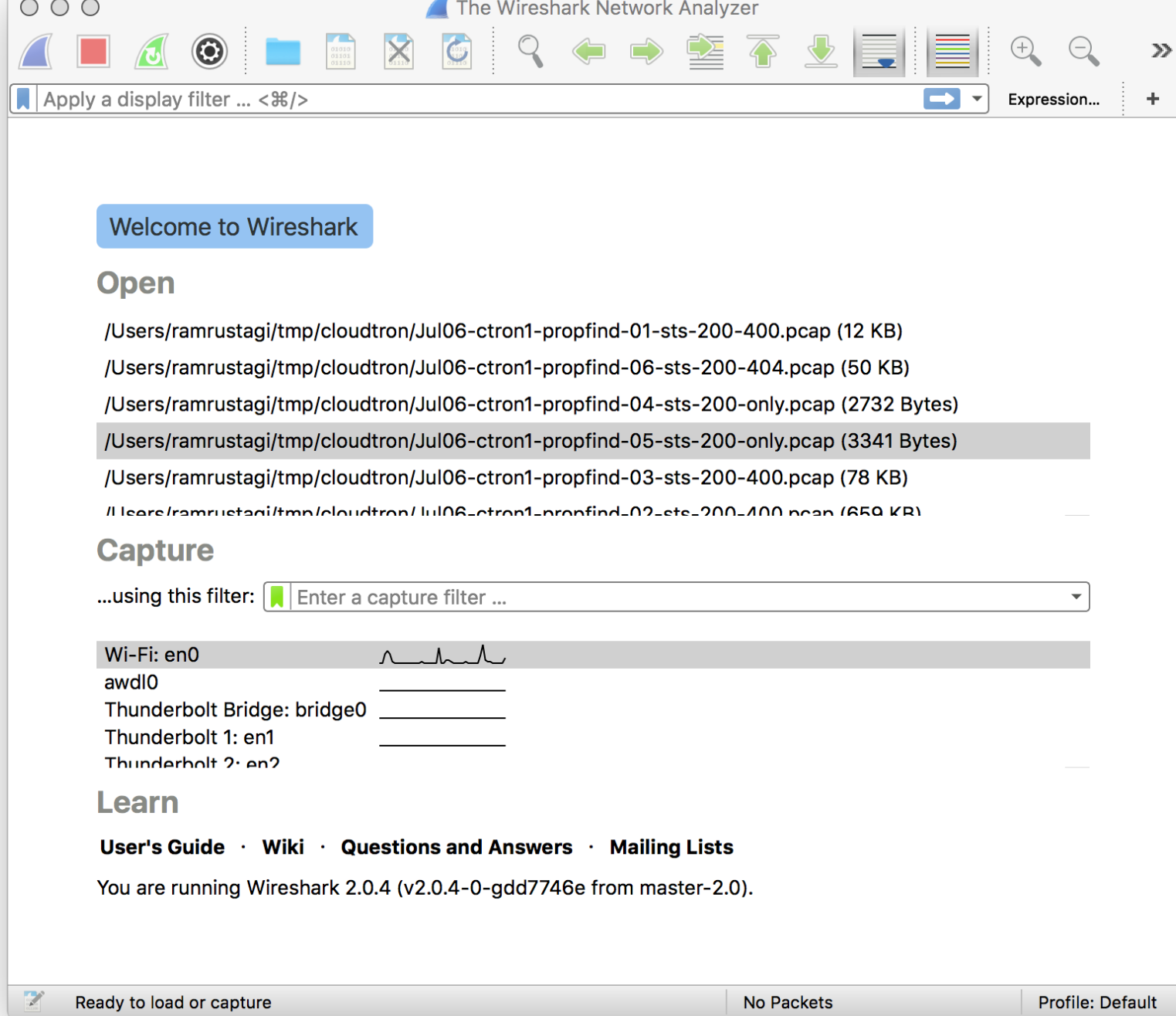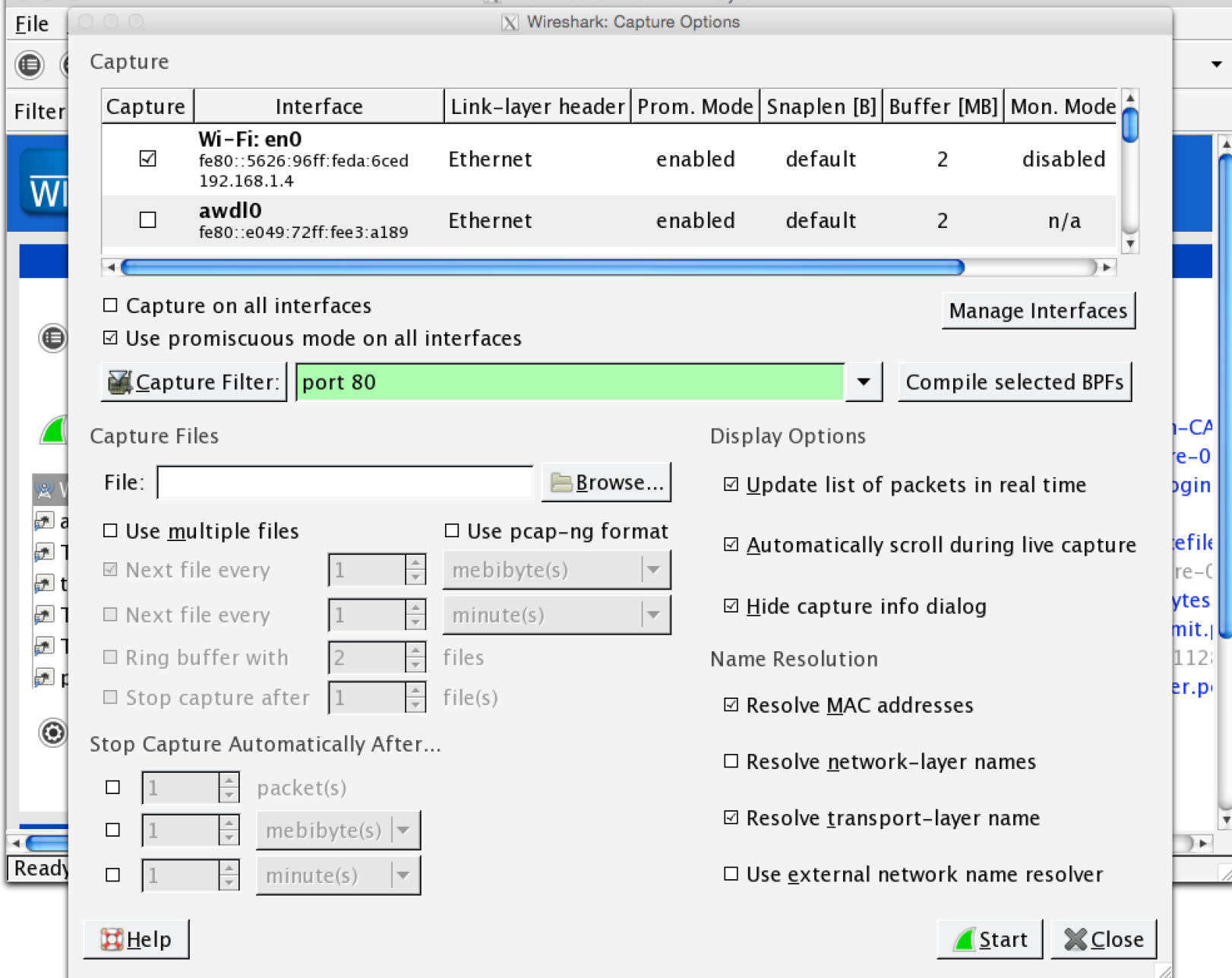
Tools - traceroute

- `traceroute` **(Windows:** `tracert`**)**
  - **Checking reachability**
    - `traceroute` **hostname**
  - **??**

Tools

- `wireshark`
  - `https://www.wireshark.org/docs/wsug_html_chunked/`
  - Enables viewing of bytes sent/recd on network
  - Capture and Display filters
  - Graphical, built on tcpdump
  - TCP session display
  - Changing UI options
- `tcpdump:` command line capture tool
  - `-o` output file
  - `-c` packet count
  - `-i` interface names

The Wireshark Network Analyzer

Apply a display filter ... <⌘/>

Expression... +

Welcome to Wireshark

Open

/Users/ramrustagi/tmp/cloudtron/Jul06-ctron1-propfind-01-sts-200-400.pcap (12 KB)

/Users/ramrustagi/tmp/cloudtron/Jul06-ctron1-propfind-06-sts-200-404.pcap (50 KB)

/Users/ramrustagi/tmp/cloudtron/Jul06-ctron1-propfind-04-sts-200-only.pcap (2732 Bytes)

/Users/ramrustagi/tmp/cloudtron/Jul06-ctron1-propfind-05-sts-200-only.pcap (3341 Bytes)

/Users/ramrustagi/tmp/cloudtron/Jul06-ctron1-propfind-03-sts-200-400.pcap (78 KB)

/Users/ramrustagi/tmp/cloudtron/Jul06-ctron1-propfind-02-sts-200-400.pcap (659 KB)

Capture

...using this filter: Enter a capture filter ...

Wi-Fi: en0

awdl0

Thunderbolt Bridge: bridge0

Thunderbolt 1: en1

Thunderbolt 2: en2

Learn

User's Guide · **Wiki** · **Questions and Answers** · **Mailing Lists**

You are running Wireshark 2.0.4 (v2.0.4-0-gdd7746e from master-2.0).

Ready to load or capture

No Packets

Profile: Default

Identifying Layers

Wireshark: UI Options

- Color coding
- Time format
- Packet reordering (in display)
- Defining protocol
- Using display filter
- Following TCP Stream
- Capture packets with proper capture filter
    - Needed for analyzing packet layers

Wireshark capture filters

- Traffic between A and either B or C
  ```
  host A and \( B or C\)
  ```
- Traffic between A any host except B
  ```
  host A and not B
  ```
- Capture web traffic
  ```
  port 80
  ```
- Capture ICMP traffic (e.g. ping, traceroute)
  ```
  icmp
  ```
- Capture web traffic
  ```
  port 80
  ```
- Capture traffic for a subnet
  ```
  net 10.211.55.0/24
  ```

Wireshark Display Filters

- **Source IP filter**
  - `ip.src == 10.1.1.1`
- **Destination IP filter**
  - `ip.dst == 10.1.1.101`
  - `ip.dst != 10.1.1.101`
- **Protocol filter**
  - `http || icmp`
- **Port number**
  - `tcp.port eq 80`
- **TCP Seq**
  - `tcp.stream eq 1`

Wireshark - Misc

- Saving file
  - Saving selected packets
- Reading from file
- Time display format
- Statistics
- Other options

tcpdump

- Command line interface
  - ASCII content
  - Capture full packet
  - Capture filters
  - Output file
  - Ethernet frame display

`tcpdump` - Usage Examples

- To capture between two specified hosts and save
  ```
  $ sudo tcpdump -n -i eth0 -s 0 —w
  file.pcap host <A> and host <B>
  ```

- To capture 100 packets received from host X
  ```
  $ sudo tcpdump -n -i eth0 -c 100
  src <X>
  ```

- To capture first 256 bytes and display in ASII
  ```
  $ sudo tcpdump -n -i eth0 -s 256 -
  A
  ```

- To capture with (link level) ethernet headers
  ```
  $ sudo tcpdump -n -i eth0 -e
  ```

Use Case `1` : Using `nc`

- Take two systems in your home/lab and connect to a network. These can be connected by directing connecting them wia a LAN Cable (ethernet). If LAN cable is a issue, Enable mobile hotspot on your phone and connect the labopts to your mobile hotspot.

- If windows machine, install "netcat for windows"
  - (https://joncraton.org/blog/46/netcat-for-windows/)
  - It is available by default on Linux/Mac.

- Identify the IP address of each system.

- Establish a chat session between and exchange few messages.

Use Case 2: Using `ping`

- Connect your in your home/lab to internet.
- Check reachability of google.com by sending 10 pkts
  - ping -c10 www.google.com (linux/mac)
  - ping -n 10 (www.google.com) (windows)
- Analyze the response time (min, max, avg and standard deviation).
- Repeat the above with 100 pkts with pkt size of 1000 bytes and analyze the response.
- Repeat the above 200 pkts, break down your internet connectivity for 30 seconds after 50 pkts
  - Analyze the packet loss and response time.

Use Case 3 : Using `traceroute`

- Connect your in your home/lab to internet.
- Traceoute <u>google.com</u> and identify all the routers in the path your home to google
- Analyze the response times of various routers.
  - Do you find any router at n^{th} hop which takes more time than router at $(n+1)^{th}$ hop
- Repeat the above for your other favourite websites.

Use Case 4: Using `wget/curl`

- use wget to download a file using rate limit.
- Use curl to dump headers of a URL.

Use Case 5: Using `wireshark`

- Download and install wireshark on your sytem (www.wireshark.org).

- Launch wireshark application, select the active internet interface (e.g.etherent, wireless etc or even any) and start capturing with any caputre filter.

  - Access your favourite website e.g. your academic institution website in the browse.

  - Stop the capture after 1 minute in wireshark.

- Analyze the number of packets received.

- Identify all the protocols that are identified by wireshark

Use Case 6: Using `wireshark`

- Connect two systems on the network and identify their IP addresses.
- Launch wireshark application and specify capture filter as other system's IP address.
- Initiate a chat session (using `nc`) between two systems and exchange few messages.
- Analyze the nc chat packets in wireshark.
- Analyze the number of packets received.
- Explore why there is no layer 7 packet in wireshark even though you are using nc as an application.

Use Case 7 : Using `tcpdump`

- Connect two systems on the network and identify their IP addresses.
- Launch `tcpdump` application with capture filter as IP address of your college website with packet size of 1500
- Open a browser and access your college website.
- Analyze the first 5 packets in tcpdump.
  - Analyze 3-way handshake
  - Analyze application data sent
  - Analyze response received

Summary

- `nc`
- `ping`
- `traceroute`
- `wget`
- `curl`
- `tcpdump`
- `wireshark`