

# ***MOBILE COMPUTING***

## ***ETIT 452***

Faculty name: Mr. Yogesh Sharma

Student name: Chander Jindal

Roll No.:06514802719

Semester:8C4



Maharaja Agrasen Institute of Technology, PSP Area,

Sector – 22, Rohini, New Delhi – 110086

# *MAHARAJA AGRASEN INSTITUTE OF*



## *TECHNOLOGY*

### **Vision of the Institute**

To nurture young minds in a learning environment of high academic value and imbibe spiritual and ethical values with technological and management competence.

### **Mission of the Institute**

The Institute shall endeavor to incorporate the following basic missions in the teaching methodology:

### **Engineering Hardware – Software Symbiosis**

Practical exercises in all Engineering and Management disciplines shall be carried out by Hardware equipment as well as the related software enabling deeper understanding of basic concepts and encouraging inquisitive nature.

### **Life – Long Learning**

The Institute strives to match technological advancements and encourage students to keep updating their knowledge for enhancing their skills and inculcating their habit of continuous learning.

### **Liberalization and Globalization**

The Institute endeavors to enhance technical and management skills of students so that they are intellectually capable and competent professionals with Industrial Aptitude to face the challenges of globalization.

### **Diversification**

The Engineering, Technology and Management disciplines have diverse fields of studies with different attributes. The aim is to create a synergy of the above attributes by encouraging analytical thinking.

### **Digitization of Learning Processes**

The Institute provides seamless opportunities for innovative learning in all Engineering and Management disciplines through digitization of learning processes using analysis, synthesis, simulation, graphics, tutorials and related tools to create a platform for multi- disciplinary approach.

### **Entrepreneurship**

The Institute strives to develop potential Engineers and Managers by enhancing their skills and research capabilities so that they become successful entrepreneurs and responsible citizens.

# **Vision of the Department**

To Produce “Creators of Innovative Technology”

## **Mission of the Department:**

To provide an **excellent learning environment** across the computer science discipline to inculcate professional behavior, strong ethical values, innovative **research capabilities** and leadership abilities which enable them to become successful **entrepreneurs** in this globalized world.

1. To nurture an **excellent learning environment** that helps students to enhance their problem solving skills and to prepare students to be lifelong learners by offering a solid theoretical foundation with applied computing experiences and educating them about their **professional, and ethical responsibilities**.
2. To establish **Industry-Institute Interaction**, making students ready for the industrial environment and be successful in their professional lives.
3. To promote **research activities** in the emerging areas of technology convergence.
4. To build engineers who can look into technical aspects of an engineering solution thereby setting a ground for producing successful **entrepreneur**.

## **Program Educational Objective:**

**PEO1:** To train students to have successful careers in computer engineering field or to be able to successfully pursue advanced degrees.

**PEO2:** To imbibe in students an ability to provide solutions to challenging problems in their profession by applying computer Engineering principles.

**PEO3:** Train students to communicate effectively, work collaboratively and exhibit high levels of professionalism and ethical responsibility.

**PEO4:** To motivate graduates to engage in life-long learning and professional development to adapt to rapidly changing work environment

# **Introduction to Mobile Computing Lab**

## **Objective**

The objective of the Mobile Computing lab course is to introduce CSE students to the field of Mobile Apps development and cellular networks simulations using various open source tools. It would give them hands-on experience through a set of experiments using NS3, Android and XML.

## **Course outcomes**

- To provide guidelines, design principles and experience in developing applications for small, mobile devices, including an appreciation of context and location aware services
- To develop an appreciation of interaction modalities with small, mobile devices (including interface design for non-standard display surfaces) through the implementation of simple applications using XML, Android etc.
- To introduce wireless communication and networking principles, that support connectivity to cellular networks, wireless internet and other data networks.
- To understand the use of NS3 in simulating various types, topologies of Wireless networks
- To appreciate the social and ethical issues of mobile computing, including privacy.
- Gain an understanding of performance issues in advanced wireless and mobile networks.

# LIST OF EXPERIMENTS

*(as per GGSIPU Syllabus)*

- Write a WML program to print a formatted Text on the mobile Screen using various tags.
- Write a WML program to connect multiple cards from same desk.
- Write WML program to display table with three columns Image name, Image and third column contain hyperlink to open another card.
- Write a WML program to create a form with multiple options.
- Write a WML program to use the time control and to trigger On pick event.
- Write a WML script to find maximum out of two numbers with help of inbuilt function Lang.Max() and to find absolute value with help of inbuilt function Lang.abs().
- Write a Program in NS3 to Simulate OLSR
- Write a Program in NS3 to Simulate AODV
- Write a Android Program to create list view, grid view and database connectivity
- . Make an application of using XML or Android from the following: Game, Clock, calendar, Convertor, phone book, Text Editor.

# Introduction to WML

WML is a markup language designed especially for specifying and displaying content on WAP (Wireless Application Protocol) devices. WML is part of the WAP application environment, which requires the use of WML.

WML is the wireless equivalent of HTML for the Web. WML is based on XML and derived from xHTML (the XML version of HTML). There are many differences between WML and HTML. For example, WML has a different mechanism for linking between its pages called “cards” as compared to linking between HTML pages. WML browsers are stricter than HTML browsers by not being tolerant of errors. WML browsers enforce the WML requirement of matching closing “tags”, an XML characteristic.

WML works with the WAP micro browsers found on WAP devices. This browser is cognizant of the limited capabilities of WAP devices such as

- small displays,
- limited processing power,
- limited memory,
- narrow bandwidth connection,
- limited battery use without recharging

To address the limitations of WAP devices, WML uses the metaphor of card decks, and each page is referred to as a card. The card is the basic unit of navigation and user interface. The user can view only card at a time. WML browsers read the whole deck (complete document) from the server to minimize interaction with the server. Consequently, when flipping (navigating) between the cards in a deck, the browser does not contact the server. This eliminates delays (because each card contains very little text and users are likely to move quickly from one card to another).

A WAP deck is the equivalent of a Web page, the card being the portion of the Web page that can be seen on the screen. Navigation within the cards of a deck is done within the WAP device just as scrolling a Web page is done within the Web device. (without contacting servers in both cases).

An HTML writer does not worry about screen or display boundaries. Instead, the Web browser manages issues relating to the screen boundaries. But a WML writer must be aware of screen boundaries of WAP devices when writing code for cards.

Web server requests are routed through WAP gateways (proxies). A Web server may generate WML content for WAP devices or it may simply dish out HTML (XML).

In case the Web server generates HTML (XML), the WAP gateways must convert the HTML (XML) to WML. Before sending the WML to the WAP device, the gateway compresses it to WMLC (the C in WMLC is for compressed). If the WML generated by the Web sever is WMLC, then the compression step is skipped. Incidentally, converting HTML automatically to WML typically does not produce good or even usable results. For best results, Web servers should generate WML for WAP devices.

## Advantages of WML

- Like HTML, WML is easy to use. However, compared to HTML, WML has the following advantages in the context of wireless:
  - WML is part of the WAP standard and its use is required.
  - Transmission of WML (WMLC) documents requires less bandwidth compared to HTML documents because WML documents are simpler and WML is compressed before it is sent to the WAP device.
  - Compared to HTML documents, displaying WML documents requires less processing power and memory. Consequently, a WAP device can work with a less powerful (cheaper) CPU and the use of less power means that the battery can operate longer without recharging.
  - WML provides support for limited graphics with a limited gray scale.

## Disadvantages of WML:

- Like HTML, WML does specify how the content is to be displayed. Thus micro browsers on different WAP devices are likely to display the WML content differently.
- WAP devices such as WAP phones will not accept large decks (1.4K for some WAP phones).
- There are many variations between WAP phones, for example Screen sizes, keypads, and soft keys can be different. Consequently, WML decks should be tested on at least the important WAP devices. This variation is similar to the variation found with Web browsers and their platforms. The problem is harder in case of WML because there are many more WAP devices than Web browsers and their platforms. Also, it is harder to figure out the “least common denominator”, i.e., set of features that will work reasonably well on all or most WAP devices.

## Different Tags of WML are as follows:

### Deck & Card Elements

WML Elements	Purpose
<!-->	Defines a WML comment
<wml>	Defines a WML deck (WML root)
<head>	Defines head information
<meta>	Defines meta information
<card>	Defines a card in a deck
<access>	Defines information about the access control of a deck
<template>	Defines a code template for all the cards in a deck

## Text Elements

WML Elements	Purpose
 	Defines a line break
<p>	Defines a paragraph
<table>	Defines a table
<td>	Defines a table cell (table data)
<tr>	Defines a table row
<pre>	Defines preformatted text

## Text Formatting Tags

WML Elements	Purpose
<b>	Defines bold text
<big>	Defines big text
<em>	Defines emphasized text
<i>	Defines italic text
<small>	Defines small text
<strong>	Defines strong text
<u>	Defines underlined text

## Image Elements

WML Elements	Purpose
<img>	Defines an image

## Anchor Elements

WML Elements	Purpose
<a>	Defines an anchor
<anchor>	Defines an anchor



## Event Elements

WML Elements	Purpose
<b>&lt;do&gt;</b>	Defines a do event handler
<b>&lt;onevent&gt;</b>	Defines an onevent event handler
<b>&lt;postfield&gt;</b>	Defines a postfield event handler
<b>&lt;ontimer&gt;</b>	Defines an ontimer event handler
<b>&lt;onenterforward&gt;</b>	Defines an onenterforward handler
<b>&lt;onenterbackward&gt;</b>	Defines an onenterbackward handler
<b>&lt;onpick&gt;</b>	Defines an onpick event handler

## Task Elements

WML Elements	Purpose
<b>&lt;go&gt;</b>	Represents the action of switching to a new card
<b>&lt;noop&gt;</b>	Says that nothing should be done
<b>&lt;prev&gt;</b>	Represents the action of going back to the previous card
<b>&lt;refresh&gt;</b>	Refreshes some specified card variables.

## Variable Elements

WML Elements	Purpose
<b>&lt;setvar&gt;</b>	Defines and sets a variable
<b>&lt;timer&gt;</b>	Defines a timer

## Input Elements

WML Elements	Purpose
--------------	---------

<b>&lt;input&gt;</b>	Defines an input field
<b>&lt;select&gt;</b>	Defines a select group
<b>&lt;option&gt;</b>	Defines an option in a selectable list
<b>&lt;fieldset&gt;</b>	Defines a set of input fields
<b>&lt;optgroup&gt;</b>	Defines an option group in a selectable list

# Introduction to NS3

*Some of basic steps involved in making programs in NS3 are list below:*

## **Simple client-server communication**

1. Create a simple topology of two nodes (Node1, Node2) separated by a point-to-point link.
2. Setup a UdpClient on one Node1 and a UdpServer on Node2. Let it be of a fixed data rate Rate1.
3. Start the client application, and measure end to end throughput whilst varying the latency of the link.
4. Now add another client application to Node1 and a server instance to Node2. What do you need to configure to ensure that there is no conflict?
5. Repeat step 3 with the extra client and server application instances. Show screenshots of pcap traces which indicate that delivery is made to the appropriate server instance.

## **TCP variants**

1. Create a simple dumbbell topology, two client Node1 and Node2 on the left side of the dumbbell and server nodes Node3 and Node4 on the right side of the dumbbell. Let Node5 and Node6 form the bridge of the dumbbell. Use point to point links.
2. Install a TCP socket instance on Node1 that will connect to Node3.
3. Install a UDP socket instance on Node2 that will connect to Node4.
4. Start the TCP application at time 1s.
5. Start the UDP application at time 20s at rate Rate1 such that it clogs half the dumbbell bridge's link capacity.
6. Increase the UDP application's rate at time 30s to rate Rate2 such that it clogs the whole of the dumbbell bridge's capacity.
7. Use the ns-3 tracing mechanism to record changes in congestion window size of the TCP instance over time. Use gnuplot/matplotlib to visualise plots of cwnd vs time.
8. Mark points of fast recovery and slow start in the graphs.
9. Perform the above experiment for TCP variants Tahoe, Reno and New Reno, all of which are available with ns-3.

## **TCP and router queues**

1. Create a simple dumbbell topology, two client Node1 and Node2 on the left side of the dumbbell and server nodes Node3 and Node4 on the right side of the dumbbell. Let Node5 and Node6 form the bridge of the dumbbell. Use point to point links.
2. Add drop tail queues of size QueueSize5 and QueueSize6 to Node5 and Node6, respectively.
3. Install a TCP socket instance on Node1 that will connect to Node3.
4. Install a TCP socket instance on Node2 that will connect to Node3.
5. Install a TCP socket instance on Node2 that will connect to Node4.
6. Start Node1--Node3 flow at time 1s, then measure it's throughput. How long does it take to fill link's entire capacity?
7. Start Node2--Node3 and Node2--Node4 flows at time 15s, measure their throughput.
8. Measure packet loss and cwnd size, and plot graphs throughput/time, cwnd/time and packet loss/time for each of the flows.
9. Plot graph throughput/cwnd and packet loss/cwnd for the first flow. Is there an optimal value for cwnd?
10. Vary QueueSize5 and QueueSize6.

## **Routing (Optimised Link State Routing)**

1. Create a wireless mobile ad-hoc network with three nodes Node1, Node2 and Node3. Install the OLSR routing protocol on these nodes.
2. Place them such that Node1 and Node3 are just out of reach of each other.
3. Create a UDP client on Node1 and the corresponding server on Node3.
4. Schedule Node1 to begin sending packets to Node3 at time 1s.
5. Verify whether Node1 is able to send packets to Node3.
6. Make Node2 move between Node1 and Node3 such that Node2 is visible to both A and C. This should happen at time 20s. Ensure that Node2 stays in that position for another 15s.
7. Verify whether Node1 is able to send packets to Node3.
8. At time 35s, move Node2 out of the region between Node1 and Node3 such that it is out of each other's transmission ranges again.
9. Verify whether Node1 is able to send packets to Node3.
10. To verify whether data transmissions occur in the above scenarios, use either the tracing mechanism or a RecvCallback() for Node3's socket.
11. Plot the number of bytes received versus time at Node3.
12. Show the pcap traces at Node 2's Wifi interface, and indicate the correlation between Node2's packet reception timeline and Node2's mobility.

## Wifi RTS/CTS

1. Setup a 5x5 wireless adhoc network with a grid. You may use [examples/wireless/wifi-simple-adhoc-grid.cc](https://examples.wireless/simple-adhoc-grid.cc) as a base.
2. Install the OLSR routing protocol.
3. Setup three UDP traffic flows, one along each diagonal and one along the middle (at high rates of transmission).
4. Setup the ns-3 flow monitor for each of these flows.
5. Now schedule each of the flows at times 1s, 1.5s, and 2s.
6. Now using the flow monitor, observe the throughput of each of the UDP flows. Furthermore, use the tracing mechanism to monitor the number of packet collisions/drops at intermediary nodes. Around which nodes are most of the collisions/drops happening?
7. Now repeat the experiment with RTS/CTS enabled on the wifi devices.
8. Show the difference in throughput and packet drops if any.

## Wifi Channels

1. Setup a 2-nodes wireless adhoc network. Place the nodes at a fixed distance in a 3d scenario.
2. Install all the relevant network stacks, up to and including UDP.
3. Setup a CBR transmission between the nodes, one acting as a server and one as a client. Take the `iperf[1]` behaviour as an example.
4. Setup counters and outputs for packets sent and received.
5. Schedule the simulation to run for enough time to obtain statistically relevant results (suggestion: analyze some test results and reduce the simulation time accordingly).
6. Repeat the simulation varying the distance between the nodes from a minimum of 1meter to the point where the nodes can't transmit/receive anymore.
7. Repeat the above varying the channel models and the transmission/receive parameters like node's position above the ground, transmission power, etc.
8. Show the differences between the various channel models, and comment them. Identify the channel model that is more appropriate for each case (indoor, outdoor, LoS, NLoS, etc.).

# Index

S. No.	Experiment Name	Date of performance	Date of Checking	Marks	Signature
1	Write a WML program to print a formatted Text on the mobile Screen using various tags.				
2	Write a WML program to connect multiple cards from same desk.				
3	Write WML program to display table with three columns Image name, Image and third column contain hyperlink to open another card.				
4	Write a WML program to create a form with multiple options.				
5	Write a WML program to use the time control and to trigger On pick event.				
6	Write a WML script to find maximum out of two numbers with help of inbuilt function Lang.Max() and to find absolute value with help of inbuilt function Lang.abs().				
7	Write a Program in NS3 to Simulate OLSR				
8	Write a Program in NS3 to Simulate AODV				

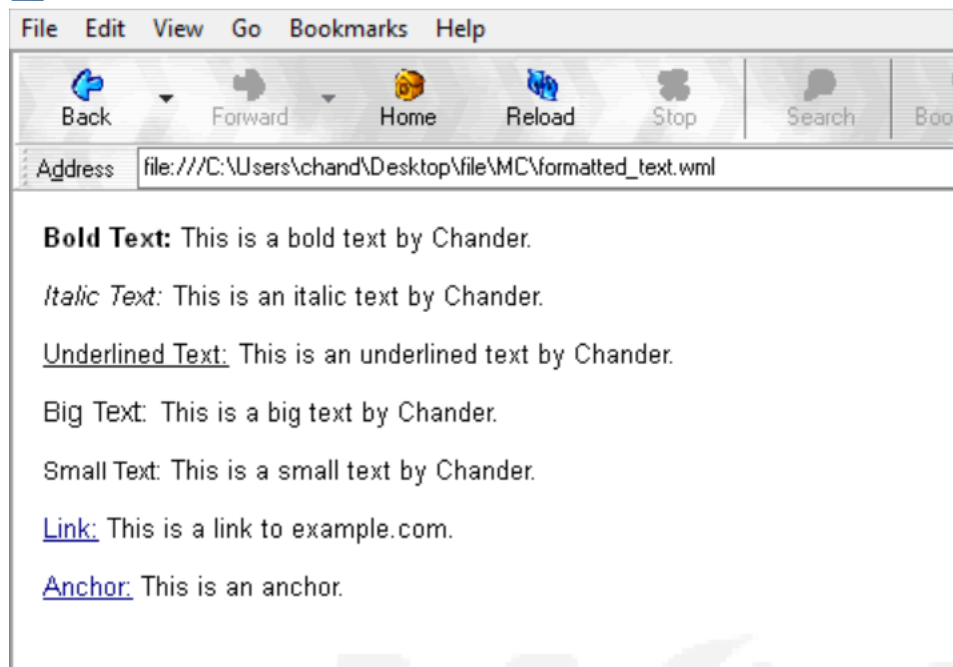
# Experiment 1

**AIM: Write a WML program to print a formatted Text on the mobile Screen using various tags.**

## *Code*

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">
<wml>
  <card id="card1" title="Formatted Text">
    <p>
      <b>Bold Text:</b> This is a bold text by Chander.
    </p>
    <p>
      <i>Italic Text:</i> This is an italic text by Chander.
    </p>
    <p>
      <u>Underlined Text:</u> This is an underlined text by Chander.
    </p>
    <p>
      <big>Big Text:</big> This is a big text by Chander.
    </p>
    <p>
      <small>Small Text:</small> This is a small text by Chander.
    </p>
    <p>
      <a href="http://www.example.com">Link:</a> This is a link to example.com.
    </p>
    <p>
      <anchor title="Anchor" href="anchor.wml">Anchor:</anchor> This is an anchor.
    </p>
  </card>
</wml>
```

**OUTPUT:**





## Viva Voice

### Q. What is the use of WML decks?

- ⇒ WML document that is used to create an application is called as deck and it provides way to add more pages. Deck provides the insertion of the data into one or more cards which is also called as pages. Deck interacts with the user and the framework on which the application is being built. Decks are stored on a configured web server that serves the purpose of including the MIME type of data.

### Q. What is WSDL?

- ⇒ WSDL is an XML notation for describing a web service. A WSDL definition tells a client how to compose a web service request and describes the interface that is provided by the web service provider. A WSDL definition is divided into separate sections that specify the logical interface and the physical details of a web service. The physical details include both endpoint information, such as HTTP port number, and binding information, which specifies how the SOAP payload is represented and which transport is used.

### Q. What Are The Wml Variable? How To Use Them?

- ⇒ Multiple cards can be contained within one deck, some mechanism needs to be in place to hold data as the user traverses from card to card. This mechanism is provided via WML variables.

WML is case sensitive. No case folding is performed when parsing a WML deck. All enumerated attribute values are case sensitive. For example, the following attribute values are all different: `id="Card1"`, `id="card1"`, and `id="CARD1"`. The `<setvar>` element is used as a result of the user executing some task. The `>setvar>` element can be used to set a variable's state within the following elements: `<go>`, `<prev>`, and `<refresh>`.

### Q. How Can We Refresh Card Variables?

- ⇒ The `refresh()` function, as suggested by its function name, is used to refresh the current card on the WML browser. It does not take any arguments:

```
WMLBrowser.refresh();
```

An empty string is returned if the function call succeeds. If any error occurs, a non-empty string is returned. What it contains depends on the WML browser you use. It should be a message explaining why the function call fails. If immediate refresh is not supported, *invalid* is returned and the refresh operation will be done after the WMLScript interpreter gives back control to the WML browser.

# Experiment 2

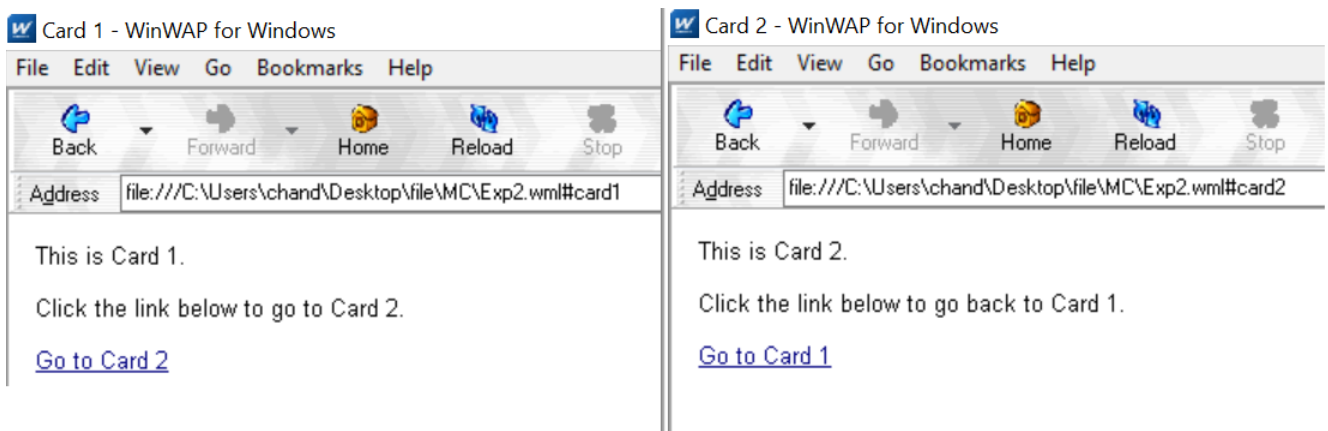
**AIM:** Write a WML program to connect multiple cards from same desk

## Code

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">
<wml>
  <card id="card1" title="Card 1">
    <p>This is Card 1.</p>
    <p>Click the link below to go to Card 2.</p>
    <p>
      <a href="#card2">Go to Card 2</a>
    </p>
  </card>

  <card id="card2" title="Card 2">
    <p>This is Card 2.</p>
    <p>Click the link below to go back to Card 1.</p>
    <p>
      <a href="#card1">Go to Card 1</a>
    </p>
  </card>
</wml>
```

## Output



## Viva Voice

### Q. What is the use of XML?

⇒ **Web publishing:** XML allows you to create interactive pages, allows the customer to customize those pages, and makes creating e-commerce applications more intuitive.

**Web searching and automating Web tasks:** XML defines the type of information contained in a document, making it easier to return useful results when searching the Web.

**General applications:** XML provides a standard method to access information, making it easier for applications and devices of all kinds to use, store, transmit, and display data.

**E-business applications:** XML implementations make electronic data interchange (EDI) more accessible for information interchange, business-to-business transactions, and business-to-consumer transactions.

**Metadata applications:** XML makes it easier to express metadata in a portable, reusable format.

### Q. What is Meta Data?

⇒ Sometimes you have the need to set up some app-wide configuration information in an Android app or need to create a class that can be used in multiple projects with a generic way of setting configuration values. This is particularly useful for things like API keys that will probably be different across apps but should be accessible in the same way. There are several ways to do it, but the simplest one is to add a meta-data node to the AndroidManifest.xml file.

### Q. What is the difference between HTML and WML?

⇒ A main difference between HTML and WML is that the basic unit of navigation in HTML is a page, while that in WML is a card.

A WML file can contain multiple cards and they form a deck.

When a WML page is accessed from a mobile phone, all the cards in the page are downloaded from the WAP server.

### Q. What is XML DOM Document?

⇒ DOM is an acronym stands for Document Object Model. It defines a standard way to access and manipulate documents. The Document Object Model (DOM) is a programming API for HTML and XML documents. It defines the logical structure of documents and the way a document is accessed and manipulated

# Experiment 3

**AIM: Write WML program to display table with three columns Image name, Image and third column contain hyperlink to open another card**

## Code

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">

<wml>
  <card id="card1" title="Image Gallery">
    <p>
      <table>
        <tr>
          <th>Image name</th>
          <th>Image</th>
          <th>Link</th>
        </tr>
        <tr>
          <td>Image 1</td>
          <td></td>
          <td><a href="#card2">Open Card 2</a></td>
        </tr>
        <tr>
          <td>Image 2</td>
          <td></td>
          <td><a href="#card3">Open Card 3</a></td>
        </tr>
      </table>
    </p>
  </card>

  <card id="card2" title="Card 2">
    <p>This is Card 2.</p>
    <p>Click the link below to go back to the Image Gallery.</p>
    <p>
      <a href="#card1">Go back to Image Gallery</a>
    </p>
  </card>

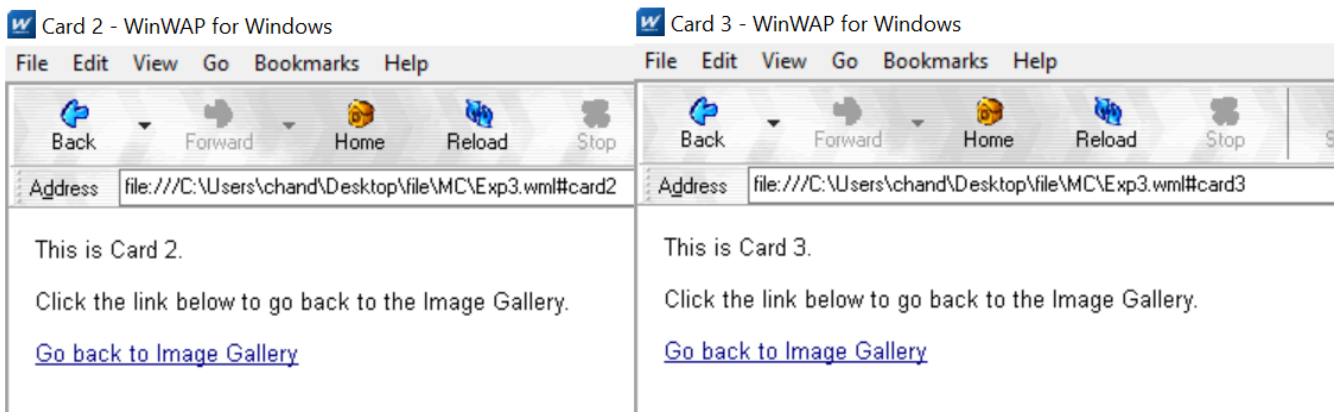
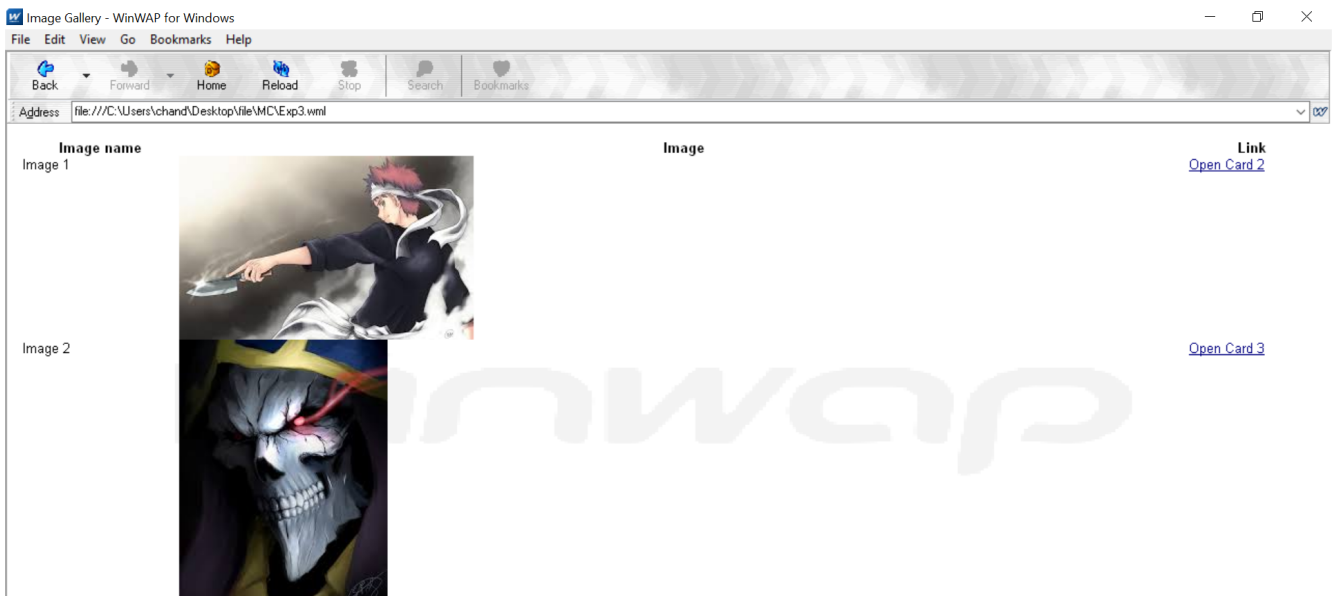
  <card id="card3" title="Card 3">
```

```

<p>This is Card 3.</p>
<p>Click the link below to go back to the Image Gallery.</p>
<p>
  <a href="#card1">Go back to Image Gallery</a>
</p>
</card>
</wml>

```

## OUTPUT:



## **Viva Questions**

### **Q. What is the function of WAP Gateway?**

⇒ A WAP gateway sits between mobile devices using the WAP protocol and the World Wide Web, passing pages from one to the other much like a proxy. This translates pages into a form suitable for the mobiles, for instance using the Wireless Markup Language (WML).

### **Q. What is Distillation technique in WAP?**

⇒ Distillation techniques convert the information from the rest of the internet into a suitable form for the WAP enabled devices. Web warden forwards all cellophane requests to a remote distillation server. Distillation server connected to rest of web and can fetch HTML pages, images

### **Q. What is the use of UAProf?**

⇒ UAProf (User agent profile) is an XML document that contains information about the user agent type and device capabilities. It is a standard defined and maintained by the Open Mobile Alliance (formerly the WAP Forum).

### **Q. Why WML is called Light weight Language?**

⇒ WAP - WML Script. WMLScript (Wireless Markup Language Script) is the client-side scripting language of WML (Wireless Markup Language). A scripting language is similar to a programming language, but is of lighter weight.

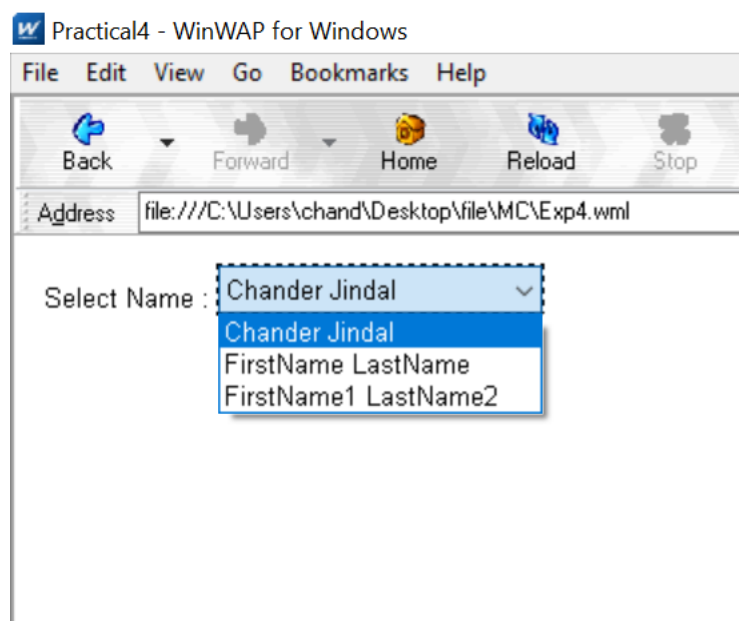
# Experiment 4

**Aim:** Write a WML program to create a form with multiple options.

**Code:**

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">
<wml>
<card title="Practical4">
<p> Select Name :
<select>
<option value="htm">Chander Jindal </option>
<option value="xml">FirstName LastName</option>
<option value="wap">FirstName1 LastName2</option>
</select>
</p>
</card>
</wml>
```

## Output



## **Viva questions**

### **Q. What is Push and Pull technique in WAP?**

⇒ In the normal client/server model, a client requests information or a service from a server. The server responds by transmitting information or performing a service to the client. This is known as pull technology—the client pulls information from the server.

In contrast to this, there is also push technology. The WAP push framework transmits information to a device without a previous user action. This technology is also based on the client/server model, but there is no explicit request from the client before the server transmits its content.

### **Q. List out the Databases used to store Data of WML pages?**

⇒ MS-Access  
Oracle 11g

### **Q. What is the advantages of using XML DOM document?**

⇒ XML structure is traversable, and it can be randomly accessed by traversing the tree.  
XML structure is modifiable, and values can be added, changed and removed

### **Q. What is DTD?**

⇒ A DTD is a Document Type Definition. A DTD defines the structure and the legal elements and attributes of an XML document. With a DTD, independent groups of people can agree on a standard DTD for interchanging data. An application can use a DTD to verify that XML data is valid.



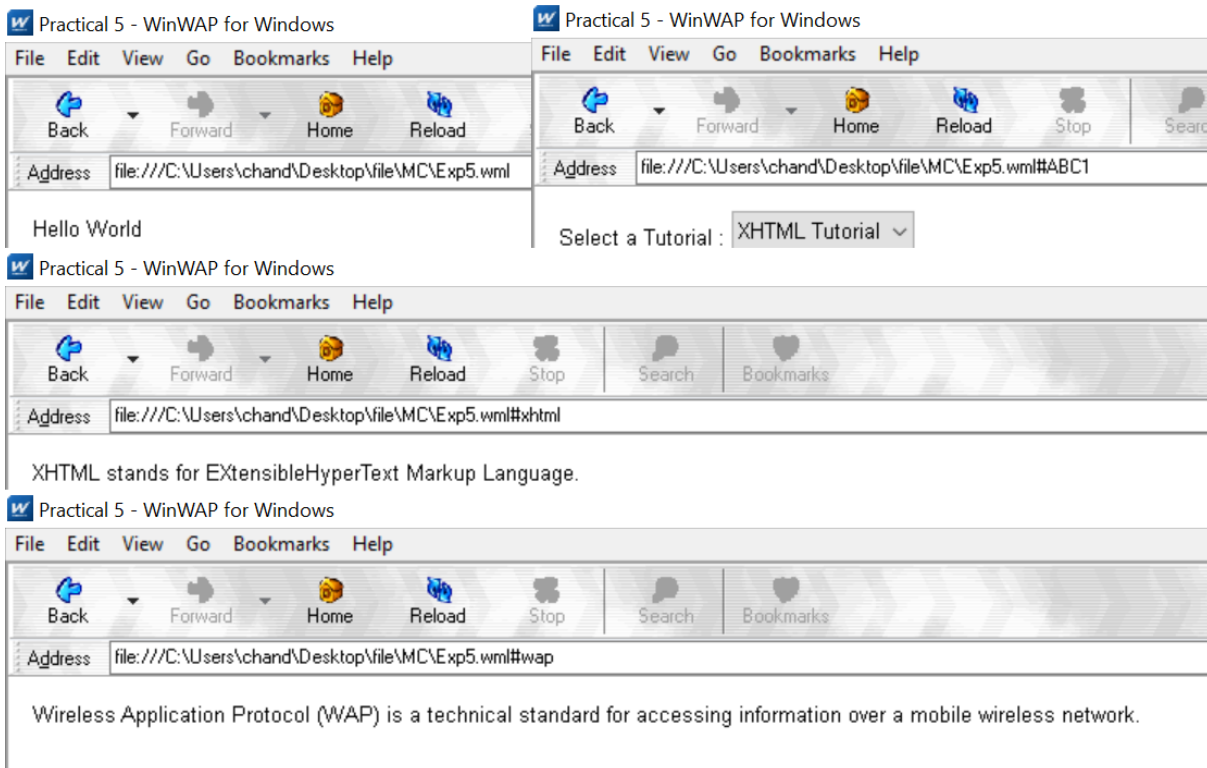
# Experiment 5

**AIM: Write a WML program to use the time control and to trigger on pick event.**

## Code

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.3//EN"
"http://www.wapforum.org/DTD/wml13.dtd">
<wml>
<card id="ABC" title="Practical 5">
<p>
Hello World
</p>
<onevent type="ontimer">
<go href="#ABC1"/>
</onevent>
<timer value="100"/>
</card>
<card id="ABC1" title="Practical 5">
<p>Select a Tutorial :
<select title="tutorials" name="selection_list">
<option onpick="#xhtml">XHTML Tutorial</option>
<option onpick="#wap">WAP Tutorial</option>
</select>
</p>
</card>
<card id="xhtml" title="Practical 5">
<p>
XHTML stands for EXtensibleHyperText Markup Language.
</p>
</card>
<card id="wap" title="Practical 5">
<p>
Wireless Application Protocol (WAP) is a technical standard for accessing information over a mobile wireless network.
</p>
</card>
</wml>
```

## OUTPUT:



**Viva Viva**

### Q. Does WAP run over GPRS?

⇒ GPRS is a method of connecting to your provider while WAP is the protocol that runs on top of GPRS. WAP is suited for GPRS only connections. There are also other services that use GPRS aside from WAP

### Q. Which Security is used in WAP?

⇒ Wireless Transport Layer Security (WTLS). WTLS is a wireless relative of the more common SSL mechanism used by all major web browsers. WTLS resembles SSL in that both rely on certificates on the client and server to verify the identity of the participants involved. While SSL implementations generally rely on RSA encryption, WTLS supports RSA, Diffie-Hellman, and Elliptic Curve encryption. WTLS also doesn't provide for end-to-end security due to WAP's current architecture and limitations of server-side Transport Layer Security (another name for SSL).

### Q. Is WML case sensitive?

⇒ Variable names in WML are case-sensitive. The first character of a variable name must be a letter or an underscore. The rest of the characters can be letters, numbers or underscores.

### Q. What does Post field tag do?

⇒ The <postfield> tag is used to post variables values to the server.

**The <postfield> tag contains information to be sent to the server along with a <go> tag.**

# Experiment 6

**AIM : Write a WML script to find maximum out of two numbers with help of inbuilt function Lang.Max() and to find absolute value with help of inbuilt function Lang.abs()**

## Code

```
//Exp6.wml file
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.3//EN"
"http://www.wapforum.org/DTD/wml13.dtd">
<wml>
<card id="card1">
<p>
<a href="exp6.wmls#findmax()">Find max</a>
<br/>
</p>
<p>
<a href="exp6.wmls#findabs()">Find abs</a>
<br/>
</p>
</card>
<card id="card2">
<p> Max of two numbers is:
$(maxnumber) </p>
</card>
<card id="card3">
<p> Absolute value is:
$(number) </p>
</card>
</wml>
```

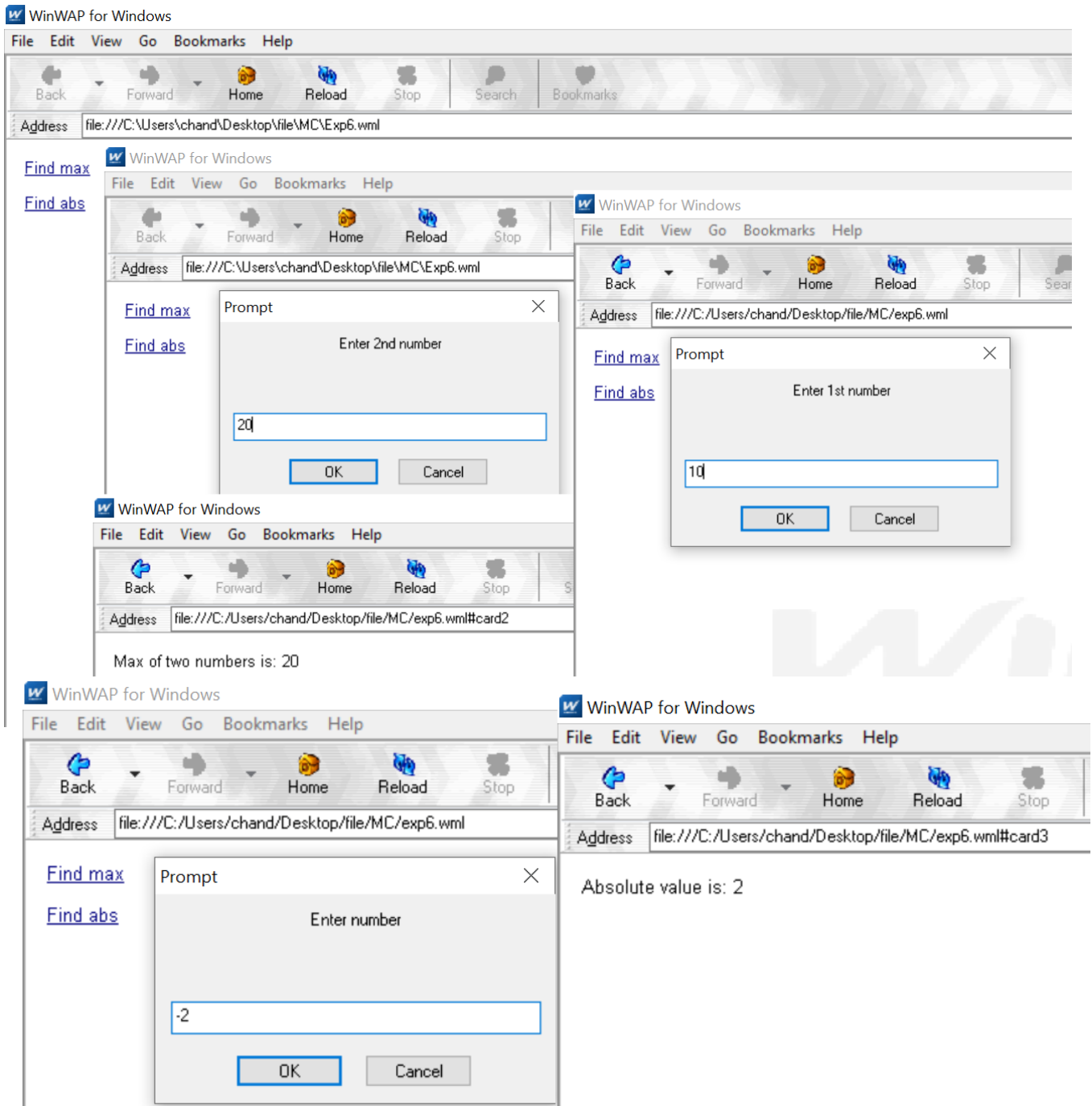
```
//Exp6.wmls file
// wmls
extern function findmax() {
var result1 = Dialogs.prompt("Enter 1st number",0);
var result2 = Dialogs.prompt("Enter 2nd number",0);
var maxnum = Lang.max(result1, result2);
WMLBrowser.setVar("number1", result1);
WMLBrowser.setVar("number2", result2);
WMLBrowser.setVar("maxnumber", maxnum);
```

```

WMLBrowser.go("exp6.wml#card2");
}
extern function findabs(){
var num = Dialogs.prompt("Enter number", 0);
var absmun = Lang.abs(num);
WMLBrowser.setVar("number", absmun);
WMLBrowser.go("exp6.wml#card3");
}

```

## Output



## Viva Questions

### Q.What is the function of WMLScript?

⇒ It is used for the client side and has many tasks that provide user input validation, generation of error messages, etc.

### Q. What is the use of WML in WMLScript?

⇒ WML provides many features to, represent the content that needs to, be displayed like, navigational support, data input, hyperlinks, etc. It has the provision to put the image and present it in variety of forms with the help of HTML. It uses other markup languages with the WAP to provide flexibility in the use of WML in WMLScript.

### Q. What is the use of WML decks?

⇒ Deck provides the insertion of the data into one or more cards which is also called as pages. Deck interacts with the user and the framework on which the application is being built.

### Q. What is the process where WML cards request the device to access WAP?

⇒ WML cards are just like pages on the Decks that are used to request the services on the device to access WAP. WAP gateway acts as a bridge between the mobile device and World Wide Web for the communication purpose. It provides the pages or cards from one system to another system using the proxy on the WWW.

### Q. What is the support of mobile devices for WMLScript?

⇒ Mobile devices are used to run and showcase the result of the input that is given and written by the use of WMLScript. WMLScript is written such that it provides hardware interfacing with the mobile using the WML.

### Q. What is the process of adding the client side logic to WAP using WMLScript?

⇒ WMLScript is similar to the JavaScript and it provides the same feature as JavaScript. WMLScript provides the client only scripting platform on the Internet that is used with the WML.

### Q. What is the purpose of using WMLScript?

⇒ WMLScript provides the validator that can be used to validate the user input given in the form. WMLScript provides the advanced functionality to write and read the code with the tools provided by it. It provides the facilities that can be accessed by the user agent and more application can run on the devices.

### Q. What are the data types used in WMLScript?

⇒ Boolean: this is the data type used for the values like true or false

Integer: this is a data type containing the numerical values

Floating-point: this is the data type used to provide the value in decimal points like 1.00, 1 e-10.

**String: stores the values in the form of characters in a contiguous memory location.**

# Experiment 7

***AIM: Write a Program in NS3 to Simulate OLSR***

## Code

```
#include <iostream>
#include <fstream>
#include <string>
#include <cassert>

#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/internet-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/applications-module.h"
#include "ns3/olsr-helper.h"
#include "ns3/ipv4-static-routing-helper.h"
#include "ns3/ipv4-list-routing-helper.h"

using namespace ns3;

NS_LOG_COMPONENT_DEFINE ("SimplePointToPointOlsrExample");

int
main (int argc, char *argv[])
{
    // Users may find it convenient to turn on explicit debugging
    // for selected modules; the below lines suggest how to do this
    #if 0
    LogComponentEnable ("SimpleGlobalRoutingExample", LOG_LEVEL_INFO);
    #endif

    // Set up some default values for the simulation. Use the

    Config::SetDefault ("ns3::OnOffApplication::PacketSize", UIntegerValue (210));
    Config::SetDefault ("ns3::OnOffApplication::DataRate", StringValue ("448kb/s"));

    //DefaultValue::Bind ("DropTailQueue::m_maxPackets", 30);

    // Allow the user to override any of the defaults and the above
    // DefaultValue::Bind ()s at run-time, via command-line arguments
    CommandLineCmd; cmd.Parse
    (argc, argv);

    // Here, we will explicitly create four nodes. In more sophisticated
    // topologies, we could configure a node factory.
    NS_LOG_INFO ("Create nodes."); NodeContainer
```

```

c;
c.Create (5);
NodeContainer n02 = NodeContainer (c.Get (0), c.Get (2));
NodeContainer n12 = NodeContainer (c.Get (1), c.Get (2));
NodeContainer n32 = NodeContainer (c.Get (3), c.Get (2));
NodeContainer n34 = NodeContainer (c.Get (3), c.Get (4));

// Enable OLSR
NS_LOG_INFO ("Enabling OLSR Routing.");
OlsrHelperolsr;

Ipv4StaticRoutingHelper staticRouting;

Ipv4ListRoutingHelperlist; list.Add
(staticRouting, 0);
list.Add (olsr, 10);

InternetStackHelper internet;
internet.SetRoutingHelper (list); // has effect on the next Install ()
internet.Install (c);

// We create the channels first without any IP addressing information
NS_LOG_INFO ("Create channels.");
PointToPointHelper p2p;
p2p.SetDeviceAttribute ("DataRate", StringValue ("5Mbps"));
p2p.SetChannelAttribute ("Delay", StringValue ("2ms"));
NetDeviceContainer nd02 = p2p.Install (n02); NetDeviceContainer
nd12 = p2p.Install (n12); p2p.SetDeviceAttribute ("DataRate",
StringValue ("1500kbps")); p2p.SetChannelAttribute ("Delay",
StringValue ("10ms")); NetDeviceContainer nd32 = p2p.Install
(n32); NetDeviceContainer nd34 = p2p.Install (n34);

// Later, we add IP addresses. NS_LOG_INFO
("Assign IP Addresses."); Ipv4AddressHelper
ipv4;
ipv4.SetBase ("10.1.1.0", "255.255.255.0");
Ipv4InterfaceContainer i02 = ipv4.Assign (nd02);

ipv4.SetBase ("10.1.2.0", "255.255.255.0");
Ipv4InterfaceContainer i12 = ipv4.Assign (nd12);

ipv4.SetBase ("10.1.3.0", "255.255.255.0");
Ipv4InterfaceContainer i32 = ipv4.Assign (nd32);

ipv4.SetBase ("10.1.4.0", "255.255.255.0");
Ipv4InterfaceContainer i34 = ipv4.Assign (nd34);

// Create the OnOff application to send UDP datagrams of size
// 210 bytes at a rate of 448 Kb/s from n0 to n4
NS_LOG_INFO ("Create Applications."); uint16_t

```

```
port = 9; // Discard port (RFC 863)
```

```
OnOffHelper onoff ("ns3::UdpSocketFactory",  
InetSocketAddress (i34.GetAddress (1), port));  
onoff.SetConstantRate (DataRate ("448kb/s"));
```

```
ApplicationContainer apps = onoff.Install (c.Get (0)); apps.Start  
(Seconds (1.0));  
apps.Stop (Seconds (10.0));
```

```
// Create a packet sink to receive these packets  
PacketSinkHelpersink ("ns3::UdpSocketFactory",  
InetSocketAddress (Ipv4Address::GetAny (), port));
```

```
apps = sink.Install (c.Get (3));  
apps.Start (Seconds (1.0));  
apps.Stop (Seconds (10.0));
```

```
// Create a similar flow from n3 to n1, starting at time 1.1  
seconds onoff.SetAttribute ("Remote",  
AddressValue (InetSocketAddress (i12.GetAddress (0),  
port))); apps = onoff.Install (c.Get (3));  
apps.Start (Seconds (1.1));  
apps.Stop (Seconds (10.0));
```

```
// Create a packet sink to receive these  
packets apps = sink.Install (c.Get (1));  
apps.Start (Seconds (1.1));  
apps.Stop (Seconds (10.0));
```

```
AsciiTraceHelper ascii;  
p2p.EnableAsciiAll (ascii.CreateFileStream ("simple-point-to-point-olsr.tr"));  
p2p.EnablePcapAll ("simple-point-to-point-olsr");
```

```
Simulator::Stop (Seconds (30));
```

```
NS_LOG_INFO ("Run  
Simulation."); Simulator::Run ();  
Simulator::Destroy ();  
NS_LOG_INFO ("Done.");
```

```
return 0;  
}
```



## Output:

```
File Edit View Search Terminal Help
99.9616 Received one packet!
99.9628 Received one packet!
99.9714 Received one packet!
99.9733 Received one packet!
99.9747 Received one packet!
99.9782 Received one packet!
99.9816 Received one packet!
99.9871 Received one packet!
99.9889 Received one packet!
99.9955 Received one packet!
99.9968 Received one packet!
[mait@CSE-114B-6 ns-3.26]$ ./waf --run OSLR-p2p
Waf: Entering directory `/home/mait/Downloads/ns-allinone-3.26/ns-3.26/build'
Waf: Leaving directory `/home/mait/Downloads/ns-allinone-3.26/ns-3.26/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (2.161s)
[mait@CSE-114B-6 ns-3.26]$ ./waf --run OSLR-p2p
Waf: Entering directory `/home/mait/Downloads/ns-allinone-3.26/ns-3.26/build'
[2235/2614] Compiling scratch/OSLR-p2p.cc
[2603/2614] Linking build/scratch/OSLR-p2p
Waf: Leaving directory `/home/mait/Downloads/ns-allinone-3.26/ns-3.26/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (4.694s)
[mait@CSE-114B-6 ns-3.26]$
```

## Viva Questions

### Q. List of Security Issues in Adhoc Networks?

Sybil  
Dos  
Negative Reply Threat.  
False Address Conflict Threat.  
Address Conflict Threat.  
Address Space Exhaustion Threat.  
Address Spoofing Threat.

### Q. What is Multi Casting?

Multicast is a communication system between a single sender and multiple receivers on a network. One address bit is reserved for multicasting and the remaining (n-1) address bits can hold a group number. Any machine can subscribe to any or all of the groups.

### Q. What is MANET?

A MANET is a type of ad hoc network that can change locations and configure itself on the fly. Because MANETS are mobile, they use wireless connections to connect to various networks. This can be a standard Wi-Fi connection, or another medium, such as a cellular or satellite transmission.

Some MANETs are restricted to a local area of wireless devices (such as a group of laptop computers), while others may be connected to the Internet.

### Q. What are the Characteristics of MANETs?

**Distributed Operation:** There is no background network for the central control of the network operations.

**Multi Hop Routing:** When a node tries to send information to other nodes which is out of its communication range, the packet should be forwarded via one or more intermediate nodes.

**Autonomous Terminal:** In MANET, each mobile node is an independent node, which could function as both a host and a router.

**Dynamic Topology:** Nodes are free to move arbitrarily with different speeds; thus, the network topology may change randomly and at unpredictable time. The nodes in the MANET dynamically establish routing among themselves as they travel around, establishing their own network.

**Light-Weight Terminals:** In maximum cases, the nodes at MANET are mobile with less CPU capability, low power storage and small memory size.

# Experiment 8

**AIM: Write a Program in NS3 to Simulate AODV**

## Code

```
#include <iostream>
#include <cmath>
#include "ns3/aodv-module.h"
#include "ns3/core-module.h"
#include "ns3/network-module.h"
#include "ns3/internet-module.h"
#include "ns3/mobility-module.h"
#include "ns3/point-to-point-module.h"
#include "ns3/v4ping-helper.h"
#include "ns3/yans-wifi-helper.h"
using namespace ns3;
class AodvExample {
public:
AodvExample ();
bool Configure (int argc, char **argv);
void Run ();
void Report (std::ostream & os);
private:
uint32_t size;
double step;
double totalTime;
bool pcap;
bool printRoutes;
NodeContainer nodes;
NetDeviceContainer devices;
Ipv4InterfaceContainer interfaces;
private:
void CreateNodes ();
void CreateDevices ();
void InstallInternetStack ();
void InstallApplications ();};
int main (int argc, char **argv){
AodvExample test;
if (!test.Configure (argc, argv))
NS_FATAL_ERROR ("Configuration failed. Aborted.");
test.Run ();
test.Report (std::cout);
return 0; }
AodvExample::AodvExample ():
size (10),
step (50),
```

```

totalTime (100),
pcap (true),
printRoutes (true) {}
bool AodvExample::Configure (int argc, char **argv) {
SeedManager::SetSeed (12345);
CommandLine cmd (__FILE__);
cmd.AddValue ("pcap", "Write PCAP traces.", pcap);
cmd.AddValue ("printRoutes", "Print routing table dumps.", printRoutes);
cmd.AddValue ("size", "Number of nodes.", size);
cmd.AddValue ("time", "Simulation time, s.", totalTime);
cmd.AddValue ("step", "Grid step, m", step);
cmd.Parse (argc, argv);
return true; }
void AodvExample::Run () {
CreateNodes ();
CreateDevices ();
InstallInternetStack ();
InstallApplications ();
std::cout << "Starting simulation for " << totalTime << " s ...\n";
Simulator::Stop (Seconds (totalTime));
Simulator::Run ();
Simulator::Destroy (); }
void AodvExample::Report (std::ostream &){}
void AodvExample::CreateNodes (){
std::cout << "Creating " << (unsigned)size << " nodes " << step << " m apart.\n";
nodes.Create (size);
for (uint32_t i = 0; i < size; ++i){
std::ostringstream os;
os << "node-" << i;
Names::Add (os.str (), nodes.Get (i)); }
MobilityHelper mobility;
mobility.SetPositionAllocator ("ns3::GridPositionAllocator", "MinX", DoubleValue (0.0),
"MinY", DoubleValue (0.0), "DeltaX", DoubleValue (step), "DeltaY", DoubleValue (0),
"GridWidth", UIntegerValue (size), "LayoutType", StringValue ("RowFirst"));
mobility.SetMobilityModel ("ns3::ConstantPositionMobilityModel");
mobility.Install (nodes); }
void AodvExample::CreateDevices (){
WifiMacHelper wifiMac;
wifiMac.SetType ("ns3::AdhocWifiMac");
YansWifiPhyHelper wifiPhy;
YansWifiChannelHelper wifiChannel = YansWifiChannelHelper::Default ();
wifiPhy.SetChannel (wifiChannel.Create ());
WifiHelper wifi;
wifi.SetRemoteStationManager ("ns3::ConstantRateWifiManager", "DataMode",
StringValue("OfdmRate6Mbps"), "RtsCtsThreshold", UIntegerValue (0));
devices = wifi.Install (wifiPhy, wifiMac, nodes);
if (pcap){
wifiPhy.EnablePcapAll (std::string ("aodv")); }
void AodvExample::InstallInternetStack (){
AodvHelper aodv;

```

```

InternetStackHelper stack;
stack.SetRoutingHelper (aadv); // has effect on the next Install ()
stack.Install (nodes);
Ipv4AddressHelper address;
address.SetBase ("10.0.0.0", "255.0.0.0");
interfaces = address.Assign (devices);
if (printRoutes){
Ptr<OutputStreamWrapper> routingStream = Create<OutputStreamWrapper> ("aadv.routes",
    std::ios::out);
aadv.PrintRoutingTableAllAt (Seconds (8), routingStream);}}
void AadvExample::InstallApplications (){
V4PingHelper ping (interfaces.GetAddress (size - 1));
ping.SetAttribute ("Verbose", BooleanValue (true));
ApplicationContainer p = ping.Install (nodes.Get (0));
p.Start (Seconds (0));
p.Stop (Seconds (totalTime) - Seconds (0.001));
// move node away
Ptr<Node> node = nodes.Get (size/2);
Ptr<MobilityModel> mob = node->GetObject<MobilityModel> ();
Simulator::Schedule (Seconds (totalTime/3), &MobilityModel::SetPosition, mob, Vector (1e5,
    1e5, 1e5));
}

```

### Output

```

2262/2614] Compiling scratch/scratch-simulator.cc
2530/2614] Linking build/scratch/second
2531/2614] Linking build/scratch/first
2532/2614] Linking build/scratch/star
2568/2614] Linking build/scratch/OSLR-p2p
2599/2614] Linking build/scratch/static-routing-slash32
2600/2614] Linking build/scratch/subdir/subdir
2601/2614] Linking build/scratch/AADV
2602/2614] Linking build/scratch/scratch-simulator
2603/2614] Linking build/scratch/DSOV
laf: Leaving directory /home/maif/Downloads/ns-allinone-3.26/ns-3.26/build
Build commands will be stored in build/compile_commands.json
'build' finished successfully (16.495s)
Creating 10 nodes 100 m apart.
Starting simulation for 10 s ...
PING 10.0.0.10 56(84) bytes of data.
64 bytes from 10.0.0.10: icmp_seq=0 ttl=56 time=2057 ms
64 bytes from 10.0.0.10: icmp_seq=1 ttl=56 time=1059 ms
64 bytes from 10.0.0.10: icmp_seq=2 ttl=56 time=60 ms
64 bytes from 10.0.0.10: icmp_seq=3 ttl=56 time=8 ms
-- 10.0.0.10 ping statistics --
10 packets transmitted, 4 received, 60% packet loss, time 9999ms
rtt min/avg/max/mdev = 8/796/2057/969.9 ms
maif@CSE-114B-6 ns-3.26$

```

## Viva Question

### Q. How routing in Adhoc networks different from fixed networks?

Nodes in the ad hoc networks are constantly moving therefore, the same principles in fixed networks cannot be applied in wireless ad hoc network. In fixed networks like the internet, the IP addresses are used as identification but since the nodes in the ad hoc networks are moving, they don't use the concept of IP addresses.

### Q. What is hidden and exposed terminal problem in Adhoc Networks?

In a formal way hidden terminals are nodes in a wireless network that are out of range of other nodes or a collection of nodes. Consider a wireless networking, each node at the far edge of the access point's range, which is known as A, can see the access point, but it is

unlikely that the same node can see a node on the opposite end of the access point's range, C. These nodes are known as hidden. The problem is when nodes A and C start to send packets simultaneously to the access point B. Because the nodes A and C are out of range of each other and so cannot detect a collision while transmitting, Carrier sense multiple access with collision detection (CSMA/CD) does not work, and collisions occur, which then corrupt the data received by the access point.

In wireless networks, when a node is prevented from sending packets to other nodes because of a neighboring transmitter is known as the exposed node problem. Consider the below wireless network having four nodes labeled A, B, C, and D, where the two receivers are out of range of each other, yet the two transmitters (B, C) in the middle are in range of each other. Here, if a transmission between A and B is taking place, node C is prevented from transmitting to D as it concludes after carrier sense that it will interfere with the transmission by its neighbor node B. However note that node D could still receive the transmission of C without interference because it is out of range from B.

**Q. What is Hiper Access?**

A developing broadband wireless local loop (WLL) access technology specified in the Broadband Radio Access Networks (BRAN) project chartered by the European Telecommunications Standards Institute (ETSI). HiperACCESS is targeting frequencies in the 40.5

**Q. What are hybrid routing Protocols?**

Hybrid Routing Protocol (HRP) is a network routing protocol that combines Distance Vector Routing Protocol (DVRP) and Link State Routing Protocol (LSRP) features. HRP is used to determine optimal network destination routes and report network topology data modifications.

HRP is also known as Balanced Hybrid Routing (BHR).