

VASAVI COLLEGE OF ENGINEERING (Autonomous)
IBRAHIMBAGH, HYDERABAD – 500 031
Department of Computer Science & Engineering
INNOVATION IN TEACHING

Course Name: CN (Computer Networks)
Topic Name: Network layer Functionalities

Faculty Name: T Nishitha
Year/Sem: III year/V Sem

Teaching aid/Tool Used: Network Simulator-3

Description of the Tool:

NS-3 is a discrete-event network simulator for Internet systems, targeted primarily for research and educational use.

Tool usage in Teaching:

NS-3 installation and execution of example programs

List of packages for installing ns-3 in Ubuntu operating system

Packages required for installing ns-3:

1. gcc
2. g++
3. python
4. python-dev

Packages required for installing netanim:

1. qt4-dev-tools

packages required for installing PyViz:

1. libgtk-3-dev
2. python-pygoocanvas
3. python-pygraphviz

packages required for installing TraceMetrics:

1. openjdk-8-jdk (openjdk-7-jdk if OS is less than Ubuntu 16.04)

packages required for installing gnuplot:

1. gnuplot

packages required for installing wireshark:

1. wireshark

command to install all packages at once:

```
sudo apt-get install gcc g++ python python-dev qt4-dev-tools libgtk-3-dev python-pygoocanvas python-pygraphviz openjdk-8-jdk gnuplot wireshark
```

steps to install ns-3:

1. Download ns-allinone-3.26.tar.bz2 and unzip it.
2. Go to ns-allinone-3.26 and give the following command:
./build.py --enable-examples --enable-tests
(This command will install ns-3, NetAnim and PyViz)

You are done with it, if you do not see any errors!

Similar instructions to install ns-3 can be found at the following link:

<http://mohittahiliani.blogspot.in/2015/10/installing-ns-3-on-ubuntu-simplified.html>

details to install ns-3 in other operating systems can be found at the following link:

<http://www.nsnam.org/wiki/installation>

Steps to play with first.cc

Step 1: Copy first.cc from ns-allinone-3.26/ns-3.26/examples/tutorial

Step 2: paste it in ns-allinone-3.26/ns-3.26/scratch

Step 3: go to ns-allinone-3.26/ns-3.26 via terminal and run the program with following command:

(Note: ./waf command must be always run from ns-allinone-3.26/ns-3.26)

```
./waf --run scratch/first
```

(above command prints the output on the console)

Step 4: to see animated output, run the program with following command:

```
./waf --run scratch/first --vis
```

(above command animates the output using PyViz, if it is installed)

Changes required to generate .pcap files for first.cc

Step 5: open first.cc that is placed in ns-allinone-3.26/ns-3.26/scratch

Step 6: Type the following line before “Simulator::Run() :”

```
PointToPoint.EnablePcapAll(“first”);
```

Step 7: run the program with following command:

```
./waf --run scratch/first
```

(above command prints the output on console and generates two pcap files)

Step 8: check whether two pcap files are generated by using the following

```
Ls *.pcap
```

(above command displays all the pcap files in ns-3.26 directory)

Step 9: open the first pcap file by using the following command:

```
Wireshark first-0-0.pcp
```

(above command opens the pcap file in wireshark)

Changes required to use NetAnim and generate .xml file for first.cc

Step 10: Add the following header files in first.cc

```
#include “ns3/netanim-module.h”
```

```
#include “ns3/mobility-module.h”
```

Step 11: type the following lines before “Simulator::Run() ;”

```
MobilityHelper mobility;
```

```
mobility.SetMobilityModel (“ns3::ConstantPositionMobilityModel”);
```

```
mobility.Install (nodes);
```

```
AnimationInterface anim (“first.xml”);
```

```
AnimationInterface::SetConstantPosition (nodes.Get (0), 10, 25);
```

```
AnimationInterface::SetConstantPosition (nodes.Get (1), 40, 25);
```

```
Anim.EnablePacketMetadata (true); //what protocol, port no//do it false//
```

Step 12: Run the program with following command:

```
./waf -run scratch/first
```

(above command prints the output on console and generates one xml file)

Step 13: check whether the xml file is generated by using the following

```
Ls *.xml
```

(above command displays all the xml files in ns-3.26 directory)

Step 14: Load the xml file after giving the following command:

```
../netanim-3.107/NetAnim
```

(above command opens the NetAnim window if it is installed)

Topic Name: Network layer Functionalities

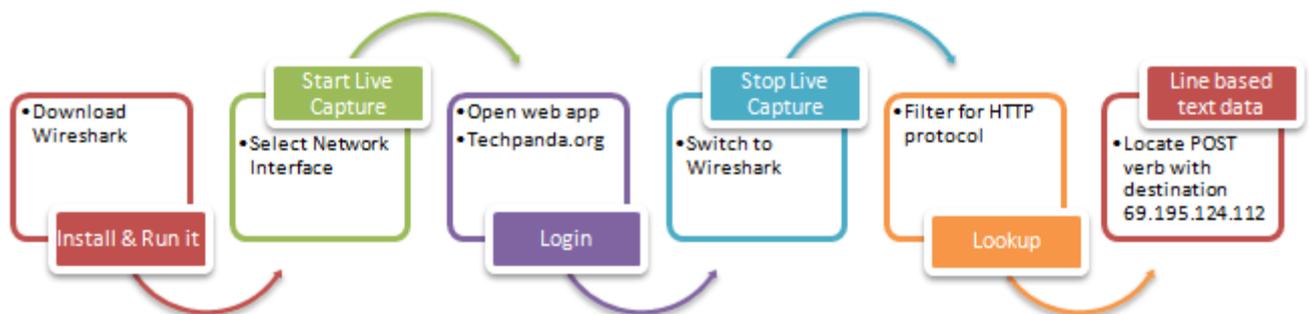
Teaching aid/Tool Used: Wireshark

Description of the Tool: Capture, Filter and inspect packets using Wireshark

Wireshark, a network analysis tool captures packets in real time and display them in human-readable format. Wireshark includes filters, color coding, and other features that let you dig deep into network traffic and inspect individual packets.

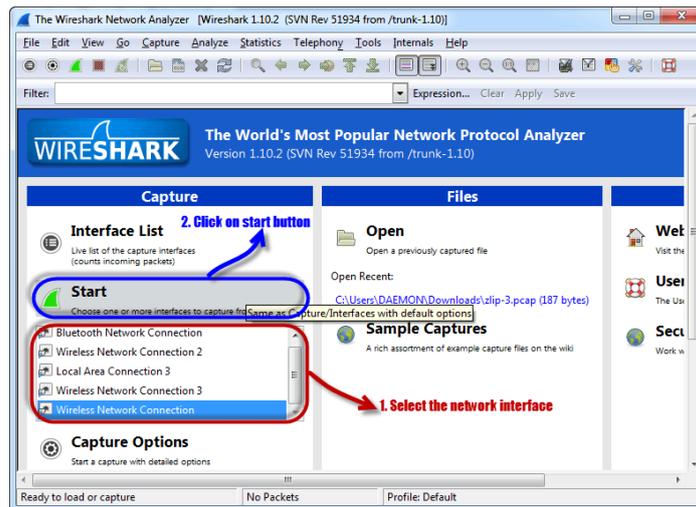
Professionals use it to debug network protocol implementations, examine security problems and inspect network protocol internals.

Tool usage in Teaching: The illustration below shows you the steps that are required to understand sniffing of packets using Wireshark.



Step 1: Download Wireshark from this link <http://www.wireshark.org/download.html>

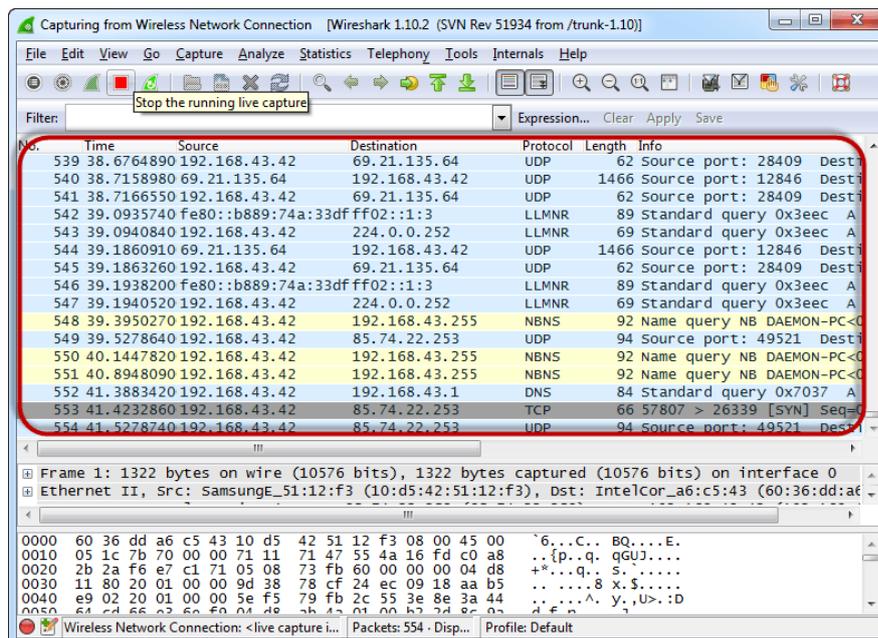
- Open Wireshark
- You will get the following screen



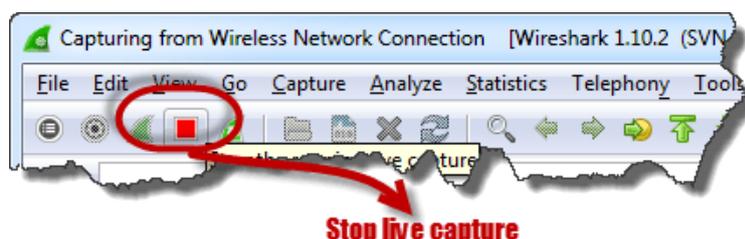
Step 2: Select the network interface you want to sniff. Note for this demonstration, we are using a wireless network connection. If you are on a local area network, then you should select the local area network interface.

Step 3: Click on start button as shown above

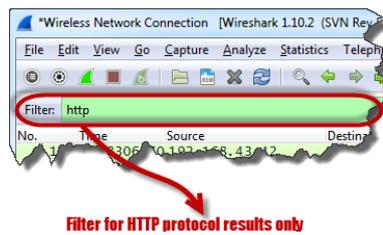
Step 4: Open your web browser and type <http://www.vce.ac.in/>



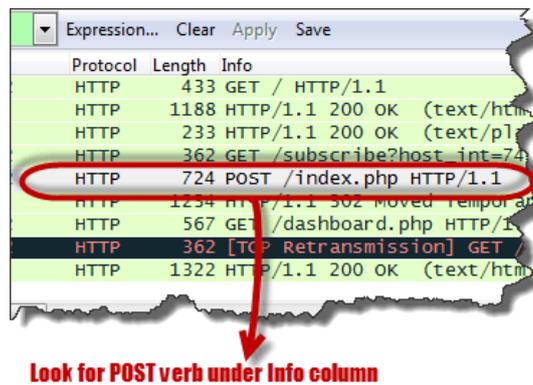
Step 5: Go back to Wireshark and stop the live capture



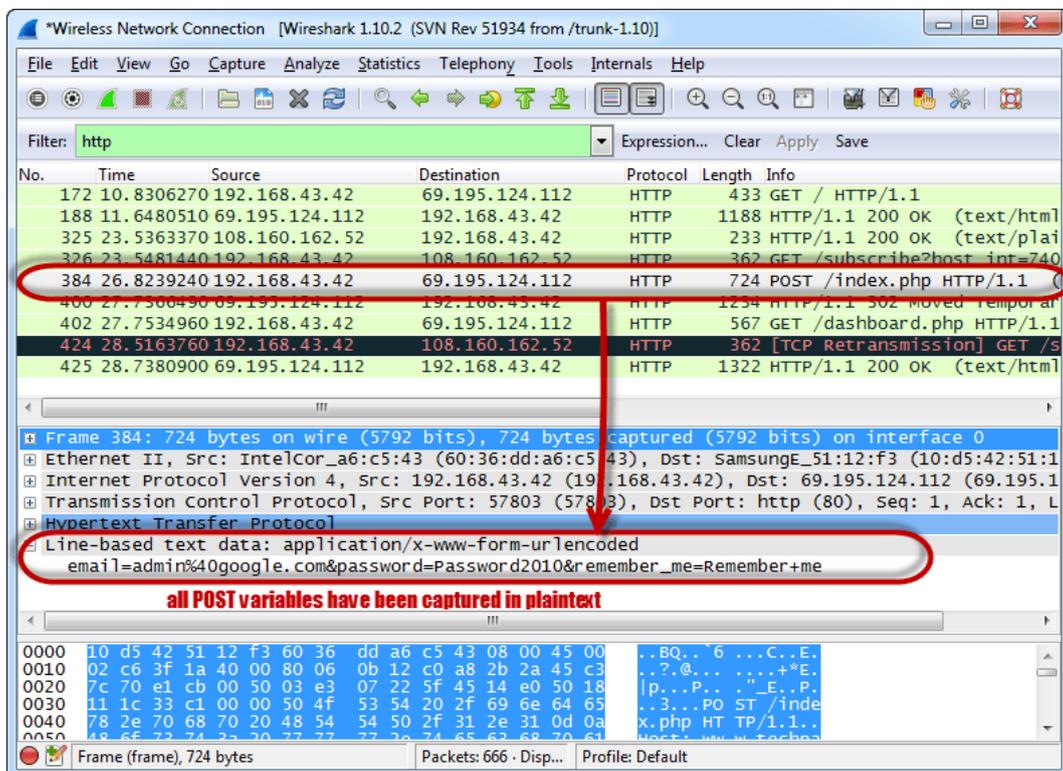
Step 6: Filter for HTTP protocol results only using the filter textbox



Step 7: Locate the Info column and look for entries with the HTTP verb POST and click on it



Step 8: Just below the log entries, there is a panel with a summary of captured data. Look for the summary that says Line-based text data: application/x-www-form-urlencoded



- You should be able to view the plaintext values of all the POST variables submitted to the server via HTTP protocol.

Summary

- Network sniffing is intercepting packages as they are transmitted over the network
- Passive sniffing is done on a network that uses a hub. It is difficult to detect.
- Active sniffing is done on a network that uses a switch. It is easy to detect.

The screenshot shows the Wireshark interface with a network capture. A red box highlights a list of packets. The table below represents the data from this list:

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|------------|---------------------|----------------|----------|--------|---------------------------|
| 539 | 38.6764890 | 192.168.43.42 | 69.21.135.64 | UDP | 62 | Source port: 28409 Desti |
| 540 | 38.7158980 | 69.21.135.64 | 192.168.43.42 | UDP | 1466 | Source port: 12846 Desti |
| 541 | 38.7166550 | 192.168.43.42 | 69.21.135.64 | UDP | 62 | Source port: 28409 Desti |
| 542 | 39.0935740 | fe80::b889:74a:33df | ff02::1:3 | LLMNR | 89 | Standard query 0x3eec A |
| 543 | 39.0940840 | 192.168.43.42 | 224.0.0.252 | LLMNR | 69 | Standard query 0x3eec A |
| 544 | 39.1860910 | 69.21.135.64 | 192.168.43.42 | UDP | 1466 | Source port: 12846 Desti |
| 545 | 39.1863260 | 192.168.43.42 | 69.21.135.64 | UDP | 62 | Source port: 28409 Desti |
| 546 | 39.1938200 | fe80::b889:74a:33df | ff02::1:3 | LLMNR | 89 | Standard query 0x3eec A |
| 547 | 39.1940520 | 192.168.43.42 | 224.0.0.252 | LLMNR | 69 | Standard query 0x3eec A |
| 548 | 39.3950270 | 192.168.43.42 | 192.168.43.255 | NBNS | 92 | Name query NB DAEMON-PC<C |
| 549 | 39.5278640 | 192.168.43.42 | 85.74.22.253 | UDP | 94 | Source port: 49521 Desti |
| 550 | 40.1447820 | 192.168.43.42 | 192.168.43.255 | NBNS | 92 | Name query NB DAEMON-PC<C |
| 551 | 40.8948090 | 192.168.43.42 | 192.168.43.255 | NBNS | 92 | Name query NB DAEMON-PC<C |
| 552 | 41.3883420 | 192.168.43.42 | 192.168.43.1 | DNS | 84 | Standard query 0x7037 A |
| 553 | 41.4232860 | 192.168.43.42 | 85.74.22.253 | TCP | 66 | 57807 > 26339 [SYN] Seq=C |
| 554 | 41.5278740 | 192.168.43.42 | 85.74.22.253 | UDP | 94 | Source port: 49521 Desti |

Below the packet list, the details of Frame 1 are shown:

```

Frame 1: 1322 bytes on wire (10576 bits), 1322 bytes captured (10576 bits) on interface 0
Ethernet II, Src: samsungE_51:12:f3 (10:d5:42:51:12:f3), Dst: Intelcor_a6:c5:43 (60:36:dd:a6:
0000  60 36 dd a6 c5 43 10 d5 42 51 12 f3 08 00 45 00  6...C... BQ...E.
0010  05 1c 7b 70 00 00 71 11 71 47 55 4a 16 fd c0 a8  ..{p..q. qGUJ....
0020  2b 2a f6 e7 c1 71 05 08 73 fb 60 00 00 00 04 d8  +*...q.. s.....
0030  11 80 20 01 00 00 9d 38 78 cf 24 ec 09 18 aa b5  .. ...8 x.$.....
0040  e9 02 20 01 00 00 5e f5 79 fb 2c 55 3e 8e 3a 44  .. ...^.. y.,U>.:D
0050  64 cd 66 02 60 f0 04 d8 2b 42 01 00 b2 2d 8c 02  d f n  ?

```