DCCN LAB 10

Q1.
TCL script:

```
set ns [new Simulator -multicast on]
$ns color 1 Blue
$ns color 2 Red
set tracefile1 [open out.tr w]
$ns trace-all $tracefile1
set namfile [open out.nam w]
$ns namtrace-all $namfile

proc finish {} {
    global ns tracefile1 namfile
    $ns flush-trace
    close $tracefile1
    close $namfile
    exec nam out.nam &
    exit 0
}

set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
set n6 [$ns node]
set n7 [$ns node]
set n8 [$ns node]

$ns duplex-link $n0 $n1 5Mb 2ms DropTail
$ns duplex-link $n1 $n2 5Mb 1ms DropTail
$ns duplex-link $n2 $n3 5Mb 1ms DropTail
$ns duplex-link $n0 $n4 5Mb 2ms DropTail
$ns duplex-link $n4 $n3 5Mb 2ms DropTail
$ns duplex-link $n3 $n5 5Mb 2ms DropTail
$ns duplex-link $n5 $n6 5Mb 2ms DropTail
$ns duplex-link $n5 $n7 5Mb 2ms DropTail
$ns duplex-link $n6 $n8 5Mb 2ms DropTail
$ns duplex-link $n7 $n8 5Mb 2ms DropTail

$ns duplex-link-op $n0 $n1 orient right-up
```

DCCN LAB 10

```
$ns duplex-link-op $n0 $n1 orient right-up
$ns duplex-link-op $n1 $n2 orient right
$ns duplex-link-op $n2 $n3 orient right
$ns duplex-link-op $n4 $n3 orient right-up
$ns duplex-link-op $n3 $n5 orient right
$ns duplex-link-op $n5 $n6 orient right-up
$ns duplex-link-op $n5 $n7 orient right-down
$ns duplex-link-op $n6 $n8 orient right-down
$ns duplex-link-op $n7 $n8 orient right-up

set mproto DM
set mrthandle [$ns mrtproto $mproto {}]

set tcp [new Agent/TCP]
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n8 $sink
$ns connect $tcp $sink
$tcp set fid_ 1

set ftp [new Application/FTP]
$ftp attach-agent $tcp

set grp0 [Node allocaddr]

set udp [new Agent/UDP]
$ns attach-agent $n0 $udp
$udp set fid_ 2
$udp set dst_addr_ $grp0
$udp set dst_port_ 0

set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp

set rcvr1 [new Agent/LossMonitor]
set rcvr2 [new Agent/LossMonitor]
set rcvr3 [new Agent/LossMonitor]

$ns attach-agent $n6 $rcvr1
$ns attach-agent $n7 $rcvr2
$ns attach-agent $n8 $rcvr3
```

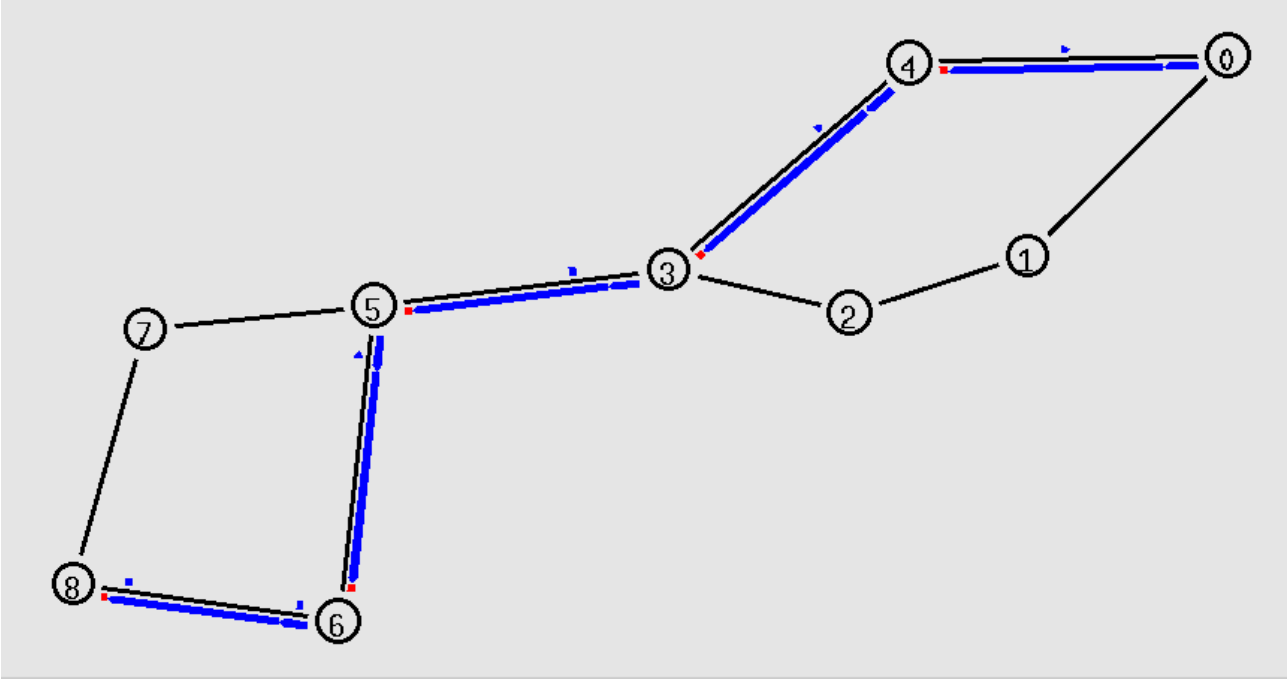
DCCN LAB 10

```
$ns at 0.3 "$n6 join-group $rcvr1 $grp0"
$ns at 0.3 "$n7 join-group $rcvr2 $grp0"
$ns at 0.3 "$n8 join-group $rcvr3 $grp0"

$ns at 0.1 "$ftp start"
$ns at 0.1 "$cbr start"
$ns at 89.5 "$ftp stop"
$ns at 89.5 "$cbr stop"

$ns at 90.0 "finish"
$ns run
```

Output:



a. Find the maximum time taken by packet to reach at node 8 from source node, foreach type of traffic.

AWK script:

```
BEGIN {
        max_time_ftp = 0;
        max_time_cbr = 0;
}
{
    if($5=="tcp"){
        if($1=="r" && $4=="8"){
```

DCCN LAB 10

```
                time_taken = $2 - $10;
                if(time_taken>max_time_ftp) max_time_ftp =
time_taken;
        }
    }
}
{
    if($1=="+" && $4=="8"){
        time_taken = $2-$10;
        if(time_taken>max_time_cbr) max_time_cbr =
time_taken;
    }
}
END{
    print("Max time taken by ftp packets: ",max_time_ftp);
    print("Max time taken by cbr packets: ",max_time_cbr);
}
```

Output:

```
Max time taken by ftp packets:  81.4247
Max time taken by cbr packets:  81.421
nit@nit-HP-EliteDesk-800-G1-SFF:~/120CS0124/Lab 10 23 Mar$
```

b. Find the number of packets received at node 8, for each type of traffic.

AWK script:

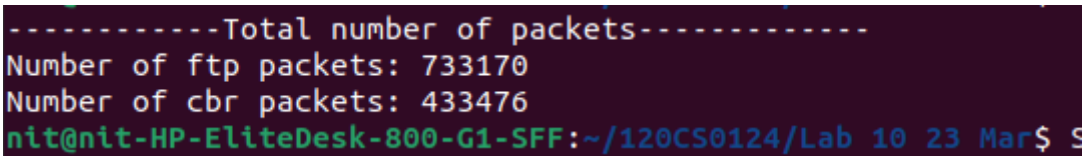
```
BEGIN {
    cbr=0;
    ftp=0;
}

{
    if($5=="tcp"){
        ftp=ftp+1;
    }
    if($5=="cbr"){
        cbr=cbr+1;
    }
}
```

DCCN LAB 10

```
END {
	printf("------------Total number of packets------------\n");
	printf("Number of ftp packets: %d\n",ftp);
	printf("Number of cbr packets: %d\n",cbr);
}
```

Output:



Q2.

AWK script:

```
BEGIN {
	recv=0;
	gotime = 1;
	time = 0;
	time_interval=1;
}
#body
{
	event = $1
		time = $2
		node_id = $3
		level = $4
		pktType = $7
	 packet_size = $8;

 if(time>gotime) {

  print gotime, (packet_size * recv * 8.0)/1000; #packet size * ...
gives results in kbps
  gotime+= time_interval;
  recv=0;
  }

#=========Calculate throughput==========

if (( event == "r") && ( pktType == "tcp" ) && ( level=="AGT" ))
{
```

DCCN LAB 10

```
 recv++;
}

} #body


END {
;
}
```

Output: