Don Bosco Institute of Technology
Department of Information Technology
Wireless Technology

BE-IT **SEM 7**

Name: Christine Polly **Roll no: 17**

Experiment No 10

Theory:

WHAT IS HANDOFF? :

A handoff refers to the process of transferring an active call or data session from one cell in a cellular network to another or from one channel in a cell to another. A well-implemented handoff is important for delivering uninterrupted service to a caller or data session user. In simple terms when the user is making a call, the particular signal is accepted by the cell tower in its vicinity and if the user decides to move to another location, far from the initial cell tower, the signal might go out of range, hence the cell tower will hand over the connection to another cell tower near to the current location of the user thereby giving an uninterrupted service. Similarly, the same concept applies in Wi-Fi where devices interact with router (Wireless Access Point) to send or receive data.

HANDOFF SIMULATION :

In the following simulation we show how handoff is attempted from one access point to another access point. When one node changes its position, it changes its access points as well to whichever is near. We simulate the following concept by using 2 Access points and 8 nodes (devices).

Since it's a wireless network, we have to configure a lot of settings for the node such as the channel type, the type of routing protocol we are using (here dumb agent), MAC type, link layer, propagation type, etc. We use DumbAgent as the network layer protocol when no routing is required in the test scenario. This is typically used when we need to calculate the MAC layer performance specifically. In DumbAgent it is assumed that the scenario is such that each source and destination node are directly neighbours and within one hop distance from each which enables us to not have any routing techniques. We define all this in the initial segment and assign them to different variables.

Now to initialise all the nodes with this configuration we run them through a loop and set the nodes with same properties.
NETWORK SETTINGS set val(chan) Channel/WirelessChannel ;#Channel Type
set val(prop) Propagation/TwoRayGround ;# radio-propagation model
set val(netif) Phy/WirelessPhy ;# network interface type
set val(mac) Mac/802_11 ;# MAC type
set val(ifq) Queue/DropTail/PriQueue ;# interface queue type
set val(ll) LL ;# link layer type
set val(ant) Antenna/OmniAntenna ;# antenna model
set val(ifqlen) 50 ;# max packet in ifq
set val(nn) 10 ;# number of mobilenodes
set val(rp) DumbAgent ;# routing protocol
set val(x) 600
set val(y) 600#setting mac of each node. It gets the mac-id (by the function getMac). Before this, it was assigned the MAC type configuration but each node has a unique MAC address hence we loop through all nodes and give them unique IDs. Each device in a network has its own MAC ID which

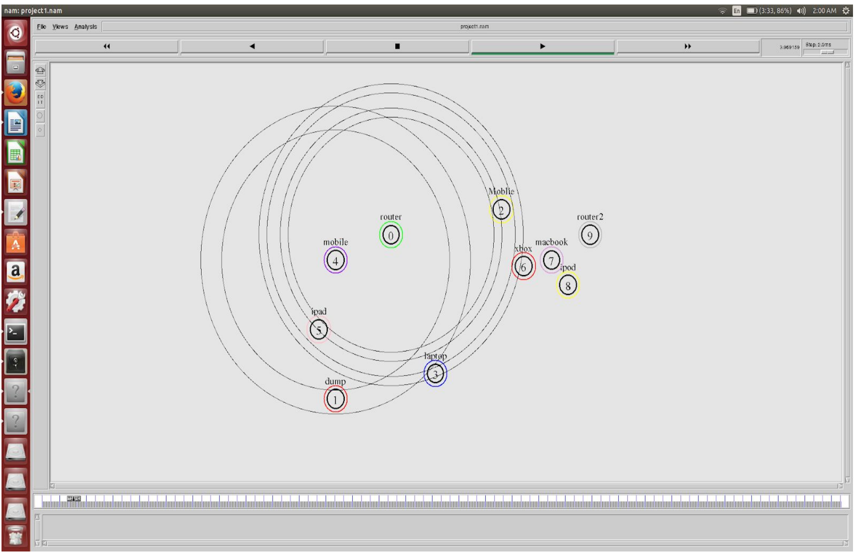
is unique. It is a type of an identifier for it. We get the MAC address of each node by using the getMac function .

```
set mac_($i) [$node_($i) getMac 0]
```

In the next step, we assign positions of each node and give them a unique label and colour to make them distinguishable.

```
$ns_ at 0.0 "$node_(0) add-mark m1 green circle"
$ns_ at 0.0 "$node_(1) add-mark m1 red circle"
$ns_ at 0.0 "$node_(2) add-mark m1 yellow circle"
$ns_ at 0.0 "$node_(3) add-mark m1 blue circle"
```

Then we setup the access points and give their MAC IDs respectively. Now there are two types of scan type -active scan type and passive. When the nodes are set as the scan type active, it makes sure that the nodes are continuously scanning for nearby access points. Whereas in passive the nodes don't scan continuously but scan only after a set time. In our case, around 100ms and check if any beacon signal is sent by the nearest AP. We set the node 1 as active since it acts as a null agent for all the nodes. That is all the UPD packets of the nodes are dumped here since it should always remain active to receive the nodes.



The other nodes are set as passive and interact only after a certain time of 100ms. Now we define what kind of protocol the nodes use (TCP or UDP). We assign them with the UDP protocol and the application using UDP protocol is CBR (constant bitrate). We use UDP as we don't need to know about the acknowledgement. We then attach the application CBR to each of the nodes.

```
Application/Traffic/CBR set packetSize_ 1023
Application/Traffic/CBR set rate_ 256Kb
```

All the null agents are set and the UDP application of each node is connected to node 1 (acting as destination for all the nodes) . With set null0 ,we are defining the destination of udp by a pointer called null .The next command defines the destination node and we attach the null pointer to that destinatio node that receives traffic .with connect command we are attaching source to the destination .

Connecting nodes with null agent

```
set null0 [new Agent/Null]
$ns_ attach-agent $node_(1) $null0
$ns_ connect $udp1(2) $null0
```

```
set null1 [new Agent/Null]
$ns_ attach-agent $node_(1) $null1
$ns_ connect $udp1(3) $null1
```

At time 2, 3, 4... each node starts sending packets (ie; the CBR starts) and connects to the nearest access point. Each node scans for the nearest active access point every 100ms.
The nodes are then set to start sending bits at the time allotted as follows.

```
$ns_ at 8.0 "$cbr1(2) start"
$ns_ at 2.0 "$cbr1(3) start"
$ns_ at 3.0 "$cbr1(4) start"
$ns_ at 4.0 "$cbr1(5) start"
$ns_ at 5.0 "$cbr1(6) start"
$ns_ at 6.0 "$cbr1(7) start"
$ns_ at 7.0 "$cbr1(8) start"
```

POSITIONS

The initial positions and the final positions of the nodes examined are :

```
$ns_ at 10.0 "$node_(4) setdest 590.0 350.0 1000.0"
$ns_ at 35.0 "$node_(5) setdest 460.0 360.0 1000.0"
$ns_ at 50.0 "$node_(3) setdest 590.0 350.0 1000.0"
$ns_ at 52.0 "$node_(3) setdest 100.0 360.0 1000.0"
```

Here the node 4 changes its position from 590 to 1000. The node 4 is initially connected to the access point AP1. When it moves from position 590 to 1000, as the access points send beacon signals to all the nodes, node 4 recognizes the signal from AP2 as it leaves the range of the access point AP1. Here a handoff is attempted. Similarly any node that changes its position, scans for the nearest access point and handoff is attempted between the access points. After the handoff is attempted, the node begins to interact with the new access point it is connected to.

The following are all the handoffs of the various nodes that are attempted from the code:

OUTPUT

```
Client 4: Handoff Attempted

Client 4: Handoff from AP 0 to AP 9

Client 5: Handoff Attempted

Client 5: Handoff from AP 0 to AP 9

Client 3: Handoff Attempted

Client 3: Handoff from AP 0 to AP 9
```

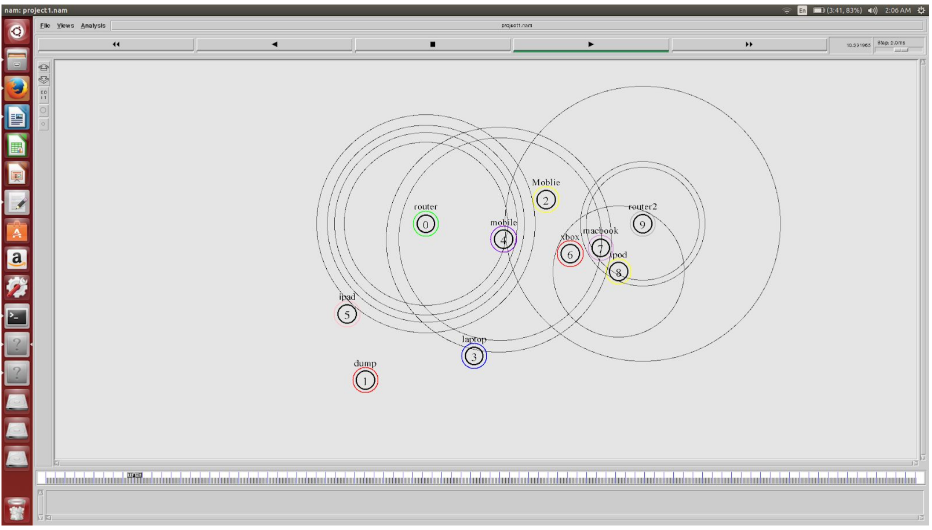
Client 3: Handoff Attempted

Client 3: Handoff from AP 9 to AP 0

The node 4 which attempts a handoff at time 10s
node 5 at 35.
node 3 at 50.
node 3 again at 55, as it comes back to its initial access point .

As we can see, the mobile is moving from the vicinity of router 1 to router 2, and here the handoff occurs. The router 2 accepts the handoff and the mobile starts interacting with the router 2.

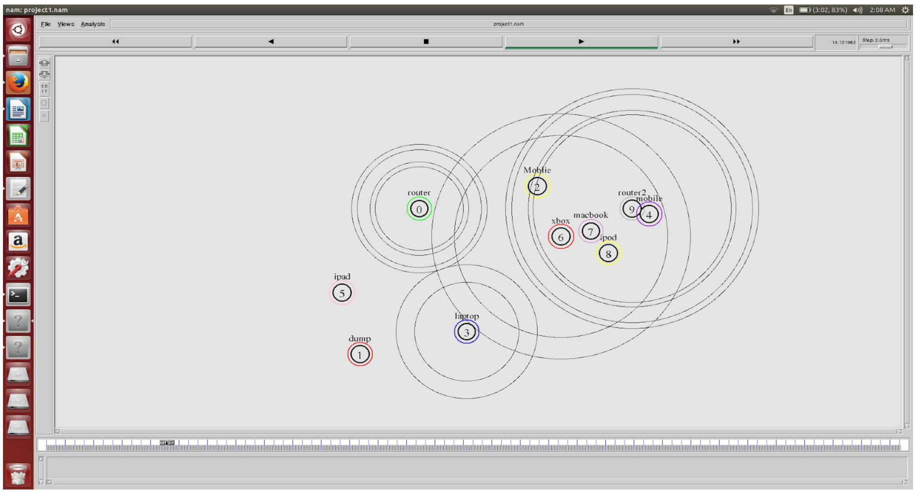
The new position of the mobile, near access point 2 .



FINISH PROCEDURE

```
proc stop {} {
global ns_ tracefd
$ns_ flush-trace
close $tracefd
exec nam project1.nam
exit 0
}
```

All the data of the simulation are saved in the trace file.


nam: project1.nam
File
Views
Analysis
project1.nam
router
0
Moblie
2
router2
9
mobile
4
macbook
7
xbox
6
ipod
8
ipad
5
laptop
3
dump
1