Department of Computer Engineering

Academic Term : Jan-Apr 2023

**Class : ** T.E Computer Sem -VI
**Subject : ** Mobile Computing

Practical No:	**4**
Title:	Illustration of Hidden Terminal Problem (NS2)
Date of Performance:	24/02/2025
Date of Submission:	24/02/2025
Roll No:	9913
Name of the Student:	Mark Lopes

Evaluation:

Sr. No	Rubric	Grade
1	**On time Completion & Submission(2)**	
2	**Output(3)**	
3	**Code Optimization(3)**	
4	**Knowledge of the topic(2)**	
5	**Total (10)**	

Signature of the Teacher :

ExperimentNo.:4

Aim: Illustration of Hidden Terminal Problem (NS2)

thoery:

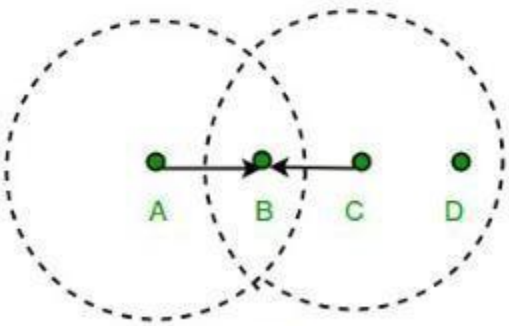
A wireless network with lack of centralized control entity, sharing of wireless bandwidth amongnetworkaccessnodesi.e.mediumaccesscontrol(MAC)nodesmustbeorganizedindecentralized manner. The hidden terminal problem occurs when a terminal is visible from awirelessaccess point(APs), butnot from othernodescommunicatingwiththatAP.Thissituationleadsthedifficultiesinmediumaccesscontrolsublayeroverwirelessnetworking.

In a formal way hidden terminal are nodes in a wireless network that are out of range of othernode or a collection of nodes. Consider a wireless networking, each node at the far edge of theaccess point's range, which is known as A, can see the access point, but it is unlikely that thesame node can see a node on the opposite end of the access point's range, C. These nodes areknown as hidden. The problemis when nodes Aand C start to send packets simultaneously tothe access point B. Because the nodes A and C are out of range of eachother and so cannotdetectacollisionwhiletransmitting,Carriersensemultipleaccesswithcollisiondetection(CSMA/CD) does not work, and collisions occur, which then corrupt the data received by theaccesspoint.Toovercomethe hidden node problem,RTS/CTShandshaking(IEEE 802.11RTS/CTS) is implemented in conjunction with the Carrier sense multiple accesses with collisionavoidance(CSMA/CA)scheme. Thesameproblemexistsina MANET.

The transmission range of access point A reaches at B, but not at access point C, similarlytransmission range of access point C reaches B, but not at A. These nodes are known as hiddenterminals. The problem occurs when nodes A and C start to send data packets simultaneously tothe accesspoint B.BecausetheaccesspointsAand Care outof range of eachother andresultant they cannot detect a collision while transmitting, Carrier sense multiple access withcollision detection (CSMA/CD) does not work, and collisions occur, which then corrupt the datareceivedby theaccess pointB due tothehiddenterminalproblem.

Thehiddenterminalanalogyisdescribedasfollows:

- TerminalAsendsdatatoB,terminalCcannothearA
- TerminalCwantstosenddatatoB,terminalCsenses a"free"medium(CSfails)andstartstransmitting
- CollisionatBoccurs,Acannotdetectthiscollision(CDfails)andcontinueswithitstransmissionto B
- TerminalAis"hidden"fromCandviceversa.



Hidden Node Problem

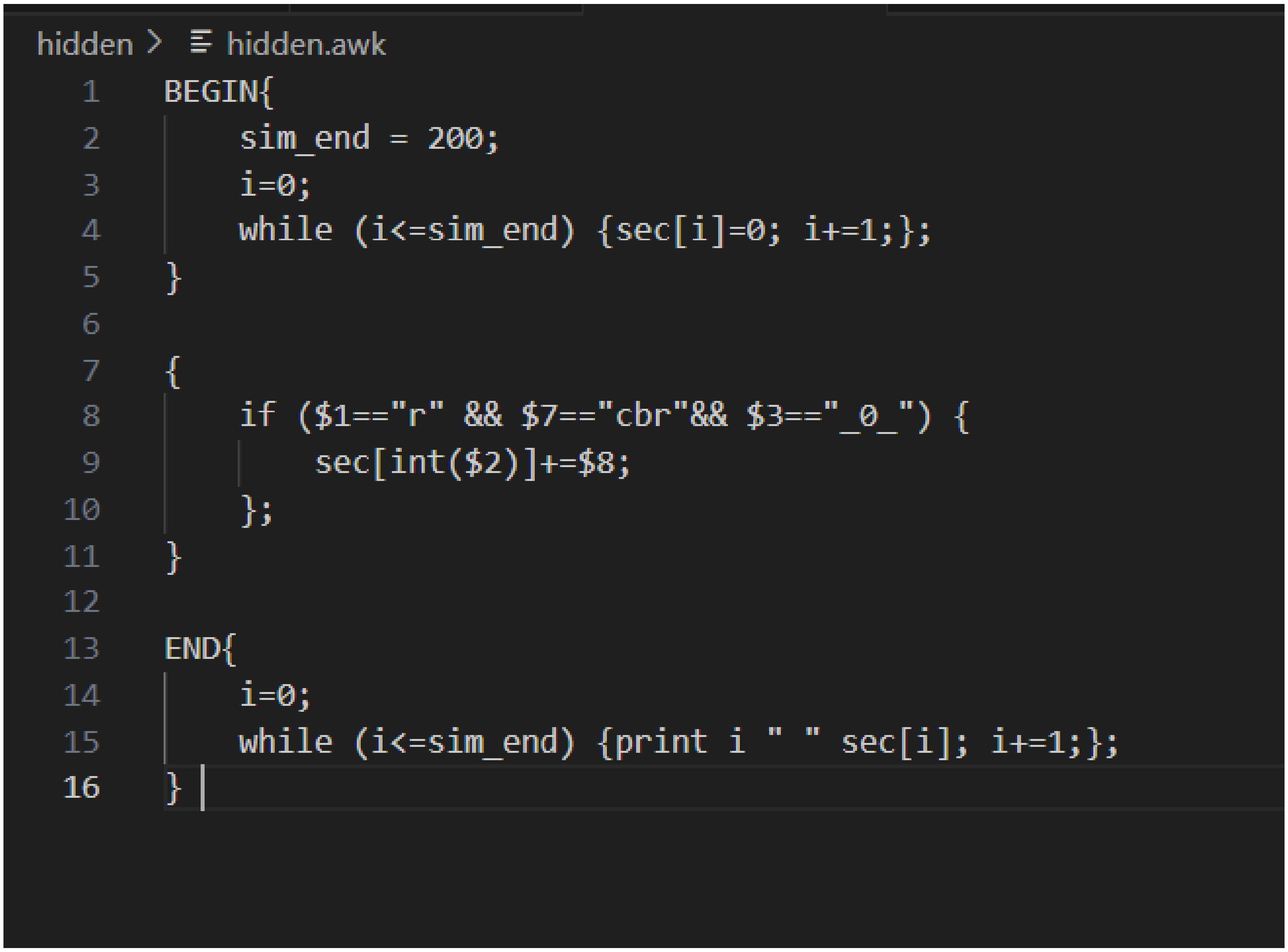
Thesolutionofhiddenterminalproblemisasfollows.

WhenAwantstosendapackettoB,AfirstsendsaRequest-to-send(RTS)toB.Onreceiving RTS,B respondsbysendingClear-to-Send(CTS).

WhenCoverhearsaCTS,itkeepsquietforthedurationofthetransfer.Transfe rdurationis includedin bothRTSandCTS.

RTSandCTSareshortframes,reducescollisionchance.

1. Hidden



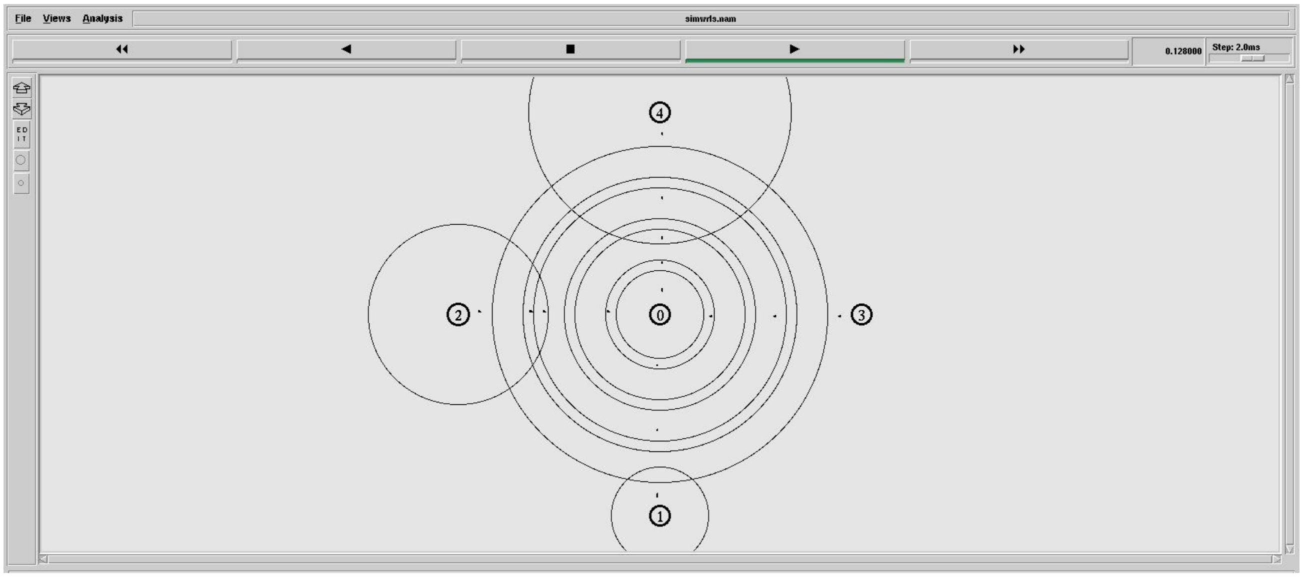
```
BEGIN{
    sim_end = 200;
    i=0;
    while (i<=sim_end) {sec[i]=0; i+=1;};
}

{
    if ($1=="r" && $7=="cbr"&& $3=="_0_") {
        sec[int($2)]+=$8;
    };
}

END{
    i=0;
    while (i<=sim_end) {print i " " sec[i]; i+=1;};
}
```

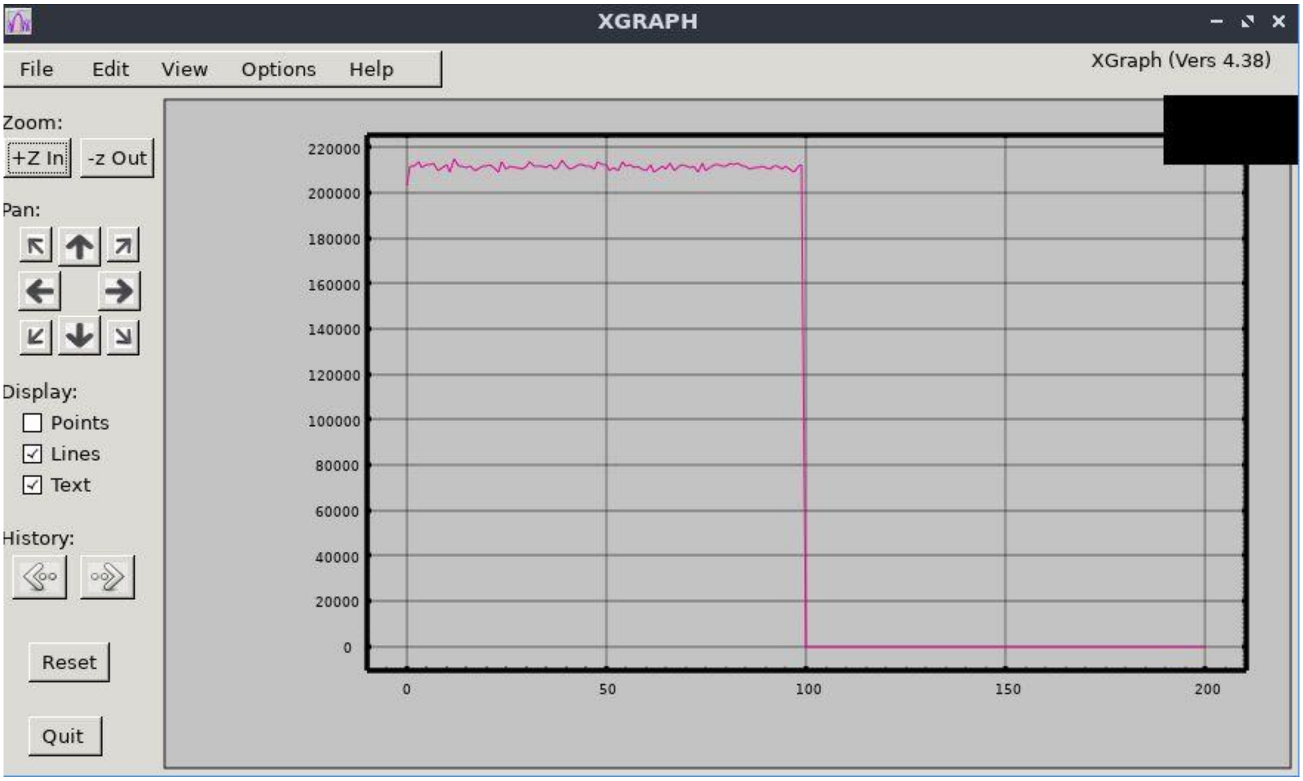
```
# Define options

set val(chan)         Channel/WirelessChannel    ;# channel type

set val(prop)         Propagation/FreeSpace   ;# 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)       10000                    ;# max packet in ifq

set val(nn)           5                        ;# number of mobilenodes

set val(rp)           DSR                      ;# routing protocol

set val(x)            600              ;# X dimension of topography

set val(y)            600            ;# Y dimension of topography
set val(stop)      100           ;# time of simulation end
set val(R)       300
set opt(tr)    out.tr
set ns       [new Simulator]
set tracefd  [open $opt(tr) w]
set windowVsTime2 [open win.tr w]
set namtrace      [open simwrls.nam w]
  Mac/802_11 set dataRate_        1.2e6
Mac/802_11 set RTSThreshold_    100
$ns trace-all $tracefd
#$ns use-newtrace
$ns namtrace-all-wireless $namtrace $val(x) $val(y)

# set up topography object
set topo       [new Topography]

$topo load_flatgrid $val(x) $val(y)

create-god $val(nn)

#
#  Create nn mobilenodes [$val(nn)] and attach them to the channel.
#

# configure the nodes

    $ns node-config -adhocRouting $val(rp) \
           -llType $val(ll) \
           -macType $val(mac) \
           -ifqType $val(ifq) \
           -ifqLen $val(ifqlen) \
           -antType $val(ant) \
           -propType $val(prop) \
           -phyType $val(netif) \
           -channelType $val(chan) \
           -topoInstance $topo \
           -agentTrace ON \
           -routerTrace ON \
           -macTrace ON \
           -movementTrace ON

Phy/WirelessPhy set CSThresh 30.5e-10
   for {set i 0} {$i < $val(nn) } { incr i } {
      set node_($i) [$ns node]
   }
$node_(0) set X_ $val(R)
$node_(0) set Y_ $val(R)
$node_(0) set Z_ 0
$node_(1) set X_ $val(R)
$node_(1) set Y_ 0
$node_(1) set Z_ 0
$node_(2) set X_ 0
$node_(2) set Y_ $val(R)
$node_(2) set Z_ 0
$node_(3) set X_ [expr $val(R) *2]
$node_(3) set Y_ $val(R)
$node_(3) set Z_ 0
$node_(4) set X_ $val(R)
$node_(4) set Y_ [expr $val(R) *2]
$node_(4) set Z_ 0
```

```
 for {set i 0} {$i<$val(nn)} {incr i} {
  $ns initial_node_pos $node_($i) 30
 }
# Generation of movements
 $ns at 0 "$node_(1) setdest $val(R) $val(R) 3.0"
$ns at 0 "$node_(2) setdest $val(R) $val(R) 3.0"
$ns at 0 "$node_(3) setdest $val(R) $val(R) 3.0"
$ns at 0 "$node_(4) setdest $val(R) $val(R) 3.0"
# Set a TCP connection between node_(0) and node_(1)
set tcp [new Agent/TCP/Newreno]
#$tcp set class_ 2
set tcp [new Agent/UDP]
$tcp set class_ 2
set sink [new Agent/Null]
$ns attach-agent $node_(1) $tcp
$ns attach-agent $node_(0) $sink
$ns connect $tcp $sink
set ftp [new Application/Traffic/CBR]
$ftp attach-agent $tcp
$ns at 0.0 "$ftp start"
# ################################################
# For coloring but doesnot work
# ################################################
$tcp set fid_ 1
$ns color 1 blue
#/////////////////////////////////////////////////
set tcp [new Agent/UDP]
$tcp set class_ 2
set sink [new Agent/Null]
$ns attach-agent $node_(2) $tcp
```

```
$ns attach-agent $node_(0) $sink
$ns connect $tcp $sink
set ftp [new Application/Traffic/CBR]
$ftp attach-agent $tcp
$ns at  0.0 "$ftp start"
set tcp [new Agent/UDP]
$tcp set class_ 2
set sink [new Agent/Null]
$ns attach-agent $node_(3) $tcp
$ns attach-agent $node_(0) $sink
$ns connect $tcp $sink
set ftp [new Application/Traffic/CBR]
$ftp attach-agent $tcp
$ns at  0.0 "$ftp start"
set tcp [new Agent/UDP]
$tcp set class_ 2
set sink [new Agent/Null]
$ns attach-agent $node_(4) $tcp
$ns attach-agent $node_(0) $sink
$ns connect $tcp $sink
set ftp [new Application/Traffic/CBR]
$ftp attach-agent $tcp
$ns at  0.0 "$ftp start"
# Telling nodes when the simulation ends
#for {set i 0} {$i < $val(nn) } { incr i } {
 #   $ns at $val(stop) "$node_($i) reset";
#}

# ending nam and the simulation
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
```

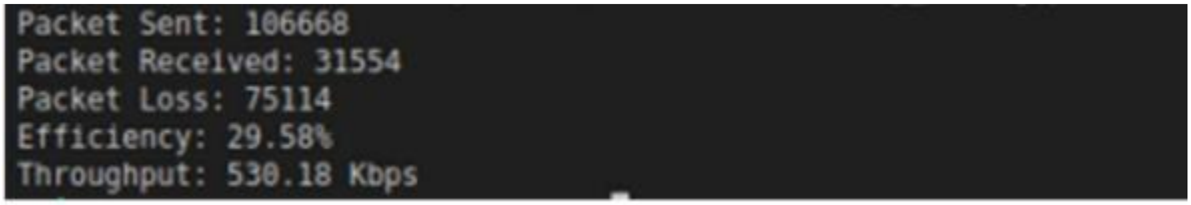
```
$ns at $val(stop) "stop"

$ns at $val(stop) "puts \"end simulation\" ; $ns halt"

proc stop {} {

 exec awk -f fil.awk out.tr > out.xgr

exec xgraph out.xgr &


    global ns tracefd namtrace

    $ns flush-trace

    close $tracefd

   close $namtrace

    exec nam simwrls.nam &

}


      $ns run
```



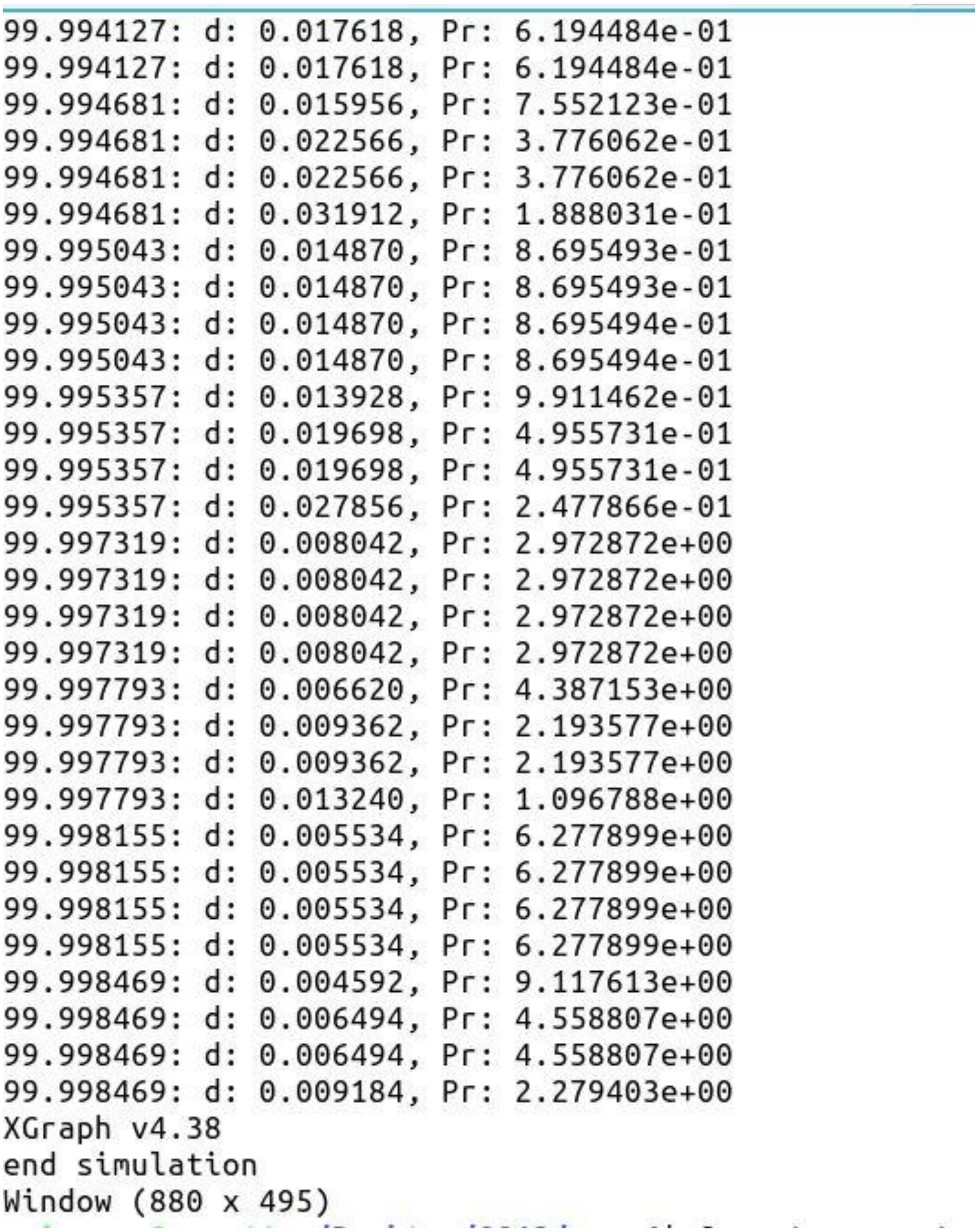


[redacted]

Efficiency and Throughput:

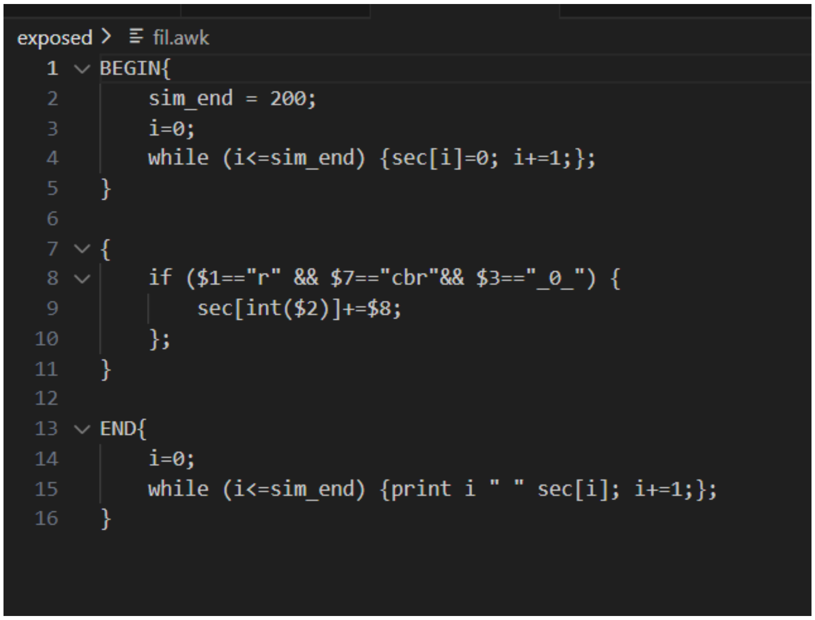


```
Packet Sent: 106668
Packet Received: 31554
Packet Loss: 75114
Efficiency: 29.58%
Throughput: 530.18 Kbps
```



```
99.994127: d: 0.017618, Pr: 6.194484e-01
99.994127: d: 0.017618, Pr: 6.194484e-01
99.994681: d: 0.015956, Pr: 7.552123e-01
99.994681: d: 0.022566, Pr: 3.776062e-01
99.994681: d: 0.022566, Pr: 3.776062e-01
99.994681: d: 0.031912, Pr: 1.888031e-01
99.995043: d: 0.014870, Pr: 8.695493e-01
99.995043: d: 0.014870, Pr: 8.695493e-01
99.995043: d: 0.014870, Pr: 8.695494e-01
99.995043: d: 0.014870, Pr: 8.695494e-01
99.995357: d: 0.013928, Pr: 9.911462e-01
99.995357: d: 0.019698, Pr: 4.955731e-01
99.995357: d: 0.019698, Pr: 4.955731e-01
99.995357: d: 0.027856, Pr: 2.477866e-01
99.997319: d: 0.008042, Pr: 2.972872e+00
99.997319: d: 0.008042, Pr: 2.972872e+00
99.997319: d: 0.008042, Pr: 2.972872e+00
99.997319: d: 0.008042, Pr: 2.972872e+00
99.997793: d: 0.006620, Pr: 4.387153e+00
99.997793: d: 0.009362, Pr: 2.193577e+00
99.997793: d: 0.009362, Pr: 2.193577e+00
99.997793: d: 0.013240, Pr: 1.096788e+00
99.998155: d: 0.005534, Pr: 6.277899e+00
99.998155: d: 0.005534, Pr: 6.277899e+00
99.998155: d: 0.005534, Pr: 6.277899e+00
99.998155: d: 0.005534, Pr: 6.277899e+00
99.998469: d: 0.004592, Pr: 9.117613e+00
99.998469: d: 0.006494, Pr: 4.558807e+00
99.998469: d: 0.006494, Pr: 4.558807e+00
99.998469: d: 0.009184, Pr: 2.279403e+00
XGraph v4.38
end simulation
Window (880 x 495)
```

2. Exposed



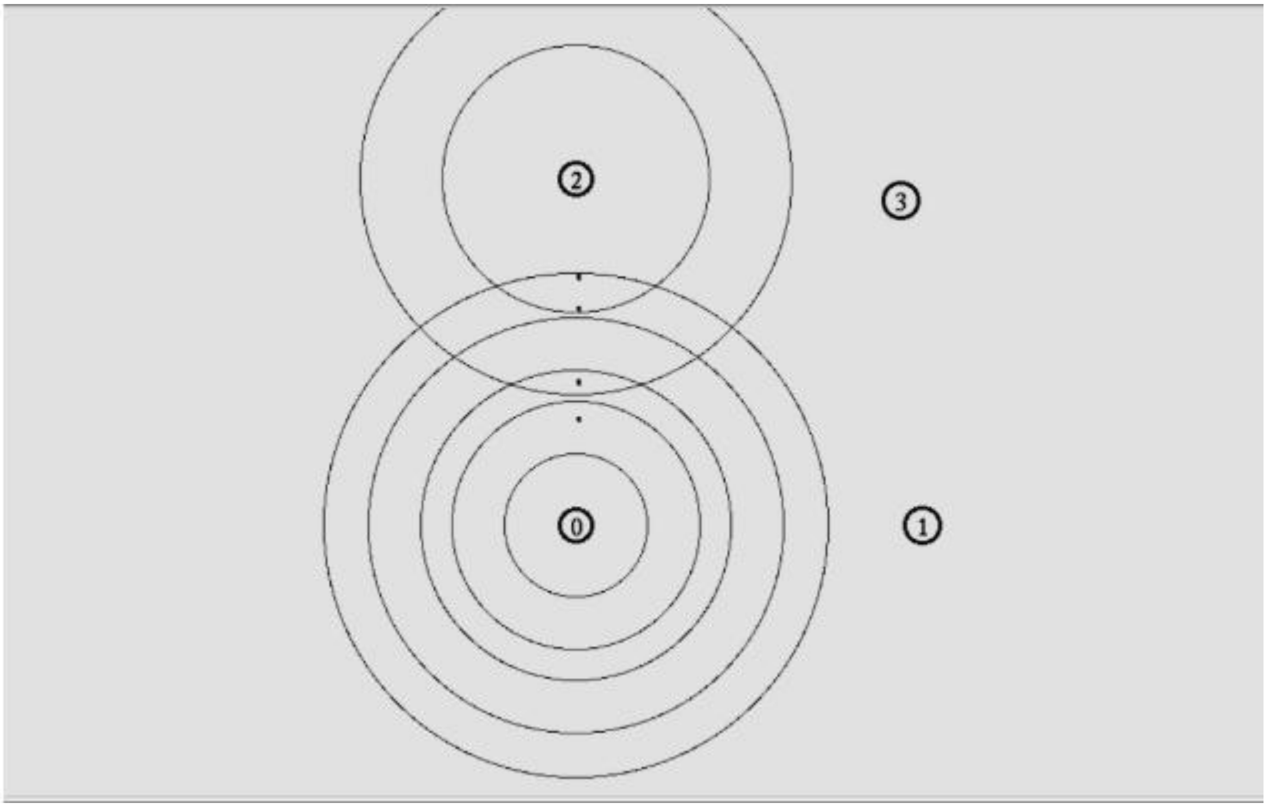
```
BEGIN{
    sim_end = 200;
    i=0;
    while (i<=sim_end) {sec[i]=0; i+=1;};
}

{
    if ($1=="r" && $7=="cbr"&& $3=="_0_") {
        sec[int($2)]+=$8;
    };
}

END{
    i=0;
    while (i<=sim_end) {print i " " sec[i]; i+=1;};
}
```

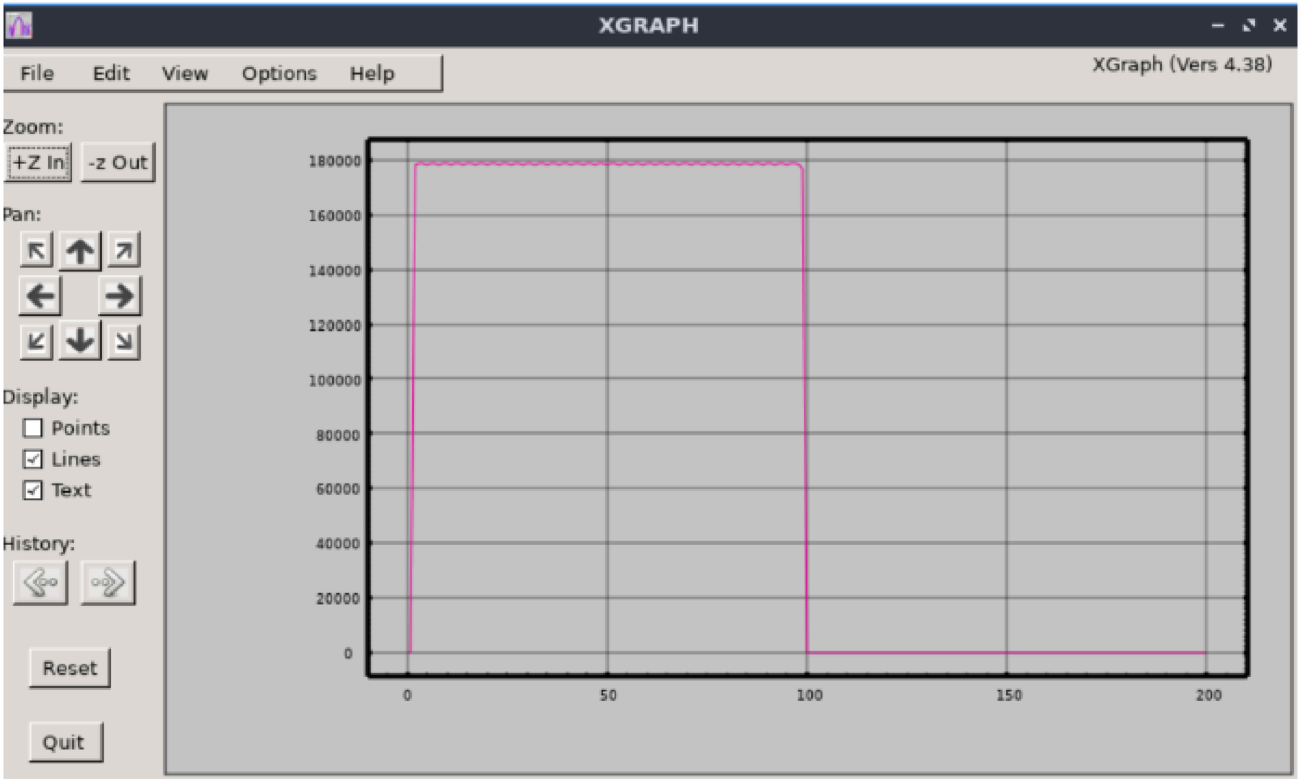
```
# Define options

set val(chan)         Channel/WirelessChannel    ;# channel type

set val(prop)         Propagation/FreeSpace      ;# 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)       10000                      ;# max packet in ifq

set val(nn)           4                          ;# number of mobilenodes

set val(rp)           DSR                        ;# routing protocol

set val(x)            600                        ;# X dimension of topography

set val(y)            600                        ;# Y dimension of topography

set val(stop)         100                        ;# time of simulation end

set val(R)            300                        ;# radius for node placement

set opt(tr)           out.tr

set ns                [new Simulator]

set tracefd           [open $opt(tr) w]

set namtrace          [open simwrls.nam w]

Mac/802_11 set dataRate_       1.2e6

Mac/802_11 set RTSThreshold_   100

$ns trace-all $tracefd

$ns namtrace-all-wireless $namtrace $val(x) $val(y)


# set up topography object

set topo              [new Topography]

$topo load_flatgrid $val(x) $val(y)


# Create God

create-god $val(nn)
```

```
# Create nn mobilenodes and attach them to the channel
# Configure nodes
$ns node-config -adhocRouting $val(rp) \
        -llType $val(ll) \
        -macType $val(mac) \
        -ifqType $val(ifq) \
        -ifqLen $val(ifqlen) \
        -antType $val(ant) \
        -propType $val(prop) \
        -phyType $val(netif) \
        -channelType $val(chan) \
        -topoInstance $topo \
        -agentTrace ON \
        -routerTrace ON \
        -macTrace ON \
        -movementTrace ON

Phy/WirelessPhy set CSThresh 30.5e-10
for {set i 0} {$i < $val(nn) } { incr i } {
    set node_($i) [$ns node]
}

# Set initial positions
$node_(0) set X_ $val(R)
$node_(0) set Y_ $val(R)
$node_(0) set Z_ 0
$node_(1) set X_ [expr $val(R)*2]
$node_(1) set Y_ $val(R)
$node_(1) set Z_ 0
```

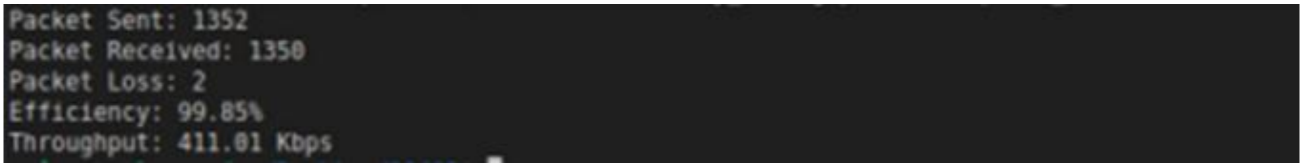
```
$node_(2) set X_ $val(R)
$node_(2) set Y_ [expr $val(R)*2]
$node_(2) set Z_ 0
$node_(3) set X_ [expr $val(R)*2]
$node_(3) set Y_ [expr $val(R)*2]
$node_(3) set Z_ 0

# Setup initial positions for all nodes
for {set i 0} {$i < $val(nn)} {incr i} {
    $ns initial_node_pos $node_($i) 30
}

# Exposed Terminal Problem simulation
# Node (0) communicates with Node (2) and Node (1) communicates with Node (3).
# Nodes (0) and (1) are not in range of each other but are both in range of node (2).

# Node 0 and Node 1 transmit to Node 2 and Node 3 respectively causing interference at Node 2.
$ns at 1.0 "$node_(0) setdest $val(R) $val(R) 5.0"
$ns at 1.0 "$node_(1) setdest $val(R)*2 $val(R) 5.0"
$ns at 1.0 "$node_(2) setdest $val(R) $val(R) 3.0"
$ns at 1.0 "$node_(3) setdest $val(R)*2 $val(R)*2 3.0"

# Set up TCP connections for communication
# Node 0 sends to Node 2
set tcp0 [new Agent/TCP/Newreno]
set sink0 [new Agent/Null]
$ns attach-agent $node_(0) $tcp0
$ns attach-agent $node_(2) $sink0
$ns connect $tcp0 $sink0
set ftp0 [new Application/Traffic/CBR]
```

```
$ftp0 attach-agent $tcp0
$ns at 2.0 "$ftp0 start"

# Node 1 sends to Node 3
set tcp1 [new Agent/TCP/Newreno]
set sink1 [new Agent/Null]
$ns attach-agent $node_(1) $tcp1
$ns attach-agent $node_(3) $sink1
$ns connect $tcp1 $sink1
set ftp1 [new Application/Traffic/CBR]
$ftp1 attach-agent $tcp1
$ns at 2.0 "$ftp1 start"

# Set a TCP connection between node_(0) and node_(1)
set tcp2 [new Agent/UDP]
set sink2 [new Agent/Null]
$ns attach-agent $node_(2) $tcp2
$ns attach-agent $node_(0) $sink2
$ns connect $tcp2 $sink2
set ftp2 [new Application/Traffic/CBR]
$ftp2 attach-agent $tcp2
$ns at 2.0 "$ftp2 start"

# End the simulation
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "stop"
$ns at $val(stop) "puts \"end simulation\" ; $ns halt"

# Function to process the end of the simulation
proc stop {} {
    exec awk -f fil.awk out.tr > out.xgr

    exec xgraph out.xgr &


    global ns tracefd namtrace

    $ns flush-trace

    close $tracefd

    close $namtrace

    exec nam simwrls.nam &

}


# Run the simulation

$ns run
```



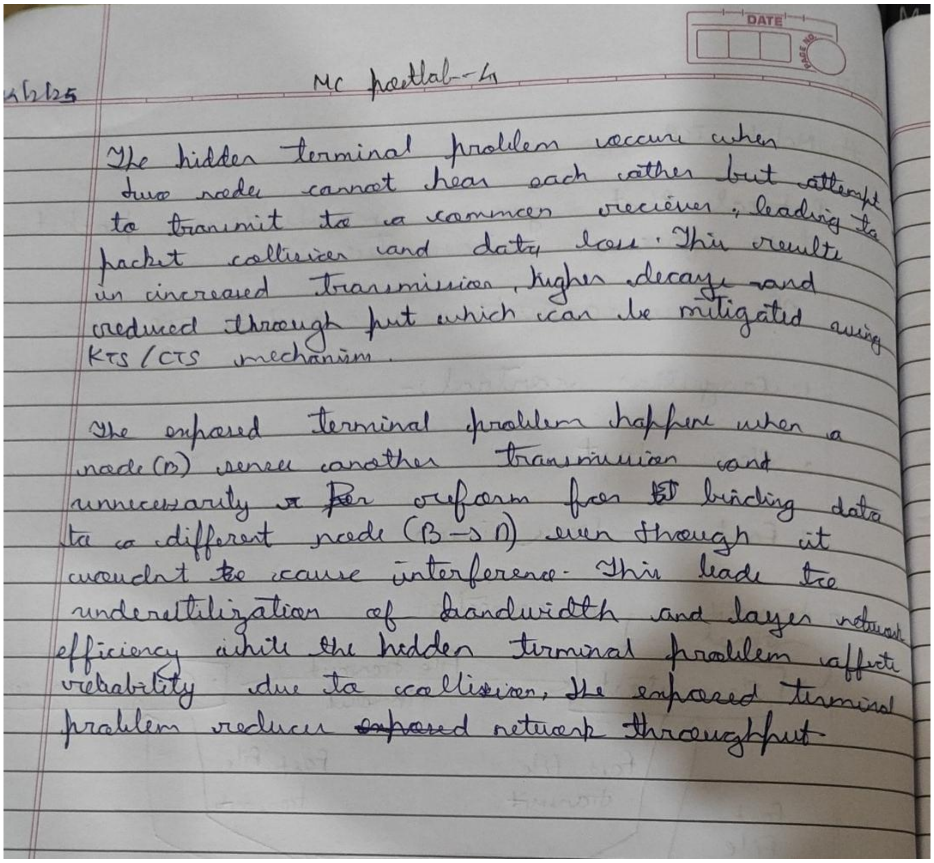


Efficiency and throughput:



```
Packet Sent: 1352
Packet Received: 1350
Packet Loss: 2
Efficiency: 99.85%
Throughput: 411.01 Kbps
```

Postlab:



4/2/25

MC postlab-4

The hidden terminal problem occurs when two nodes cannot hear each other but attempt to transmit to a commen reciever, leading to packet collision and data loss. This results in increased transmission, higher decay and reduced through put which can be mitigated using RTS/CTS mechanism.

The exposed terminal problem happens when a node (B) senses another transmission and unnecessarily ~~Per~~ reform from sending data to a different node (B -> D) even though it wouldnt ~~be~~ cause interference. This leads to underutilization of bandwidth and layer network efficiency while the hidden terminal problem affects reliability due to collision, the exposed terminal problem reduces ~~exposed~~ network throughput

Conclusion:

The hidden terminal problem severely impacts network performance by leading to frequent collisions and retransmissions. Implementing RTS/CTS or alternative MAC protocols can help alleviate this issue, enhancing the reliability and efficiency of wireless communication.