# Assignment No.

**Lab Assignment on Unit IV and Unit V:**
Use network simulator NS2 to implement:
a. Wired network of 4 nodes.
b. Wireless network with AoDV

Date:-

Marks:- Signature:-

## Title: Lab Assignment on Unit IV and Unit V:

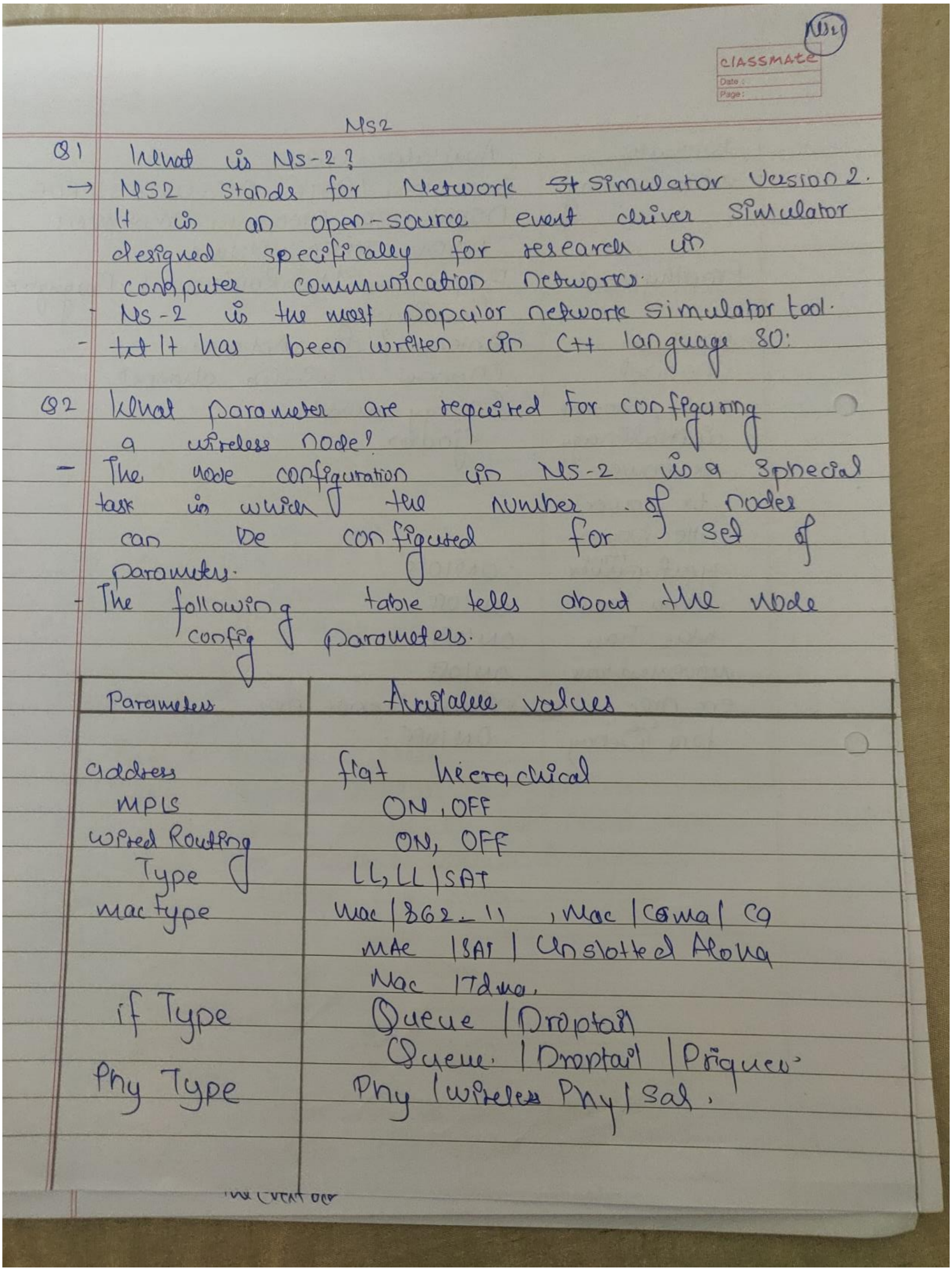
Use network simulator NS-2 to implement:
a. Wired network of 4 nodes.
b. Wireless network with AoDV

```
# A 3-node example for ad-hoc simulation with AODV

# Define options
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    ;# 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)            3                          ;# number of mobilenodes
set val(rp)            AODV                       ;# routing protocol
set val(x)             500                        ;# X dimension of topography
set val(y)             400                        ;# Y dimension of topography
set val(stop)          150                        ;# time of simulation end

set ns              [new Simulator]
set tracefd         [open simple-dsdv.tr w]
set windowVsTime2 [open win.tr w]
set namtrace       [open simwrls1.nam w]

$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 OFF \
                        -movementTrace ON

                for {set i 0} {$i < $val(nn) } { incr i } {
                    set node_($i) [$ns node]
                }

# Provide initial location of mobilenodes
$node_(0) set X_ 5.0
$node_(0) set Y_ 5.0
$node_(0) set Z_ 0.0

$node_(1) set X_ 490.0
$node_(1) set Y_ 285.0
$node_(1) set Z_ 0.0

$node_(2) set X_ 150.0
$node_(2) set Y_ 240.0
$node_(2) set Z_ 0.0

# Generation of movements
$ns at 10.0 "$node_(0) setdest 250.0 250.0 3.0"
$ns at 15.0 "$node_(1) setdest 45.0 285.0 5.0"
$ns at 110.0 "$node_(0) setdest 480.0 300.0 5.0"

# Set a TCP connection between node_(0) and node_(1)
set tcp [new Agent/TCP/Newreno]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $node_(0) $tcp
$ns attach-agent $node_(1) $sink
$ns connect $tcp $sink
```

```
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 10.0 "$ftp start"




# Define node initial position in nam
for {set i 0} {$i < $val(nn)} { incr i } {
# 30 defines the node size for nam
$ns initial_node_pos $node_($i) 30
}

# 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 150.01 "puts \"end simulation\" ; $ns halt"
proc stop {} {
   global ns tracefd namtrace
   $ns flush-trace
   close $tracefd
   close $namtrace
   exec nam simwrls1.nam &
}

$ns run
```

```
set ns [new Simulator]
#create file for analysis mode
set tr [open out.tr w]
$ns trace-all $tr
#create file for Animation Mode
set namtr [open out.nam w]
$ns namtrace-all $namtr
#Create Node
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
#Create Link
$ns duplex-link $n0 $n1 10Mb 5ms DropTail
$ns duplex-link $n2 $n0 10Mb 5ms DropTail
$ns duplex-link $n3 $n0 10mb 5ms DropTail
#Create Orientation
$ns duplex-link-op $n0 $n1 orient right
$ns duplex-link-op $n0 $n2 orient left-up
$ns duplex-link-op $n0 $n3 orient left-down
#create UDP Source
set udp0 [new Agent/UDP]
$ns attach-agent $n3 $udp0
#create UDP Destination
set null0 [new Agent/Null]
$ns attach-agent $n1 $null0
#connecting UDP Source & Destination
$ns connect $udp0 $null0
#create application traffic
set cbr0 [new Application/Traffic/CBR]
$cbr0 attach-agent $udp0
#Application start time
$ns at 1.0 "$cbr0 start"
#Application Stop time
$ns at 5.0 "$cbr0 stop"
#create TCP Source
set tcp0 [new Agent/TCP]
$ns attach-agent $n2 $tcp0
#create TCP Destination
set sink0 [new Agent/TCPSink]
```

```
$ns attach-agent $n1 $sink0
$ns connect $tcp0 $sink0
#create application traffic
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
#Application start & stop time
$ns at 2.0 "$ftp0 start"
$ns at 5.0 "$ftp0 stop"

$ns at 10.0 "$ns halt"
$ns run
```



classmate
Date :
Page :

NS2

Q1 What is NS-2?

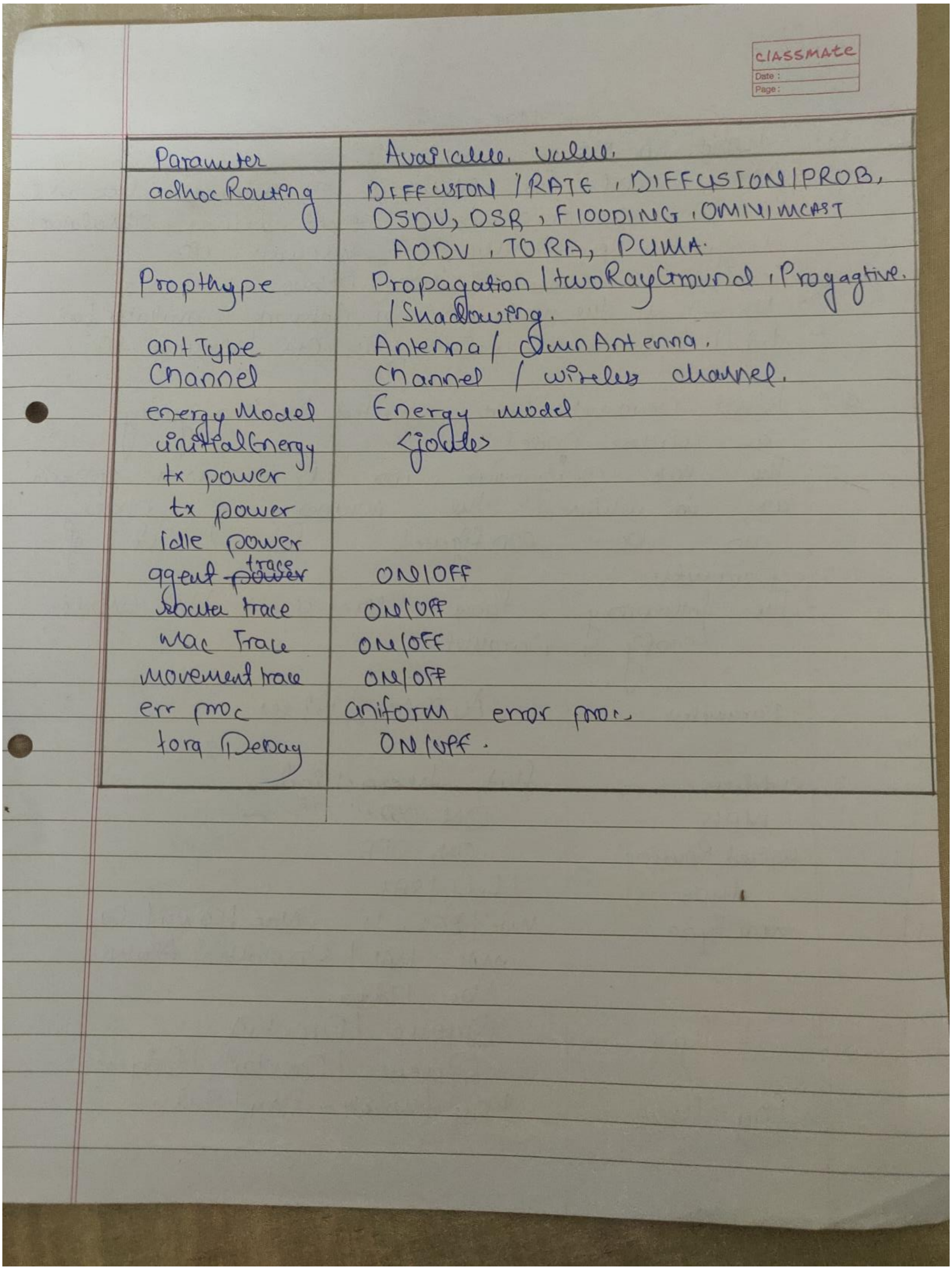
→ NS2 stands for Network ~~St~~ Simulator Version 2. It is an open-source event driver simulator designed specifically for research in computer communication networks.

- NS-2 is the most popular network simulator tool.
- ~~txt~~ It has been written in C++ language so:

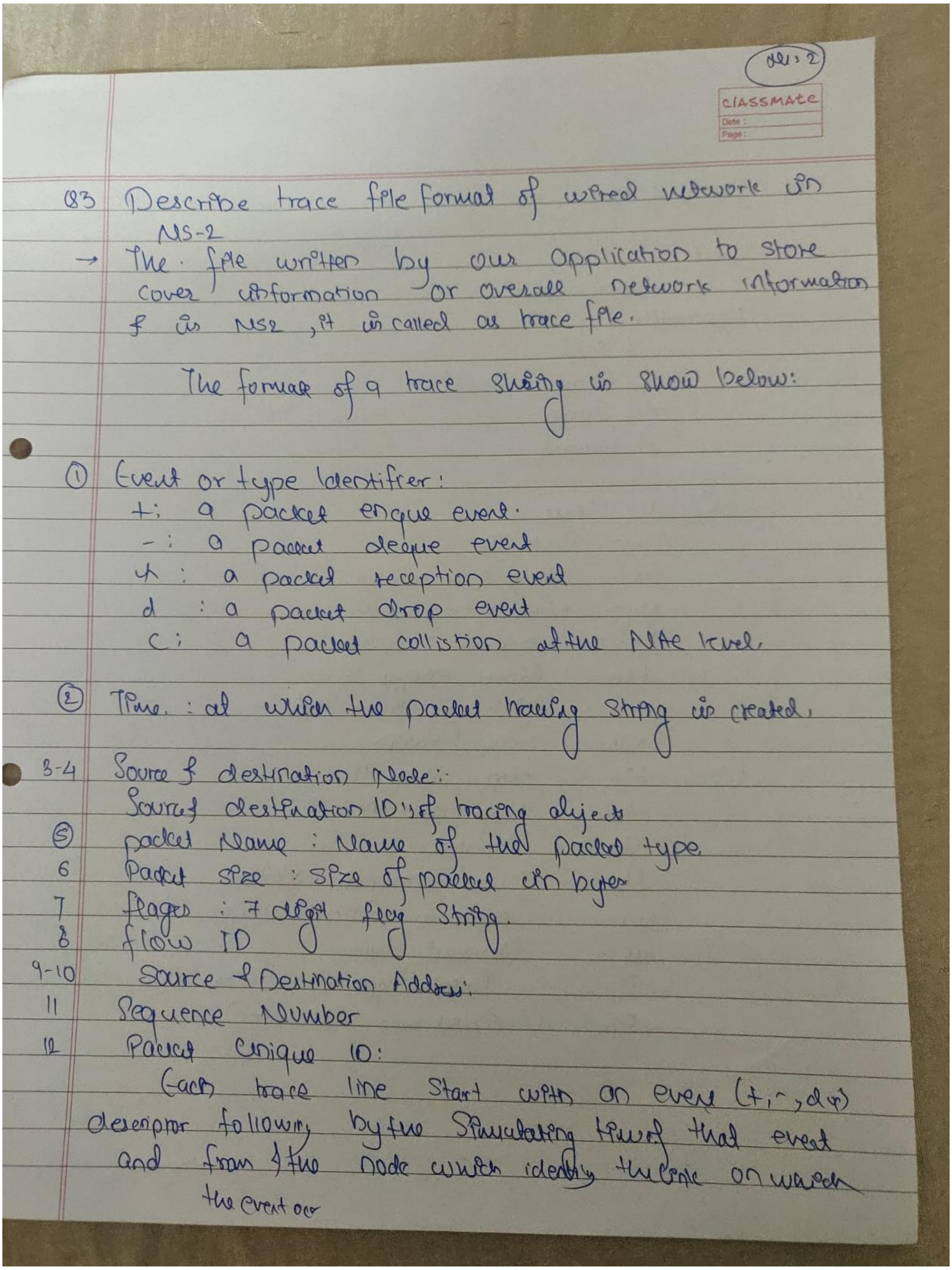
Q2 What parameter are required for configuring a wireless node?

- The node configuration in NS-2 is a special task in which the number of nodes can be configured for set of parameters.
- The following table tells about the node config parameters.

| Parameters | Available values |
|---|---|
| address | flat hierachical |
| MPLS | ON, OFF |
| wired Routing | ON, OFF |
| Type | LL, LL\|SAT |
| mac type | Mac\|802_11, Mac\|Csma\|Ca<br>MAC\|SAT\|Unslotted Aloha<br>Mac\|Tdma. |
| if Type | Queue\|Droptail<br>Queue\|Droptail\|Priqueu |
| Phy Type | Phy\|wireless Phy\|Sat. |



| Parameter | Available value. |
|---|---|
| adhoc Routing | DIFFUSION /RATE , DIFFUSION/PROB,<br>DSDV, DSR , FlOODING , OMINIMCAST<br>AODV , TORA, PUMA. |
| Propthype | Propagation / twoRayGround , Progagtive.<br>/ Shadowing. |
| ant Type | Antenna/ OmnAntenna. |
| Channel | Channel / wireless channel. |
| energy Model | Energy model |
| initialEnergy | <joule> |
| tx power | |
| tx power | |
| idle power | |
| agent ~~power~~ trace | ON/OFF |
| Router trace | ON/OFF |
| Mac Trace | ON/OFF |
| movement trace | ON/OFF |
| err proc | uniform error proc. |
| tora Delay | ON/OFF. |



As: 2

Q3 Describe trace file format of wired network in NS-2

→ The file written by our application to store cover information or overall network information f is NS2, it is called as trace file.

The format of a trace string is show below:

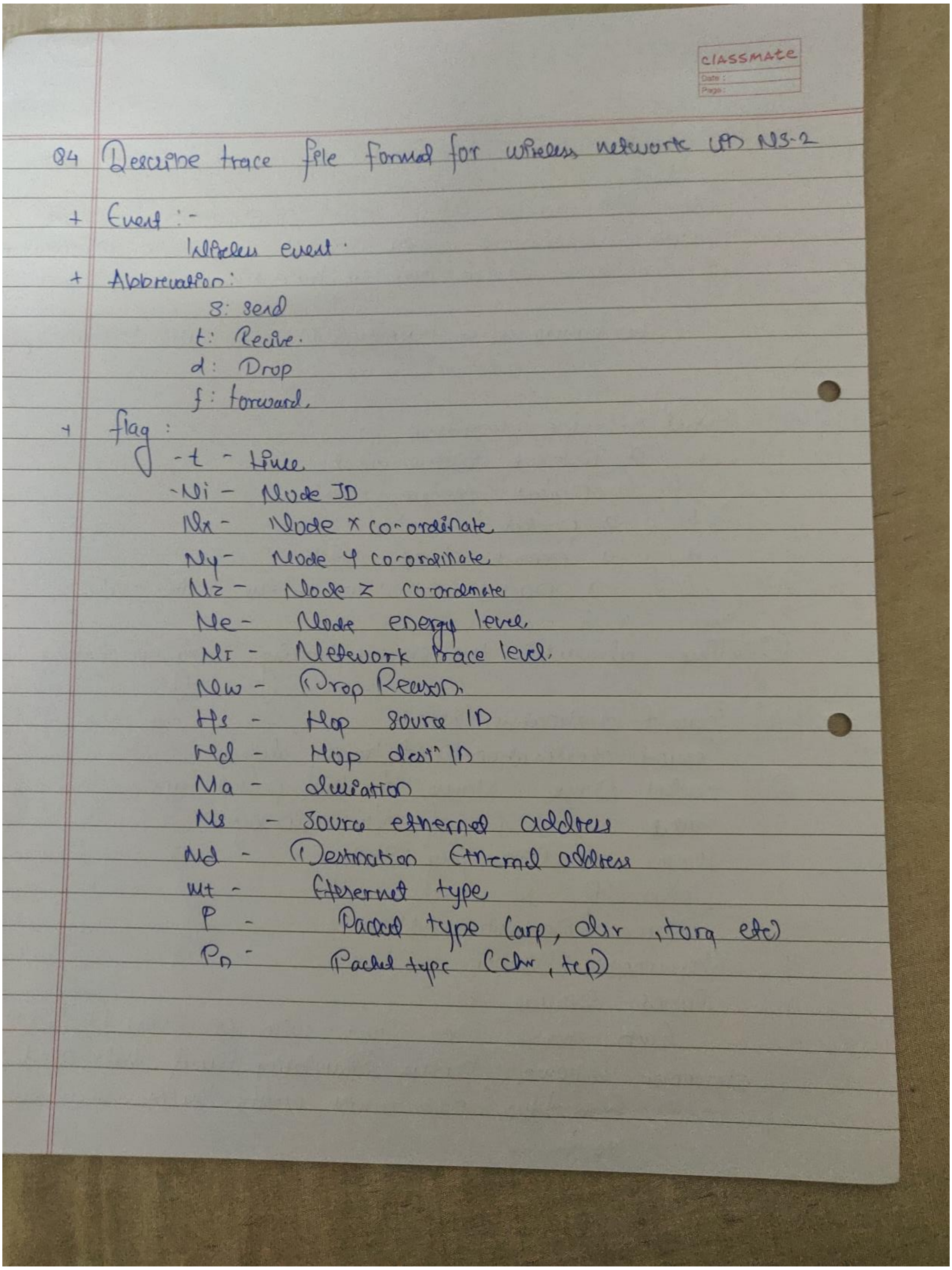
① Event or type Identifier:
+: a packet enque event.
-: a packet deque event
r: a packet reception event
d : a packet drop event
c: a packet collistion at the MAC level.

② Time : at which the packet having string is created.

3-4 Source f destination Node:
Source f destination ID's of tracing object

⑤ packet Name : Name of the packet type.
6 Packet size : size of packet in byes
7 flages : 7 digit flag String.
8 flow ID
9-10 Source & Destination Address:
11 Sequence Number
12 Packet unique ID:

Each trace line Start with an event (+, -, d r) descriptor following by the Simulating time of that event and from & the node which identify the link on which the event occr



Q4 Describe trace file formad for wireless network in NS-2

+ Event :-

  Wireless event.

+ Abbrevation:

  s: send

  r: Recive.

  d: Drop

  f: forward.

+ flag :

  -t - time

  -Ni - Node ID

  Nx - Node x co-ordinate.

  Ny - Node y co-ordinate.

  Nz - Node z co-ordinate

  Ne - Node energy level.

  Nl - Network trace level.

  Nw - Drop Reason.

  Hs - Hop source ID

  Hd - Hop destn ID

  Ma - duration

  Ms - Source ethernet address

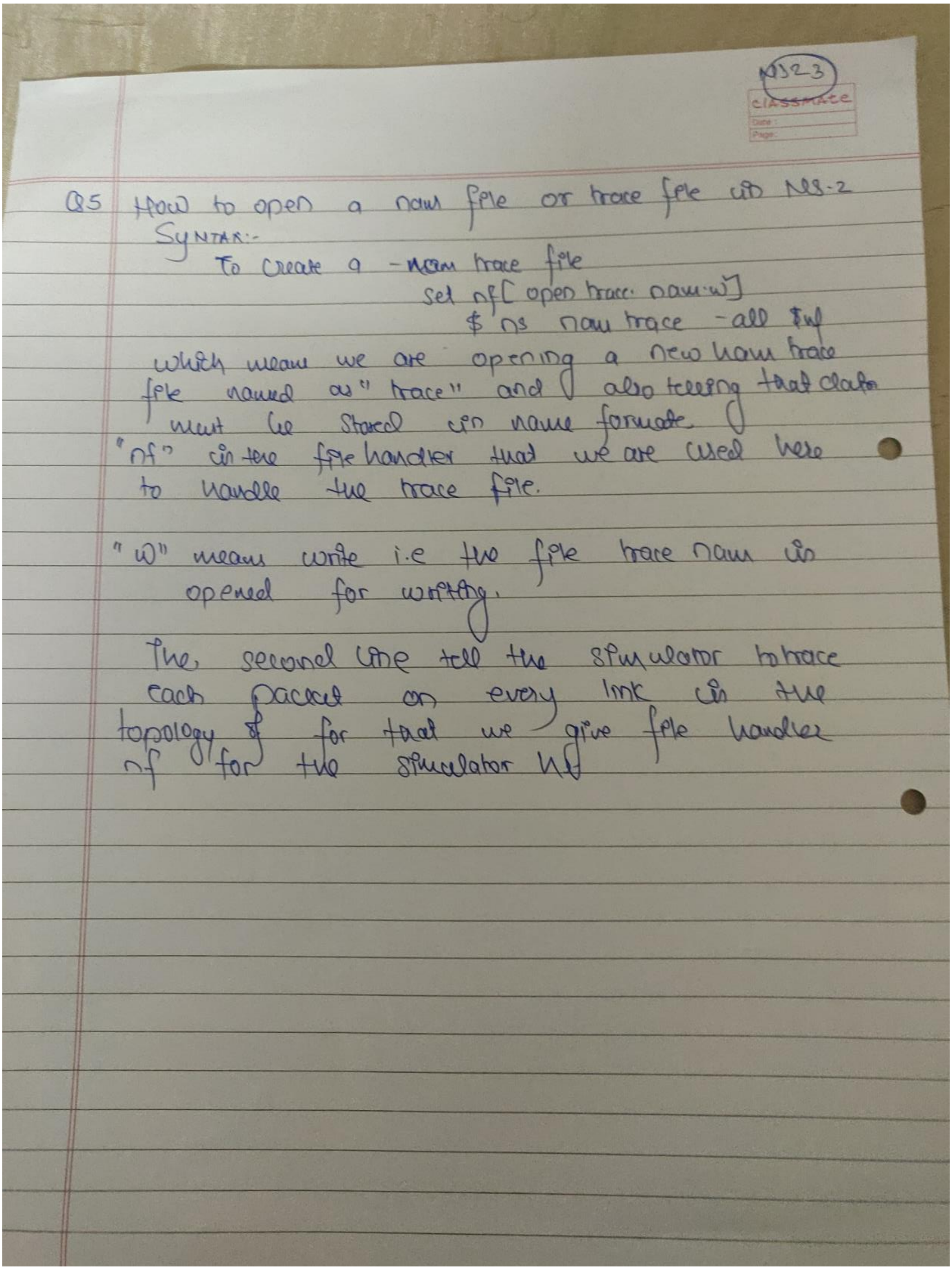
  Md - Destination Ethernet address

  Mt - Ethernet type

  P - Packet type (arp, dsr, tora etc)

  Pn - Packet type (cbr, tcp)

**Pushkaraj P Chavan TEC58**



NS23

Q5 How to open a nam file or trace file in NS-2

Syntax:-

To create a nam trace file

```
set nf [open trace.nam w]
$ns namtrace-all $nf
```

which mean we are opening a new nam trace file named as "trace" and also telling that data must be stored in nam format.

"nf" is the file handler that we are used here to handle the trace file.

"w" means write i.e the file trace nam is opened for writing.

The second line tell the simulator to trace each packet on every link in the topology for that we give file handler nf for the simulator ns.