Lab Manual for Computer Communication and Networking

Lab No. 10

Introduction of Wireshark

BAHRIA UNIVERSITY KARACHI CAMPUS
Department of Software Engineering

COMPUTER COMMUNICATION & NETWORKING

LAB EXPERIMENT # 10

Introduction to Wireshark

OBJECTIVE: -

- Getting familiar with Wireshark.

THEORY: -

Wireshark is a network packet analyzer.

- A network packet analyzer will try to capture network packets and tries to display that packet Computer as detailed as possible
- Identifying and analyzing protocols
- Identifying source & destination of traffic

In the past, such tools were either very expensive, proprietary, or both. However, with the advent of Wireshark, all that has changed.

Purposes

Here are some examples to use Wireshark for:

- network administrators use it to **troubleshoot network problems**
- network security engineers use it to **examine security problems**
- developers use it to **debug protocol implementations**
- people use it to **learn network protocol** internals

Beside these examples, Wireshark can be helpful in many other situations too.

Features

The following are some of the many features Wireshark provides:

- Available for **UNIX** and **Windows**.
- **Capture** live packet Computer from a network interface.
- Display packets with **very detailed protocol information**.
- **Open and Save** packet Computer captured.
- **Import and Export** packet Computer from and to a lot of other capture programs.
- **Filter packets** on many criteria.
- **Search** for packets on many criteria.
- **Colorize** packet display based on filters.
- Create various **statistics**... and **a lot more!**

Running Wireshark:

When you run the Wireshark program, the Wireshark graphical user interface shown in Figure 2 will be displayed. Initially, no Computer will be displayed in the various windows.

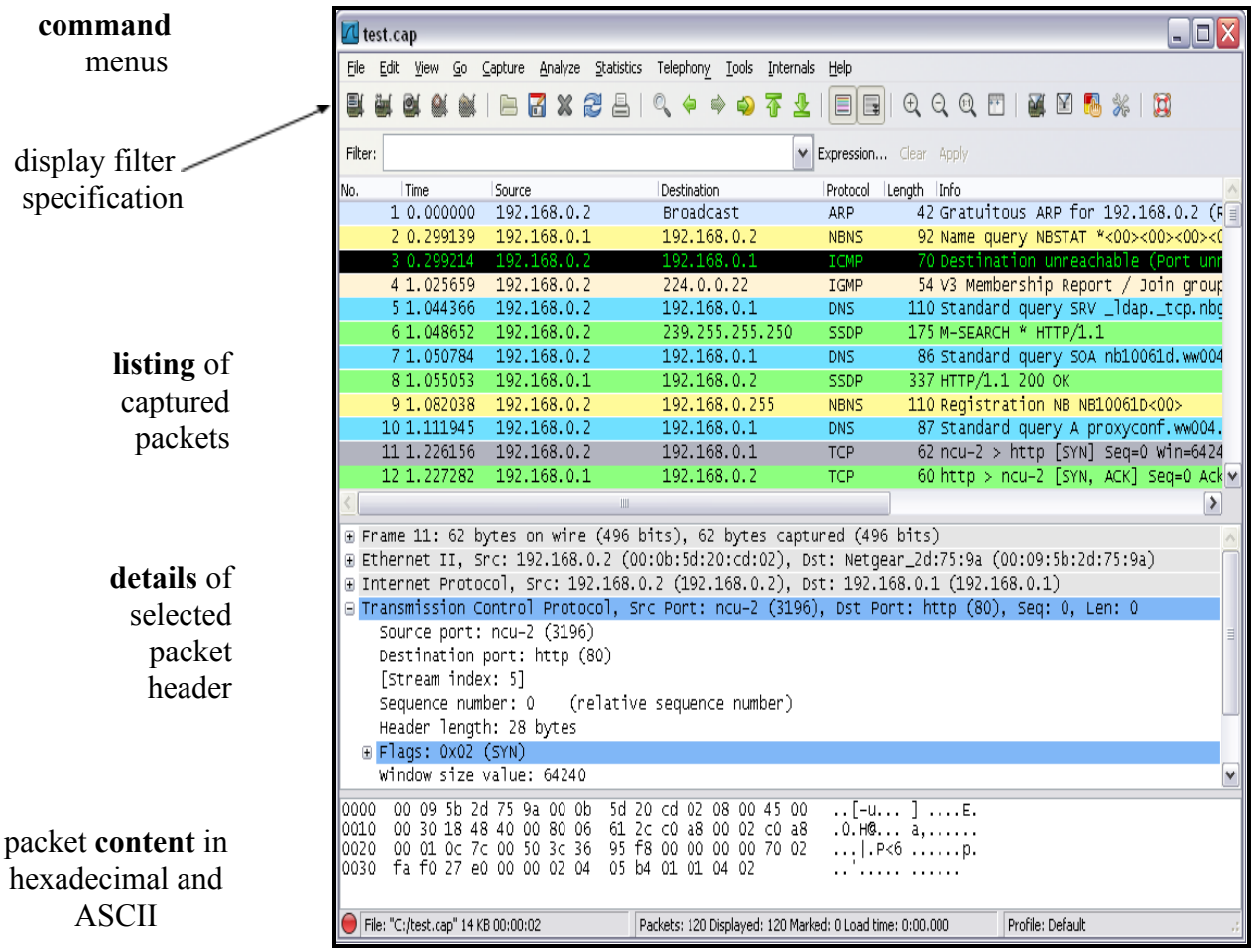


Figure 2: Wireshark Graphical User Interface

Wireshark Interface Components:

The Wireshark interface has five major components:

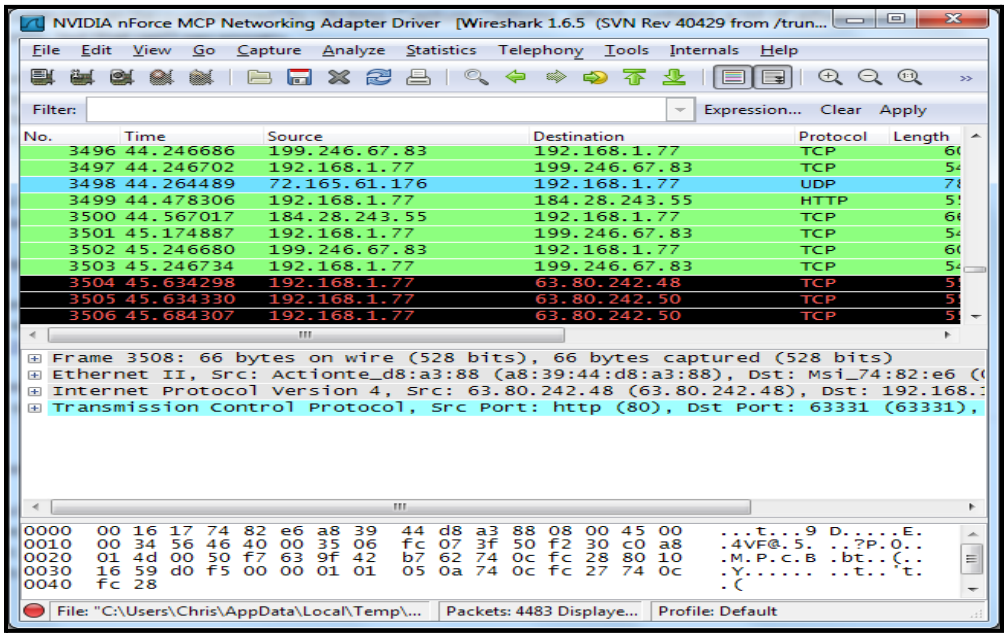
- The **command menus** are standard pulldown menus located at the top of the window. Of interest to us now are the File and Capture menus. The File menu allows you to save captured packet Computer or open a file containing previously captured packet Computer, and exit the Wireshark application. The Capture menu allows you to begin packet capture.
- The **packet-listing window** displays a one-line summary for each packet captured, including the packet number (assigned by Wireshark; this is *not* a packet number contained in any protocol's header), the time at which the packet was captured, the packet's source and destination addresses, the protocol type, and protocol-specific information contained in the packet. The packet listing can be sorted according to any of these categories by clicking on a column name. The protocol type field lists the highest-level protocol that sent or received this packet, i.e., the protocol that is the source or ultimate sink for this packet.
- The **packet-header details window** provides details about the packet selected (highlighted) in the packet listing window. (To select a packet in the packet listing

window, place the cursor over the packet's one-line summary in the packet listing window and click with the left mouse button.). These details include information about the Ethernet frame (assuming the packet was sent/received over an Ethernet interface) and IP Computergram that contains this packet. The amount of Ethernet and IP-layer detail displayed can be expanded or minimized by clicking on the plus-or-minus boxes to the left of the Ethernet frame or IP Computergram line in the packet details window. If the packet has been carried over TCP or UDP, TCP or UDP details will also be displayed, which can similarly be expanded or minimized. Finally, details about the highest-level protocol that sent or received this packet are also provided.

- The **packet-contents window** displays the entire contents of the captured frame, in both ASCII and hexadecimal format.
- Towards the top of the Wireshark graphical user interface, is the **packet display filter field,** into which a protocol name or other information can be entered to *filter* the information displayed in the packet-listing window (and hence the packet-header and packet-contents windows). In the example below, we'll use the packet-display filter field to have Wireshark hide (not display) packets except those that correspond to HTTP messages.

Color Coding:

You'll probably see packets highlighted in green, blue and black. Wireshark uses colors to help you identify the types of traffic at a glance. By default, green is TCP traffic, dark blue is DNS traffic, light blue is UDP traffic and black identifies TCP packets with problems — for example, they could have been delivered out-of-order.



Taking Wireshark for a Test Run:

<u>Capturing Packets</u>

1. Start up your favorite web browser, which will display your selected homepage.

2. Start up the Wireshark software. You will initially see no packet Computer will be displayed in the packet- listing, packet-header, or packet-contents window, since Wireshark has not yet begun capturing packets.
3. To begin packet capture, select the Capture pull down menu and select *Options.*
This will cause the "Wireshark: Capture Options" window to be displayed, as shown in Figure 3.

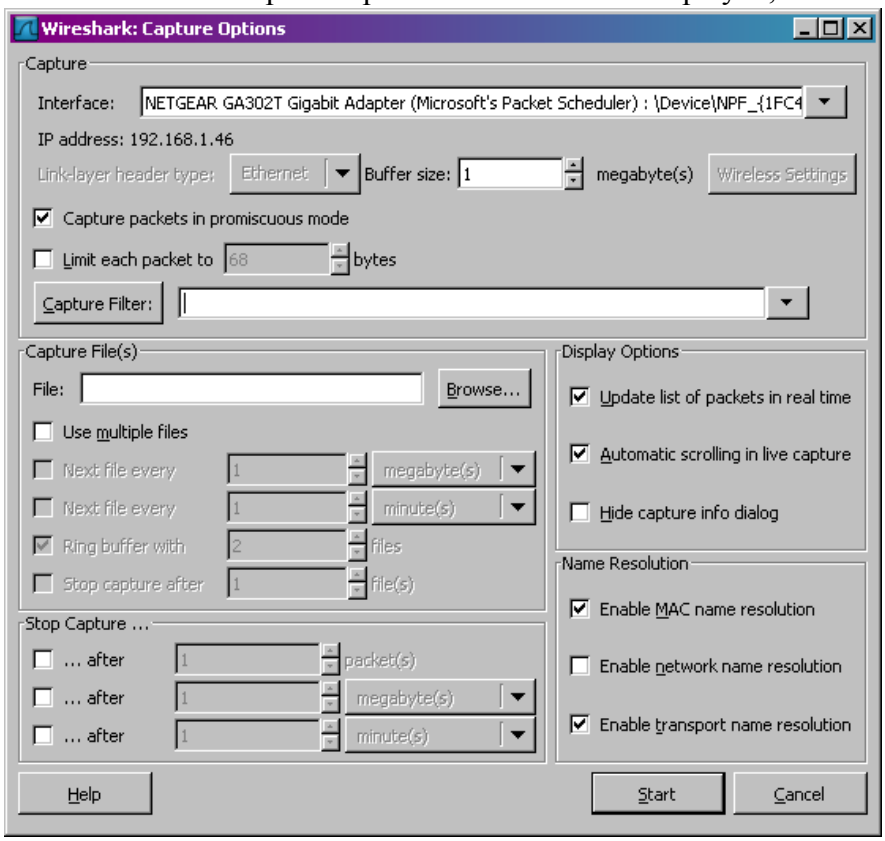
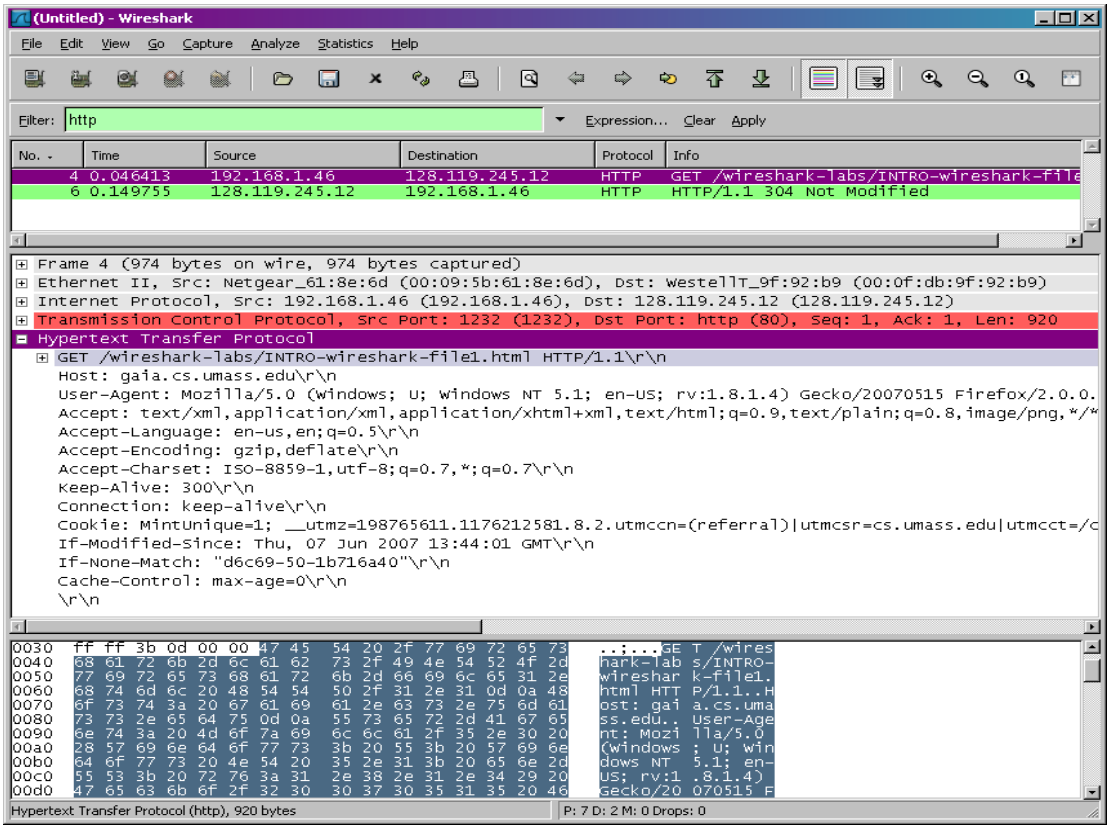


Figure 3: Wireshark Capture Options Window

4. You can use most of the default values in this window. Network interfaces (i.e., the physical connections) that your computer is connected to the network will be shown in the Interface pull down menu at the top of the Capture Options window. In case your computer has more than one active network interface (e.g., if you have both a wireless and a wired Ethernet connection), you will need to select an interface that is being used to send and receive packets (mostly likely the wired interface). After selecting the network interface (or using the default interface chosen by Wireshark), click Start. Packet capture will now begin - all packets being sent/received from/by your computer are now being captured by Wireshark!
5. Once you begin packet capture, a packet capture summary window will appear, as shown in Figure 4. This window summarizes the number of packets of several types that are being captured, and (importantly!) contains the *Stop* button that will allow you to stop packet capture. Don't stop packet capture yet.
6. While Wireshark is running, enter the URL:
http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html
and have that page displayed in your browser. To display this page, your browser will contact the HTTP server at gaia.cs.umass.edu and exchange HTTP messages with the server to download this page. The Ethernet frames containing these HTTP messages will be captured by Wireshark.

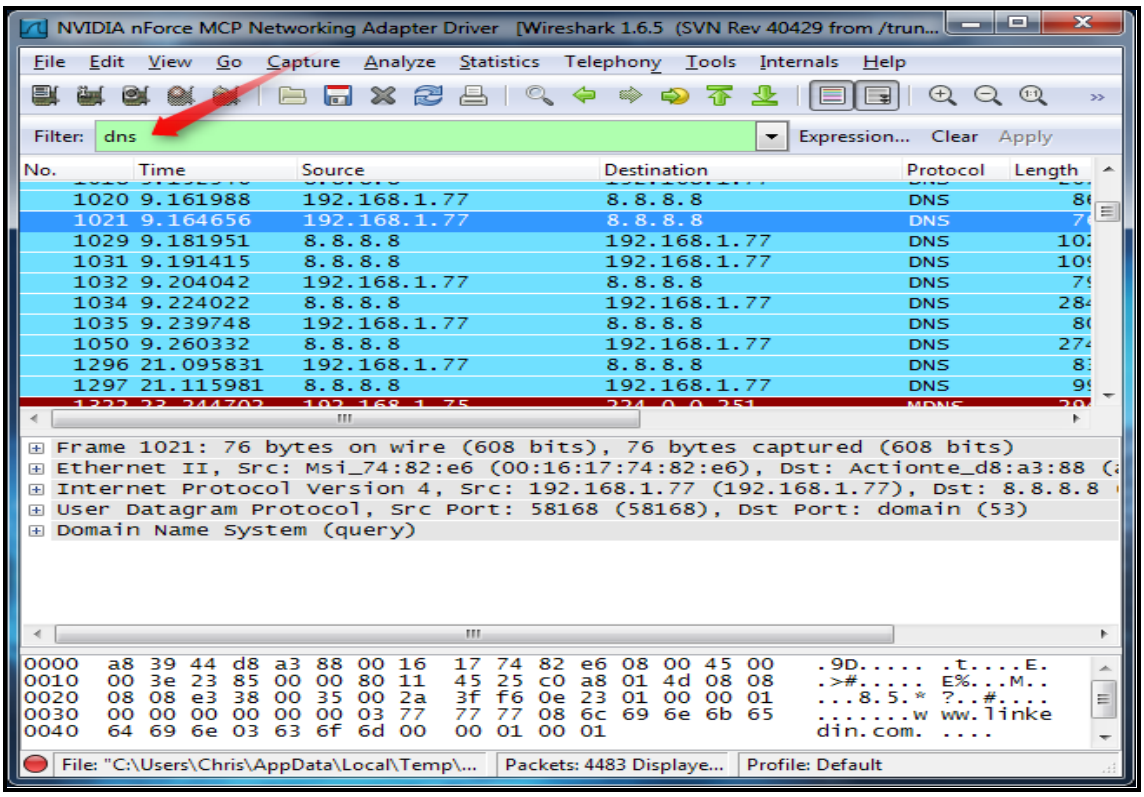
7. After your browser has displayed the INTRO-wireshark-file1.html page, stop Wireshark packet capture by selecting stop in the Wireshark capture window. This will cause the Wireshark capture window to disappear.



Filtering Packets

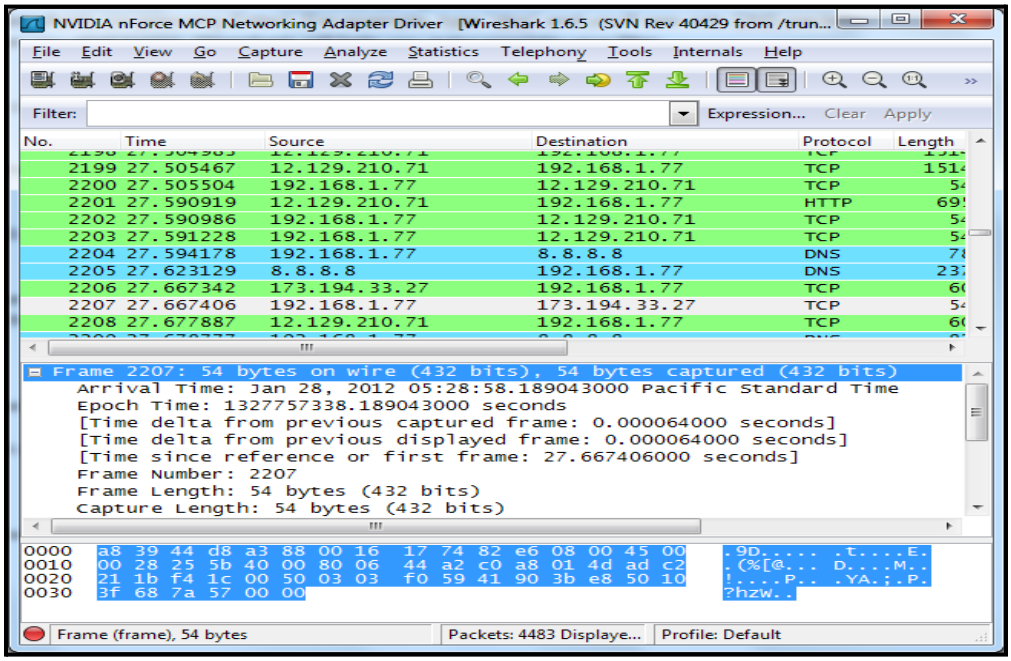
If you're trying to inspect something specific, such as the traffic a program sends when phoning home, it helps to close all other applications using the network so you can narrow down the traffic. Still, you'll likely have many packets to sift through. That's where Wireshark's filters come in.

The most basic way to apply a filter is by typing it into the filter box at the top of the window and clicking Apply (or pressing Enter). For example, type "dns" and you'll see only DNS packets. When you start typing, Wireshark will help you autocomplete your filter.



Inspecting Packets

Click a packet to select it and you can dig down to view its details:



QUESTIONS: -

1. What is the Internet address of the gaia.cs.umass.edu (also known as www-net.cs.umass.edu)? What is the Internet address of your computer?

2. Access your university website and list IP address of your computer, DNS

address and default gateway

Solution: -

Task 1:-

Pinging gaia.cs.umass.edu: -

```
C:\Users\musta\Desktop>ping gaia.cs.umass.edu

Pinging gaia.cs.umass.edu [128.119.245.12] with 32 bytes of data:
Reply from 128.119.245.12: bytes=32 time=289ms TTL=34
Reply from 128.119.245.12: bytes=32 time=326ms TTL=34
Reply from 128.119.245.12: bytes=32 time=323ms TTL=34
Reply from 128.119.245.12: bytes=32 time=278ms TTL=34

Ping statistics for 128.119.245.12:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 278ms, Maximum = 326ms, Average = 304ms
```

Ip address of my computer: -

Source
192.168.189.75

Ip address of gaia site: -

Destination
128.119.245.12

Nslookup output: -

```
C:\Users\musta\Desktop>nslookup gaia.cs.umass.edu
Server:  UnKnown
Address:  192.168.189.88

Non-authoritative answer:
Name:    gaia.cs.umass.edu
Address:  128.119.245.12
```

Task 2: -
Pinging to cms.bahria.edu.pk: -

```
C:\Users\musta\Desktop>ping cms.bahria.edu.pk

Pinging cms.bahria.edu.pk [111.68.99.12] with 32 bytes of data:
Reply from 111.68.99.12: bytes=32 time=64ms TTL=114
Reply from 111.68.99.12: bytes=32 time=85ms TTL=114
Reply from 111.68.99.12: bytes=32 time=119ms TTL=114
Reply from 111.68.99.12: bytes=32 time=74ms TTL=114

Ping statistics for 111.68.99.12:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 64ms, Maximum = 119ms, Average = 85ms
```

Ip address of pc: -

```
Source
192.168.189.75
```

DNS address of pc: -

```
DNS Servers . . . . . . . . . . . : 192.168.189.88
```

Default gateway of pc: -

```
Default Gateway . . . . . . . . . : 192.168.189.88
```

TIME BOXING:

Activity Name	**Activity Time**	**Total Time**
Instruments Allocation + Setting up Lab	10 mints	10 mints
Walk through Theory & Tasks (Lecture)	60 mints	60 mints
Implementation & Practice time	90 mints	80 mints
Evaluation Time	20 mints	20 mints
	Total Duration	180 mints

Teacher Signature: _____

Student Registration No: 69966_____