**MOBILE COMPUTING LAB**

**(ETIT – 452)**

**Faculty name:** Mr. YOGESH **Student name:** Ayush Pandey

**Roll No.:** 45014802718

**Semester:** 8th Semester

**Group:** 8-C-9



Maharaja Agrasen Institute of Technology

PSP Area, Sector – 22, Rohini, New Delhi – 110085



**MAHARAJA AGRASEN INSTITUTE OF TECHNOLOGY**

**VISION**

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

**MISSION**

**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.

**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.



**MAHARAJA AGRASEN INSTITUTE OF TECHNOLOGY**

**COMPUTER SCIENCE AND ENGINEERING DEPARTMENT**

**VISION**

To produce “Critical Thinkers of Innovative Technology”.

**MISSION**

To foster an open, multidisciplinary and highly collaborative research environment for producing world-class engineers capable of providing innovative solutions to real life problems and fulfil societal needs.

**PRACTICAL RECORD**

**PAPER CODE : ETIT-452**

**Name of the student : Ayush Pandey**

**University Roll No. : 45014802718**

**Branch : CSE**

**Group : 8C-9**

**PRACTICAL DETAILS**

1. Experiments according to MC lab syllabus prescribed by GGSIPU

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ExpNo.** | **Experiment Name** | **Date Of Performance** | **Total Marks** | **Signature with Date** |
| 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.

**Theory:**

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| align | left right  center | Aligns the paragraph. Default is "left" |
| mode | wrap nowrap | Sets whether a paragraph should wrap lines or not. |
| xml:lang | *language\_code* | Sets the language used in the element |
| class | *cdata* | Sets a class name for the element. The class name is case sensitive. An element can be connected to multiple  classes. Multiple class names within the class attribute are separated by white space |
| id | *id* | Sets a unique name for the element |

|  |  |
| --- | --- |
| WML Elements |  |
| <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 |
|  | 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 |

**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 1">

<p>Hello World</p>

<onevent type="ontimer">

<go href="#ABC1"/>

</onevent>

<timer value="50"/>

</card>

<card id="ABC1" title="Practical 1">

<p>Ayush Pandey</b><br>

<big>Ayush Pandey</big><br>

<em>Ayush Pandey</em><br>

<i>Ayush Pandey</i><br>

<small>Ayush Pandey</small><br>

<strong>Ayush Pandey</strong><br>

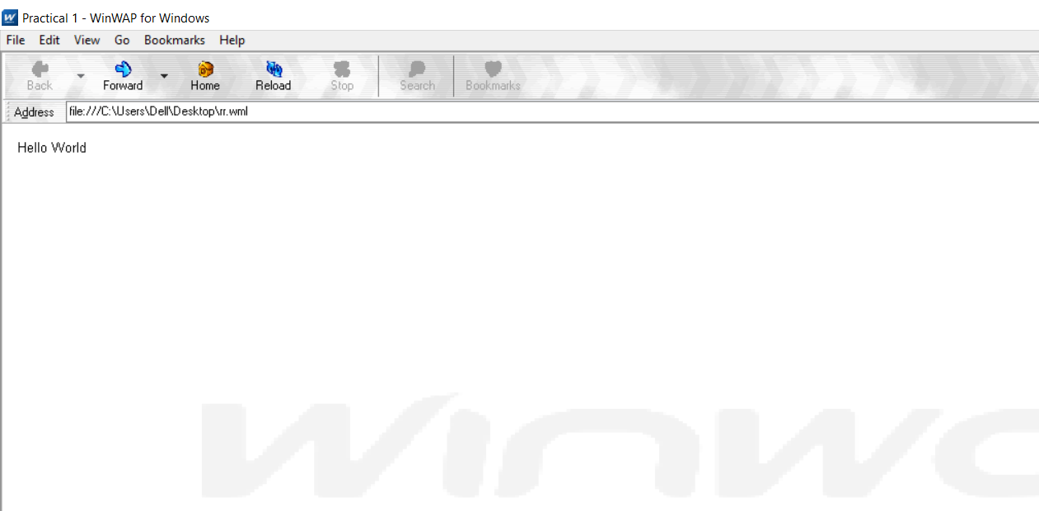
<u>Ayush Pandey</u>

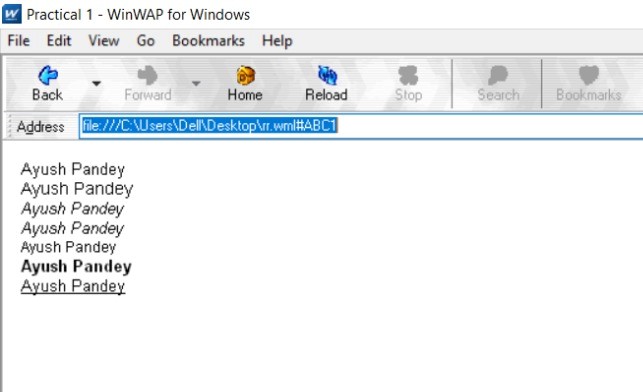
</p>

</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"?>

<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.3//EN"

"http://www.wapforum.org/DTD/wml13.dtd">

<wml>

<card id="CardNo.1" title="Program2Card1">

<p>You Are Visiting Card 1</p>

<do type="accept" name="CardNo.2" label="Continue">

<go href="#CardNo.2"/>

</do>

</card>

<card id="CardNo.2" title="Program2Card2">

<p>You Are Visiting Card 2</p>

<do type="accept" name="CardNo.3" label="Continue">

<go href="#CardNo.3"/>

</do>

</card>

<card id="CardNo.3" title="Program2Card3">

<p>You Are Visiting Card 3</p>

<do type="accept" name="CardNo.1" label="Go Back">

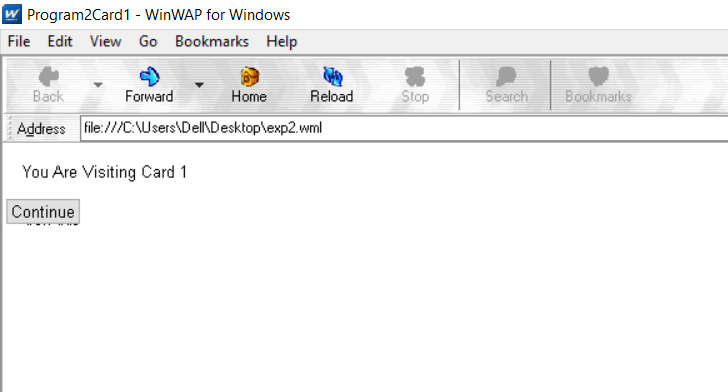
<go href="#CardNo.1"/>

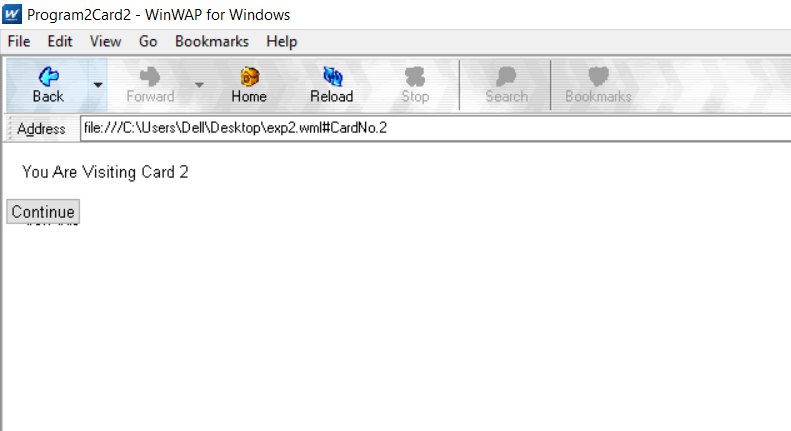
</do>

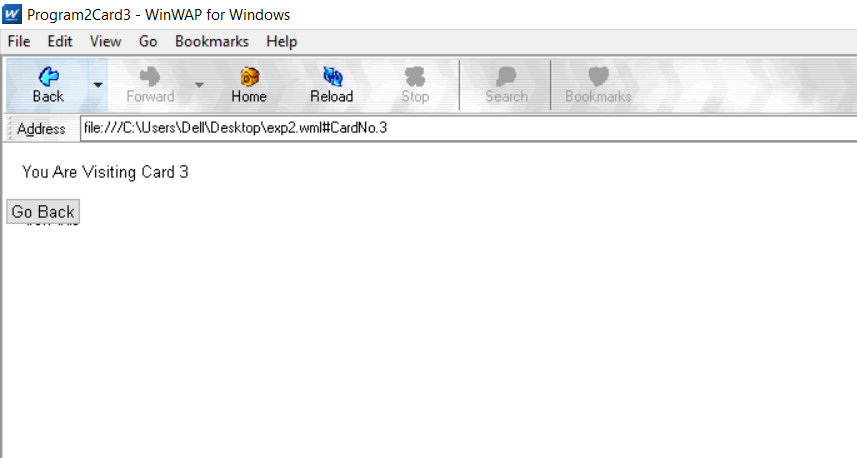
</card>

</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"?>

<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.3//EN" "http://www.wapforum.org/DTD/wml13.dtd">

<wml>

<card id="page1" title="Table in WML">

<p>

<table columns="1">

<tr>

<td>IMAGE NAME</td>

<td>IMAGE</td>

<td>LINK</td>

</tr>

<tr>

<td>Sunflower</td>

<td>

<img src="sun.jpg" alt="sun" height="62" width="60" />

</td>

<td>

<anchor>

<go href="exp3.wml#sun"/>

Link

</anchor>

</td>

</tr>

<tr>

<td>Rose</td>

<td>

<img src="rose.jpg" alt="rose" height="62" width="60" />

</td>

<td>

<anchor>

<go href="exp3.wml#rose"/>

Link

</anchor>

</td>

</tr>

<tr>

<td>Lily</td>

<td>

<img src="lily.jpg" alt="lily" height="62" width="60" />

</td>

<td>

<anchor>

<go href="exp3.wml#lily"/>

Link

</anchor>

</td>

</tr>

</table>

</p>

</card>

<card id="sun">

<img src="sun.jpg" />

</card>

<card id="rose">

<img src="rose.jpg" />

</card>

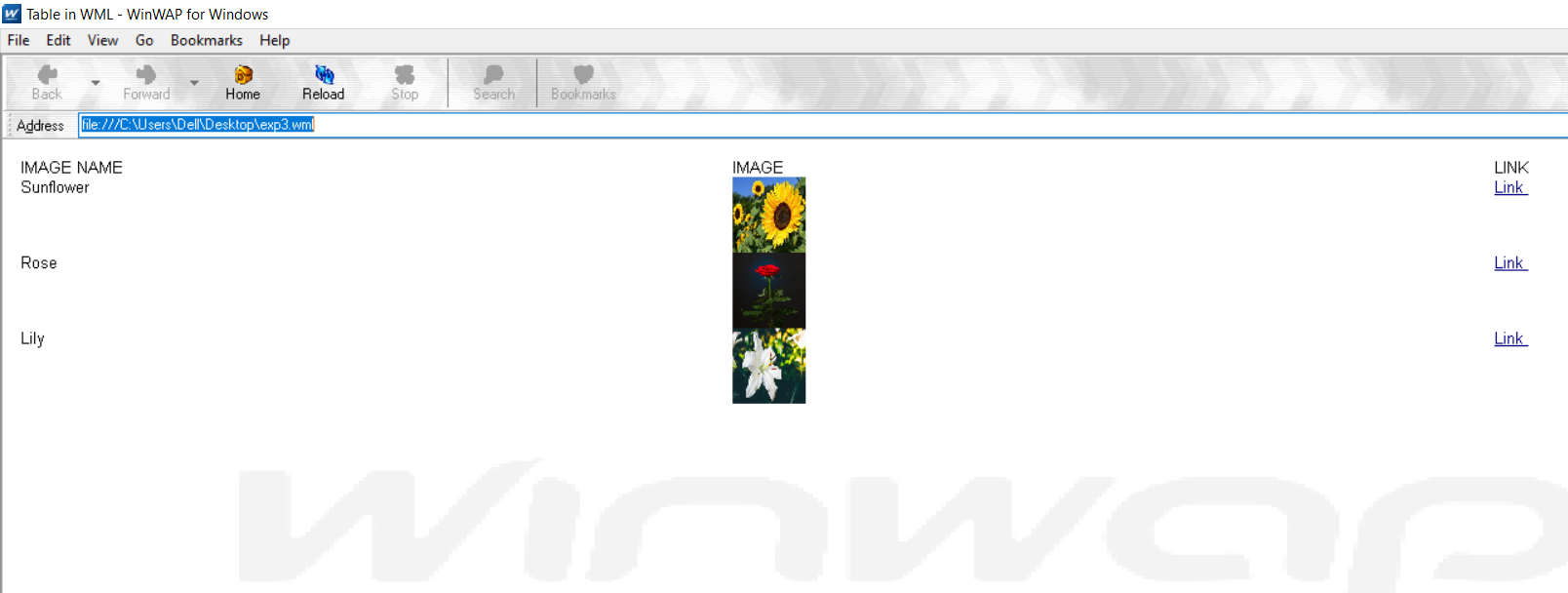
<card id="lily">

<img src="lily.jpg" />

</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">Ayush Pandey </option>

<option value="xml">Tannu Sharma</option>

<option value="wap">Garima Arora</option>

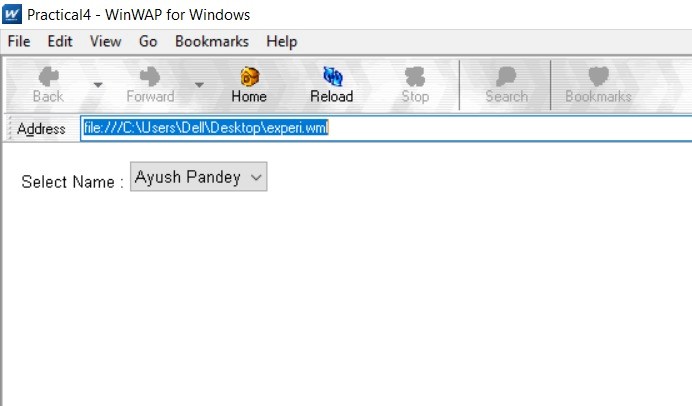
</select>

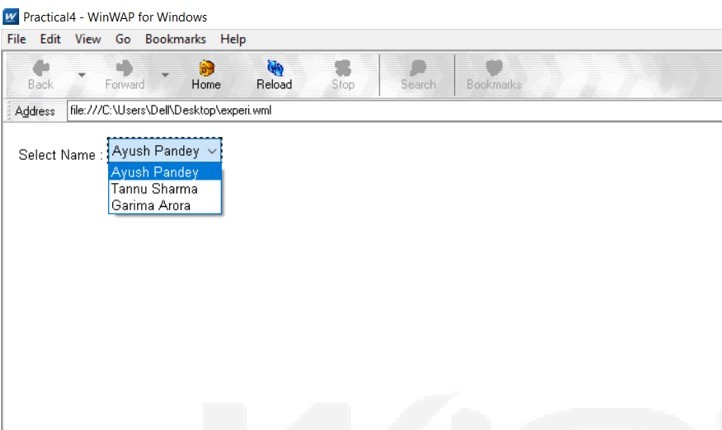
</p>

</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>

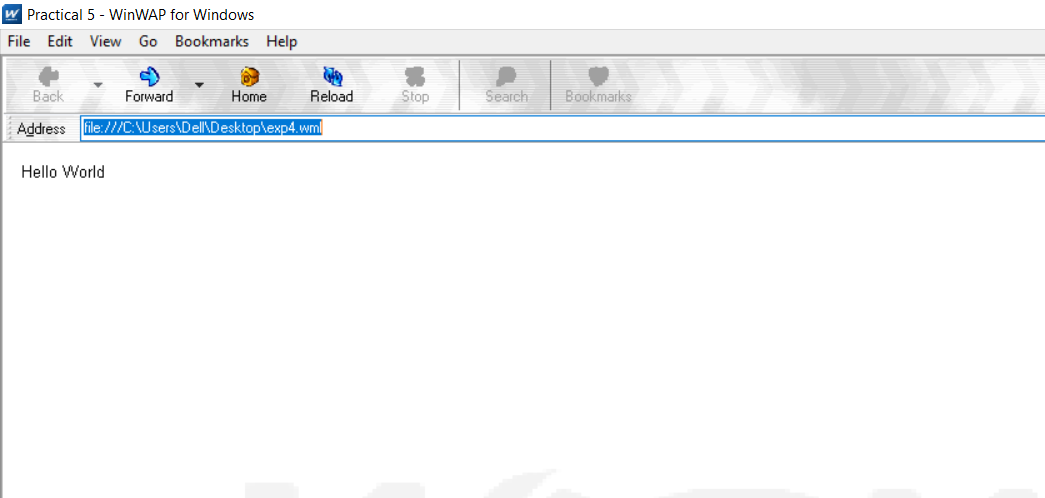
Wireless Application Protocol (WAP) is a technical standard for accessing information over a mobile wireless network.

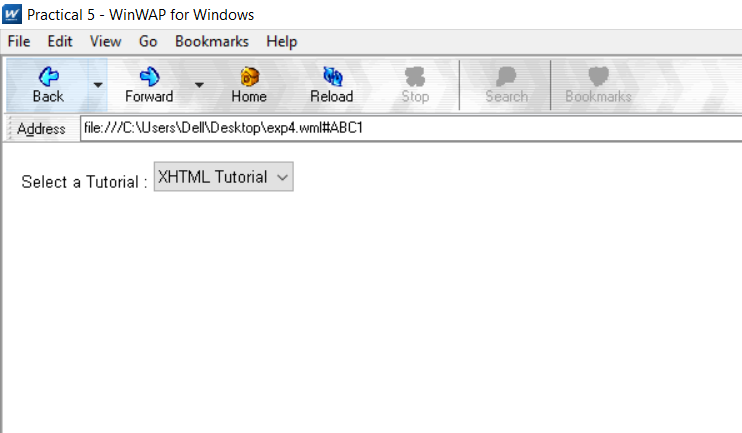
</p>

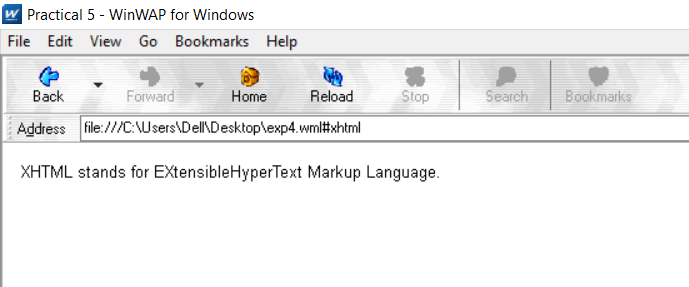
</card>

</wml>

**Output:**









**Viva Questions**

**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:**

// wml

<?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>

// 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(){

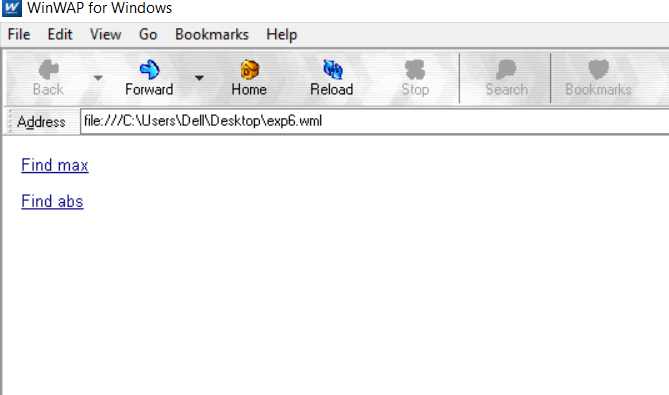
var num = Dialogs.prompt("Enter number", 0);

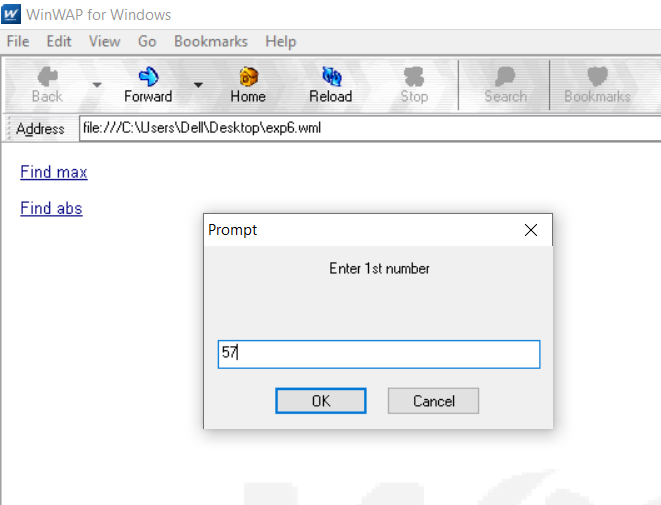
var absmun = Lang.abs(num);

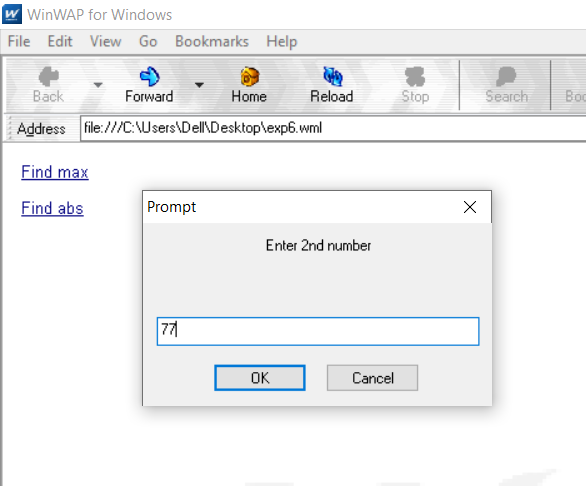
WMLBrowser.setVar("number", absmun);

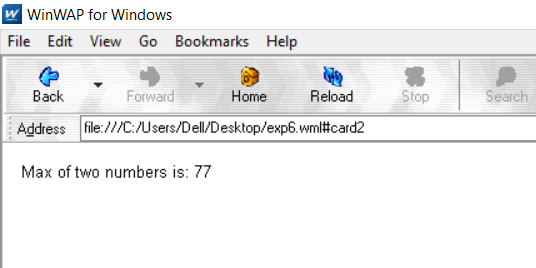
WMLBrowser.go("exp6.wml#card3");}

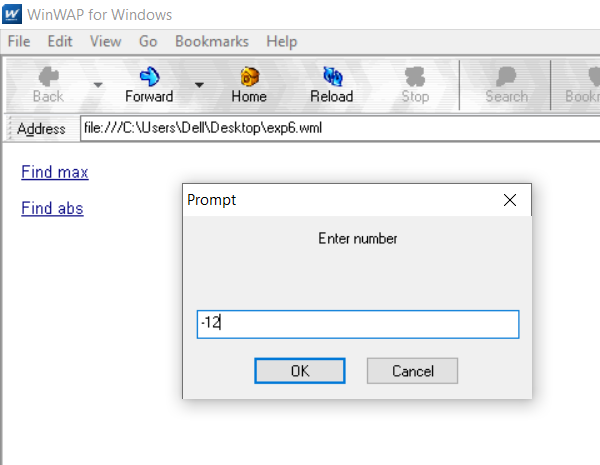
**Output:**

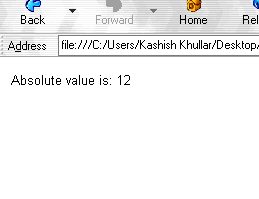












**Viva Questions**

**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 ();

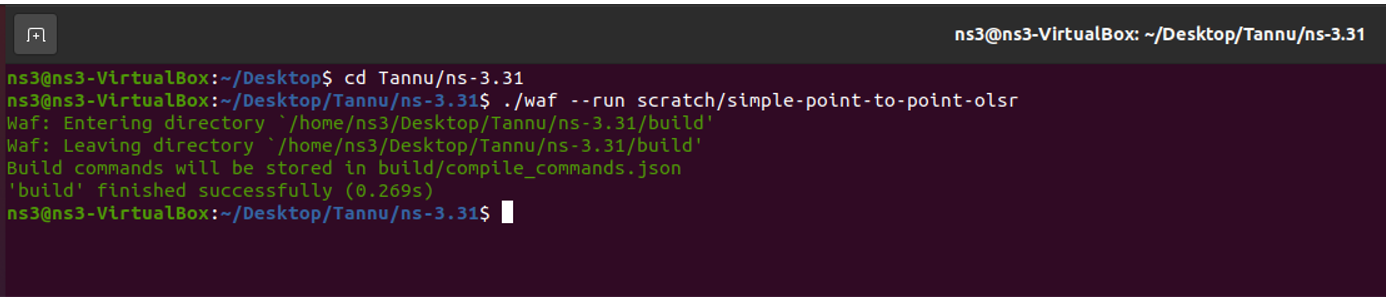
Simulator::Destroy ();

NS\_LOG\_INFO ("Done.");

return 0;

}

**Output:**



**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](https://www.nsnam.org/doxygen/namespacens3.html);

class [AodvExample](https://www.nsnam.org/doxygen/class_aodv_example.html) {

public:

[AodvExample](https://www.nsnam.org/doxygen/class_aodv_example.html" \l "ae088b79a280faec177faec7c15a30c62) ();

bool [Configure](https://www.nsnam.org/doxygen/class_aodv_example.html#a0700a4fd53f852f3eeca33d4721cc548) (int argc, char \*\*argv);

void [Run](https://www.nsnam.org/doxygen/class_aodv_example.html#a954ace72fd69c0a09193047d33d9ad61) ();

void [Report](https://www.nsnam.org/doxygen/class_aodv_example.html#a9ad0d438ce714e182d8a9b91521192e1) (std::ostream & os);

private:

[uint32\_t](https://www.nsnam.org/doxygen/classuint32__t.html) [size](https://www.nsnam.org/doxygen/class_aodv_example.html#a981713615ce1e6eaf66fe20c0fae3136);

double [step](https://www.nsnam.org/doxygen/class_aodv_example.html#af2cfac181b97285129c0d17a5edd66f0);

double [totalTime](https://www.nsnam.org/doxygen/class_aodv_example.html#a842f2dbbf8a6e4e33c275cbfa2d20bf2);

bool [pcap](https://www.nsnam.org/doxygen/class_aodv_example.html#a562ab7581322f11c301999cb365ba33e);

bool [printRoutes](https://www.nsnam.org/doxygen/class_aodv_example.html#a51ce8d78ff5c5e9725038d3c6fd29a33);

[NodeContainer](https://www.nsnam.org/doxygen/classns3_1_1_node_container.html) [nodes](https://www.nsnam.org/doxygen/class_aodv_example.html#a6a0e7d16457d9460d27f8a9407e2bb7f);

[NetDeviceContainer](https://www.nsnam.org/doxygen/classns3_1_1_net_device_container.html) [devices](https://www.nsnam.org/doxygen/class_aodv_example.html#ac84e7dbc2ac26566258b3ec8ee0dd0b3);

[Ipv4InterfaceContainer](https://www.nsnam.org/doxygen/classns3_1_1_ipv4_interface_container.html) [interfaces](https://www.nsnam.org/doxygen/class_aodv_example.html#a4208ce6498334663b627c2fa0d2f5ed5);

private:

void [CreateNodes](https://www.nsnam.org/doxygen/class_aodv_example.html#a8f39319e44e5bae98359b0b11f905281) ();

void [CreateDevices](https://www.nsnam.org/doxygen/class_aodv_example.html#a0e9be9e9b05169dfe7d440ac2dfa3252) ();

void [InstallInternetStack](https://www.nsnam.org/doxygen/class_aodv_example.html#abc2e6b6dd4791571042bc65a1340efd5) ();

void [InstallApplications](https://www.nsnam.org/doxygen/class_aodv_example.html#a181ce6b0304fa53ac0db6cb0228f93f7) ();};

int main (int argc, char \*\*argv){

[AodvExample](https://www.nsnam.org/doxygen/class_aodv_example.html) test;

if (!test.Configure (argc, argv))

[NS\_FATAL\_ERROR](https://www.nsnam.org/doxygen/group__fatal.html" \l "ga5131d5e3f75d7d4cbfd706ac456fdc85) ("Configuration failed. Aborted.");

test.Run ();

test.Report (std::cout);

return 0; }

[AodvExample::AodvExample](https://www.nsnam.org/doxygen/class_aodv_example.html" \l "ae088b79a280faec177faec7c15a30c62) ():

size (10),

step (50),

totalTime (100),

pcap (true),

printRoutes (true) {}

bool [AodvExample::Configure](https://www.nsnam.org/doxygen/class_aodv_example.html#a0700a4fd53f852f3eeca33d4721cc548) (int argc, char \*\*argv) {

SeedManager::SetSeed (12345);

[CommandLine](https://www.nsnam.org/doxygen/classns3_1_1_command_line.html) [cmd](https://www.nsnam.org/doxygen/namespacesecond.html#ae84c87468c0e4398eab4dbb6cadc17d9) (\_\_FILE\_\_);

[cmd](https://www.nsnam.org/doxygen/namespacesecond.html" \l "ae84c87468c0e4398eab4dbb6cadc17d9).AddValue ("pcap", "Write PCAP traces.", [pcap](https://www.nsnam.org/doxygen/class_aodv_example.html#a562ab7581322f11c301999cb365ba33e));

[cmd](https://www.nsnam.org/doxygen/namespacesecond.html" \l "ae84c87468c0e4398eab4dbb6cadc17d9).AddValue ("printRoutes", "Print routing table dumps.", [printRoutes](https://www.nsnam.org/doxygen/class_aodv_example.html#a51ce8d78ff5c5e9725038d3c6fd29a33));

[cmd](https://www.nsnam.org/doxygen/namespacesecond.html" \l "ae84c87468c0e4398eab4dbb6cadc17d9).AddValue ("size", "Number of nodes.", [size](https://www.nsnam.org/doxygen/class_aodv_example.html#a981713615ce1e6eaf66fe20c0fae3136));

[cmd](https://www.nsnam.org/doxygen/namespacesecond.html" \l "ae84c87468c0e4398eab4dbb6cadc17d9).AddValue ("time", "Simulation time, s.", [totalTime](https://www.nsnam.org/doxygen/class_aodv_example.html#a842f2dbbf8a6e4e33c275cbfa2d20bf2));

[cmd](https://www.nsnam.org/doxygen/namespacesecond.html" \l "ae84c87468c0e4398eab4dbb6cadc17d9).AddValue ("step", "Grid step, m", [step](https://www.nsnam.org/doxygen/class_aodv_example.html#af2cfac181b97285129c0d17a5edd66f0));

[cmd](https://www.nsnam.org/doxygen/namespacesecond.html" \l "ae84c87468c0e4398eab4dbb6cadc17d9).Parse (argc, argv);

return true; }

void [AodvExample::Run](https://www.nsnam.org/doxygen/class_aodv_example.html#a954ace72fd69c0a09193047d33d9ad61) () {

[CreateNodes](https://www.nsnam.org/doxygen/class_aodv_example.html" \l "a8f39319e44e5bae98359b0b11f905281) ();

[CreateDevices](https://www.nsnam.org/doxygen/class_aodv_example.html" \l "a0e9be9e9b05169dfe7d440ac2dfa3252) ();

[InstallInternetStack](https://www.nsnam.org/doxygen/class_aodv_example.html" \l "abc2e6b6dd4791571042bc65a1340efd5) ();

[InstallApplications](https://www.nsnam.org/doxygen/class_aodv_example.html" \l "a181ce6b0304fa53ac0db6cb0228f93f7) ();

std::cout << "Starting simulation for " << [totalTime](https://www.nsnam.org/doxygen/class_aodv_example.html#a842f2dbbf8a6e4e33c275cbfa2d20bf2) << " s ...\n";

Simulator::Stop ([Seconds](https://www.nsnam.org/doxygen/group__timecivil.html#ga33c34b816f8ff6628e33d5c8e9713b9e) ([totalTime](https://www.nsnam.org/doxygen/class_aodv_example.html" \l "a842f2dbbf8a6e4e33c275cbfa2d20bf2)));

Simulator::Run ();

Simulator::Destroy (); }

void [AodvExample::Report](https://www.nsnam.org/doxygen/class_aodv_example.html#a9ad0d438ce714e182d8a9b91521192e1) (std::ostream &){}

void [AodvExample::CreateNodes](https://www.nsnam.org/doxygen/class_aodv_example.html#a8f39319e44e5bae98359b0b11f905281) (){

std::cout << "Creating " << (unsigned)[size](https://www.nsnam.org/doxygen/class_aodv_example.html#a981713615ce1e6eaf66fe20c0fae3136) << " nodes " << [step](https://www.nsnam.org/doxygen/class_aodv_example.html#af2cfac181b97285129c0d17a5edd66f0) << " m apart.\n";

[nodes](https://www.nsnam.org/doxygen/class_aodv_example.html" \l "a6a0e7d16457d9460d27f8a9407e2bb7f).[Create](https://www.nsnam.org/doxygen/classns3_1_1_node_container.html#a787f059e2813e8b951cc6914d11dfe69) ([size](https://www.nsnam.org/doxygen/class_aodv_example.html#a981713615ce1e6eaf66fe20c0fae3136));

for ([uint32\_t](https://www.nsnam.org/doxygen/classuint32__t.html) i = 0; i < [size](https://www.nsnam.org/doxygen/class_aodv_example.html#a981713615ce1e6eaf66fe20c0fae3136); ++i){

std::ostringstream os;

os << "node-" << i;

Names::Add (os.str (), [nodes](https://www.nsnam.org/doxygen/class_aodv_example.html#a6a0e7d16457d9460d27f8a9407e2bb7f).[Get](https://www.nsnam.org/doxygen/classns3_1_1_node_container.html#aff885e5a6b02ca4a27f5725de647d96e) (i)); }

[MobilityHelper](https://www.nsnam.org/doxygen/classns3_1_1_mobility_helper.html) [mobility](https://www.nsnam.org/doxygen/namespacethird.html#ae696e36bf1ea3312c33cec7727af7ab4);

[mobility](https://www.nsnam.org/doxygen/namespacethird.html" \l "ae696e36bf1ea3312c33cec7727af7ab4).SetPositionAllocator ("ns3::GridPositionAllocator", "MinX", [DoubleValue](https://www.nsnam.org/doxygen/classns3_1_1_double_value.html) (0.0), "MinY", [DoubleValue](https://www.nsnam.org/doxygen/classns3_1_1_double_value.html) (0.0), "DeltaX", [DoubleValue](https://www.nsnam.org/doxygen/classns3_1_1_double_value.html) ([step](https://www.nsnam.org/doxygen/class_aodv_example.html#af2cfac181b97285129c0d17a5edd66f0)), "DeltaY", [DoubleValue](https://www.nsnam.org/doxygen/classns3_1_1_double_value.html) (0), "GridWidth", [UintegerValue](https://www.nsnam.org/doxygen/classns3_1_1_uinteger_value.html) ([size](https://www.nsnam.org/doxygen/class_aodv_example.html#a981713615ce1e6eaf66fe20c0fae3136)), "LayoutType", [StringValue](https://www.nsnam.org/doxygen/classns3_1_1_string_value.html) ("RowFirst"));

[mobility](https://www.nsnam.org/doxygen/namespacethird.html" \l "ae696e36bf1ea3312c33cec7727af7ab4).SetMobilityModel ("ns3::ConstantPositionMobilityModel");

[mobility](https://www.nsnam.org/doxygen/namespacethird.html" \l "ae696e36bf1ea3312c33cec7727af7ab4).Install ([nodes](https://www.nsnam.org/doxygen/class_aodv_example.html#a6a0e7d16457d9460d27f8a9407e2bb7f)); }

void [AodvExample::CreateDevices](https://www.nsnam.org/doxygen/class_aodv_example.html#a0e9be9e9b05169dfe7d440ac2dfa3252) (){

[WifiMacHelper](https://www.nsnam.org/doxygen/classns3_1_1_wifi_mac_helper.html) wifiMac;

wifiMac.[SetType](https://www.nsnam.org/doxygen/classns3_1_1_wifi_mac_helper.html#ae50d00a6612955d618c800de4e009759) ("ns3::AdhocWifiMac");

[YansWifiPhyHelper](https://www.nsnam.org/doxygen/classns3_1_1_yans_wifi_phy_helper.html) wifiPhy;

[YansWifiChannelHelper](https://www.nsnam.org/doxygen/classns3_1_1_yans_wifi_channel_helper.html) wifiChannel = YansWifiChannelHelper::Default ();

wifiPhy.[SetChannel](https://www.nsnam.org/doxygen/classns3_1_1_yans_wifi_phy_helper.html#ad2e9a27587dd4ff320435c93cc2676de) (wifiChannel.[Create](https://www.nsnam.org/doxygen/classns3_1_1_yans_wifi_channel_helper.html#a6bb01082fd42806a6b95e6684a4344d4) ());

[WifiHelper](https://www.nsnam.org/doxygen/classns3_1_1_wifi_helper.html) [wifi](https://www.nsnam.org/doxygen/namespacethird.html#ac5ca8854aaed8d24996692bb65732537);

[wifi](https://www.nsnam.org/doxygen/namespacethird.html" \l "ac5ca8854aaed8d24996692bb65732537).SetRemoteStationManager ("ns3::ConstantRateWifiManager", "DataMode",

[StringValue](https://www.nsnam.org/doxygen/classns3_1_1_string_value.html)("OfdmRate6Mbps"), "RtsCtsThreshold", [UintegerValue](https://www.nsnam.org/doxygen/classns3_1_1_uinteger_value.html) (0));

[devices](https://www.nsnam.org/doxygen/class_aodv_example.html" \l "ac84e7dbc2ac26566258b3ec8ee0dd0b3) = [wifi](https://www.nsnam.org/doxygen/namespacethird.html#ac5ca8854aaed8d24996692bb65732537).Install (wifiPhy, wifiMac, [nodes](https://www.nsnam.org/doxygen/class_aodv_example.html#a6a0e7d16457d9460d27f8a9407e2bb7f));

if ([pcap](https://www.nsnam.org/doxygen/class_aodv_example.html" \l "a562ab7581322f11c301999cb365ba33e)){

wifiPhy.[EnablePcapAll](https://www.nsnam.org/doxygen/classns3_1_1_pcap_helper_for_device.html#a4ab183a2512120200d4a0e5d8ececd49) (std::string ("aodv")); }}

void [AodvExample::InstallInternetStack](https://www.nsnam.org/doxygen/class_aodv_example.html#abc2e6b6dd4791571042bc65a1340efd5) (){

[AodvHelper](https://www.nsnam.org/doxygen/classns3_1_1_aodv_helper.html) aodv;

[InternetStackHelper](https://www.nsnam.org/doxygen/classns3_1_1_internet_stack_helper.html) [stack](https://www.nsnam.org/doxygen/namespacefirst.html#adc1ab139ff29103499c3f54f3e77189d);

[stack](https://www.nsnam.org/doxygen/namespacefirst.html" \l "adc1ab139ff29103499c3f54f3e77189d).SetRoutingHelper (aodv); // has effect on the next Install ()

[stack](https://www.nsnam.org/doxygen/namespacefirst.html" \l "adc1ab139ff29103499c3f54f3e77189d).Install ([nodes](https://www.nsnam.org/doxygen/class_aodv_example.html#a6a0e7d16457d9460d27f8a9407e2bb7f));

[Ipv4AddressHelper](https://www.nsnam.org/doxygen/classns3_1_1_ipv4_address_helper.html) [address](https://www.nsnam.org/doxygen/namespacefirst.html#a121ad9b1005eb06f5256201299b392b2);

[address](https://www.nsnam.org/doxygen/namespacefirst.html" \l "a121ad9b1005eb06f5256201299b392b2).SetBase ("10.0.0.0", "255.0.0.0");

[interfaces](https://www.nsnam.org/doxygen/class_aodv_example.html" \l "a4208ce6498334663b627c2fa0d2f5ed5) = [address](https://www.nsnam.org/doxygen/namespacefirst.html#a121ad9b1005eb06f5256201299b392b2).Assign ([devices](https://www.nsnam.org/doxygen/class_aodv_example.html#ac84e7dbc2ac26566258b3ec8ee0dd0b3));

if ([printRoutes](https://www.nsnam.org/doxygen/class_aodv_example.html" \l "a51ce8d78ff5c5e9725038d3c6fd29a33)){

[Ptr<OutputStreamWrapper>](https://www.nsnam.org/doxygen/classns3_1_1_ptr.html) routingStream = Create<OutputStreamWrapper> ("aodv.routes", std::ios::out);

aodv.[PrintRoutingTableAllAt](https://www.nsnam.org/doxygen/classns3_1_1_ipv4_routing_helper.html#a10d8ba23eca330de968fdfbca2730ec3) ([Seconds](https://www.nsnam.org/doxygen/group__timecivil.html#ga33c34b816f8ff6628e33d5c8e9713b9e) (8), routingStream);}}

void [AodvExample::InstallApplications](https://www.nsnam.org/doxygen/class_aodv_example.html#a181ce6b0304fa53ac0db6cb0228f93f7) (){

[V4PingHelper](https://www.nsnam.org/doxygen/classns3_1_1_v4_ping_helper.html) ping ([interfaces](https://www.nsnam.org/doxygen/class_aodv_example.html" \l "a4208ce6498334663b627c2fa0d2f5ed5).[GetAddress](https://www.nsnam.org/doxygen/classns3_1_1_ipv4_interface_container.html#ae708067c27bd8019edd3c0311ff150b3) ([size](https://www.nsnam.org/doxygen/class_aodv_example.html#a981713615ce1e6eaf66fe20c0fae3136) - 1));

ping.[SetAttribute](https://www.nsnam.org/doxygen/classns3_1_1_v4_ping_helper.html#a834c093b4c031d64a55b3c70c6b2c464) ("Verbose", [BooleanValue](https://www.nsnam.org/doxygen/classns3_1_1_boolean_value.html) (true));

[ApplicationContainer](https://www.nsnam.org/doxygen/classns3_1_1_application_container.html) p = ping.[Install](https://www.nsnam.org/doxygen/classns3_1_1_v4_ping_helper.html#a0599aa44cbcc6a0ad69299a7f900f944) ([nodes](https://www.nsnam.org/doxygen/class_aodv_example.html" \l "a6a0e7d16457d9460d27f8a9407e2bb7f).[Get](https://www.nsnam.org/doxygen/classns3_1_1_node_container.html#aff885e5a6b02ca4a27f5725de647d96e) (0));

p.[Start](https://www.nsnam.org/doxygen/classns3_1_1_application_container.html" \l "a8eff87926507020bbe3e1390358a54a7) ([Seconds](https://www.nsnam.org/doxygen/group__timecivil.html#ga33c34b816f8ff6628e33d5c8e9713b9e) (0));

p.[Stop](https://www.nsnam.org/doxygen/classns3_1_1_application_container.html#adfc52f9aa4020c8714679b00bbb9ddb3) ([Seconds](https://www.nsnam.org/doxygen/group__timecivil.html#ga33c34b816f8ff6628e33d5c8e9713b9e) ([totalTime](https://www.nsnam.org/doxygen/class_aodv_example.html" \l "a842f2dbbf8a6e4e33c275cbfa2d20bf2)) - [Seconds](https://www.nsnam.org/doxygen/group__timecivil.html#ga33c34b816f8ff6628e33d5c8e9713b9e) (0.001));

// move node away

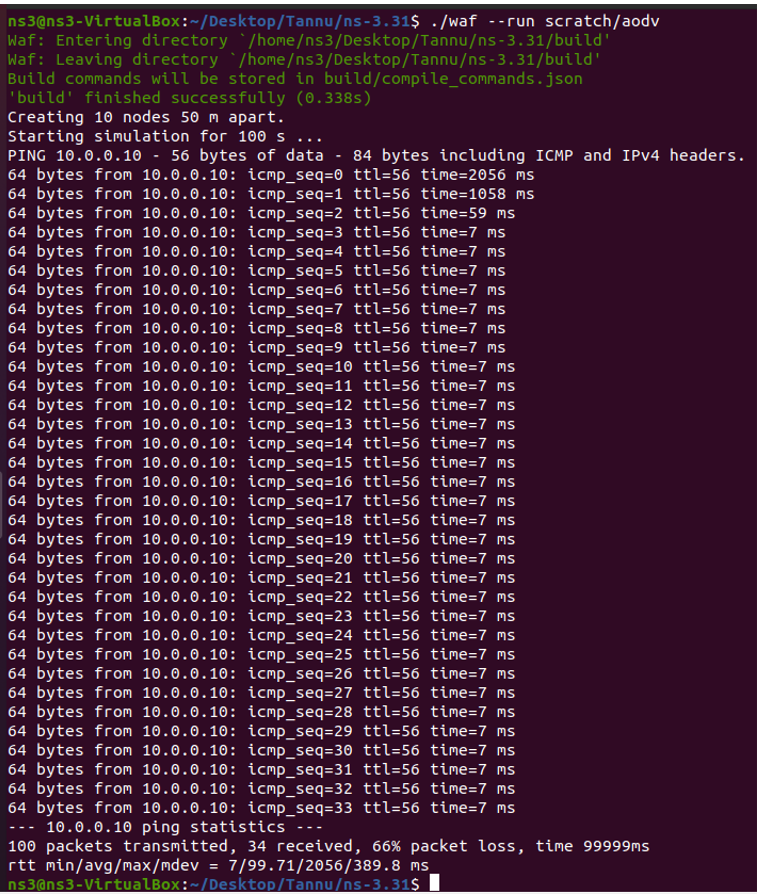
[Ptr<Node>](https://www.nsnam.org/doxygen/classns3_1_1_ptr.html) node = [nodes](https://www.nsnam.org/doxygen/class_aodv_example.html#a6a0e7d16457d9460d27f8a9407e2bb7f).[Get](https://www.nsnam.org/doxygen/classns3_1_1_node_container.html#aff885e5a6b02ca4a27f5725de647d96e) ([size](https://www.nsnam.org/doxygen/class_aodv_example.html#a981713615ce1e6eaf66fe20c0fae3136)/2);

[Ptr<MobilityModel>](https://www.nsnam.org/doxygen/classns3_1_1_ptr.html) mob = node->[GetObject](https://www.nsnam.org/doxygen/classns3_1_1_object.html" \l "a817117ab236e4ff7f80f0239d4e28731)<[MobilityModel](https://www.nsnam.org/doxygen/classns3_1_1_mobility_model.html)> ();

Simulator::Schedule ([Seconds](https://www.nsnam.org/doxygen/group__timecivil.html#ga33c34b816f8ff6628e33d5c8e9713b9e) ([totalTime](https://www.nsnam.org/doxygen/class_aodv_example.html" \l "a842f2dbbf8a6e4e33c275cbfa2d20bf2)/3), &[MobilityModel::SetPosition](https://www.nsnam.org/doxygen/wifi-ap_8cc.html" \l "aac02abe91b3e5592e31bc0f7d844bf4d), mob, Vector (1e5, 1e5, 1e5));

}

**Output:**



**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 address are used as identification but since the nodes in the ad hoc networks are moving, they dont 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 node 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).