```

#1.....................................................................

#duplex-link  n0-n2, n1-n3
#TCP agent b/w n0-n3
#UDP agent b/w n1-n3
#applications over TCP and UDP agents
#queue size to 5
#vary the bandwidth to find the no. of packets dropped and recieved by TCP/UDP
#using awk script and grep command

set ns [new Simulator]
set tf [open ex1.tr w]
set nf [open ex1.nam w]
$ns trace-all $tf
$ns namtrace-all $nf

set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

$n0 label "TCP Source"
$n3 label "TCP Sink"
$n1 label "UDP Source"
$n3 label "UDP Null"

$ns duplex-link $n0 $n2 2Mb 2ms DropTail
$ns duplex-link $n1 $n2 2Mb 2ms DropTail
$ns duplex-link $n2 $n3 0.4Mb 10ms DropTail
$ns queue-limit $n0 $n2 5

set tcp [new Agent/TCP]
set sink [new Agent/TCPSink]
set ftp [new Application/FTP]
$ns attach-agent $n0 $tcp
$ns attach-agent $n3 $sink
$ns connect $tcp $sink
$ftp attach-agent $tcp

set udp [new Agent/UDP]
set null [new Agent/Null]
set cbr [new Application/Traffic/CBR]
$ns attach-agent $n1 $udp
$ns attach-agent $n3 $null
$ns connect $udp $null
$cbr attach-agent $udp

$ns at 0.1 "$ftp start"
$ns at 1.1 "$cbr start"
$ns at 10.0 "finish"

proc finish {} {
    global ns tf nf
    $ns flush-trace
    close $tf
    close $nf
    puts "running nam..."
    exec nam ex1.nam &
    exit 0
}

$ns run

#awk -f ex1.awk ex1.tr

BEGIN {
    tcp_d = 0;
    tcp_r = 0;
    udp_d = 0;
    udp_r = 0;
}
{
    if($1 == "d" && $5 == "tcp")
        tcp_d++;
    if($1 == "r" && $5 == "tcp")
        tcp_r++;
    if($1 == "d" && $5 == "cbr")
        udp_d++;
    if($1 == "r" && $5 == "cbr")
        udp_r++;
}
END{
    printf("TCP: No. of packets: recieved = %d, dropped = %d", tcp_r, tcp_d);
    printf("UDP: No. of packets: recieved = %d, dropped = %d", udp_r, udp_d);
}


#2.............................................................................

#FTP b/w the nodes n1-n6
#Telnet b/w nodes n2-n5.
#congestion window
#throughput

set ns [new Simulator]
set tf [open ex2.tr w]
set nf [open ex2.nam w]
set cwind [open win2.tr w]
$ns trace-all $tf
$ns namtrace-all $nf

$ns color 1 Blue
$ns color 2 Red

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

$n1 label "FTP Source"
$n6 label "FTP Sink"
$n2 label "Telnet Source"
$n5 label "Telnet Sink"

$ns duplex-link $n1 $n3 2Mb 2ms DropTail
$ns duplex-link $n2 $n3 2Mb 2ms DropTail
$ns duplex-link $n3 $n4 0.4Mb 5ms DropTail
$ns duplex-link $n4 $n5 2Mb 2ms DropTail
$ns duplex-link $n4 $n6 2Mb 2ms DropTail

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

set tcp1 [new Agent/TCP]
set sink1 [new Agent/TCPSink]
set ftp1 [new Applicaion/FTP]
$ns attach-agent $n1 $tcp1
$ns attach-agent $n6 $sink1
$ns connect $tcp1 $sink1
$ftp1 attach-agent $tcp1

set tcp2 [new Agent/TCP]
set sink2 [new Agent/TCPSink]
set telnet1 [new Application/Telnet]
$ns attach-agent $n2 $tcp2
$ns attach-agent $n6 $sink2
$ns connect $tcp2 $sink2
$telnet1 attach-agent $tcp2

$tcp1 set fid_ 1
```

```
$tcp2 set fid_ 2

$ns at 1.2 "$ftp1 start"
$ns at 5.0 "$ftp1 stop"
$ns at 5.1 "telnet1 start"

proc plotWindow {tcpSource file} {
    global ns
    set time 0.01
    set now[$ns now]
    set cwnd [$tcpSource set cwnd_]
    puts $file "$now $cwnd"
    $ns at [expr $now + $time] "plotWindow $tcpSource $file"
}

$ns at 2.0 "plotWindow $tcp1 $cwind"
$ns at 5.5 "plotWindow $tcp2 $cwind"

proc finish {} {
    global ns tf nf cwind
    $ns flush-trace
    close $tf
    close $nf
    puts "running nam..."
    exec nam ex2.nam &
    exec xgragh win2.tr &
    exit 0
}

$ns run

#awk -f ex2.awk ex2.tr

BEGIN{
    last = 0;
    tcp_sz = 0;
    cbr_sz = 0;
    total_sz = 0;
}
{
    action=$1;
    time=$2;
    form=$3;
    to=$4;
    type=$5;
    pktsize=$6;

    if($1 == "r" && $5 == "tcp" && to == "4")
        tcp_sz+=pktsize;
    if($1 == "r" && $5 == "cbr" && to == "4")
        cbr_sz+=pktsize;
    total_sz+=pktsize;
}
END{
    printf("Time = %f", time);
    last=total_sz;
    printf("Throughput %f", (total_sz*8/1000000));  #megabytes  #1 byte=8bits
    print time(tcp_sz*8/1000000);
}


/*
END {
    printf("Time = %f\n", time);          # Print the value of time
    last = total_sz;                      # Assign total_sz to last for reference

    printf("Throughput = %f Mbps\n", (total_sz * 8 / 1000000));   #throughput in Mbps
    printf("TCP Throughput = %f Mbps\n", (tcp_sz * 8 / 1000000));  #throughput of TCP packets in Mbps
}
*/

#grep -c "r" ex2.tr
```

```


#3...........................................................................

#Distance vector routing protocol.
#link b/w node 1 and 4 breaks at 1.0 ms, comes up at 3.0 ms.
#source node 0 transmits packets to node 4.
#congestion window when TCP sends packets via other nodes.
#own parameters for bandwidth and delay.

set ns [new Simulator]
set tf [open ex3.tr w]
set nf [open ex3.nam w]
set cwind [open win3.tr w]
$ns trace-all $tf
$ns namtrace-all $nf
$ns rtproto DV
#set the protocol as Dist Vector #dynamic routing protocol #updates source every 2 ms

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

$ns color 1 Orange

$ns duplex-link $n0 $n1 1Mb 10ms DropTail
$ns duplex-link $n1 $n4 1Mb 10ms DropTail
$ns duplex-link $n4 $n5 1Mb 10ms DropTail
$ns duplex-link $n0 $n2 1Mb 10ms DropTail
$ns duplex-link $n2 $n3 1Mb 10ms DropTail
$ns duolex-link $n3 $n5 1Mb 10ms DropTail

$ns duplex-link-op $n0 $n1 orient up-right
$ns duplex-link-op $n1 $n4 orient right
$ns duplex-link-op $n4 $n5 orient down-right
$ns duplex-link-op $n0 $n2 orient down-right
$ns duplex-link-op $n2 $n3 orient right
$ns duplex-link-op $n3 $n5 orient up-right
#not neccesary
$ns queue-limit $n2 $n3 10
$ns queue-limit $n1 $n4 10

set tcp0 [new Agent/TCP]
set sink0 [new Agent/TCPSink]
set ftp0 [new Application/FTP]
$ns attach-agent $n0 $tcp0
$ns attach-agent $n4 $sink0
$ftp0 attach-agent $tcp0
$tcp0 set fid_ 1

$ns rtmodel-at 1.0 down $n1 $n4
$ns rtmodel-at 3.0 up $n $n4

$ns at 0.1 "$ftp start"
$ns at 10.0 "finish"

proc plotWindow {tcpSource file} {
    global ns
    set time 0.01
    set now [$ns now]
    set cwnd [$tcpSource set cwnd_]
    puts $file "$now $cwnd"
    $ns at [expr $now+$time] "plotWindow $tcpSource $file"
}
$ns at 1.0 "plotWindow $tcp0 $cwind"

proc finish {} {
    global ns tf nf cwind
    $ns flush-trace
    close $tf
    close $nf
    close $cwind
    puts "running nam..."
    exec nam ex3.nam &
    exec xgraph win3.tr &
    exit 0
}

$ns run


#4.............................................................................

#server is running a FTP application over TCP.
#client sends a request to download a file of size 10Mb form the server.
#node n0 - server , node n1 - client.
#TCP packet size is 1500 Bytes

set ns [new Simulator]
set tf [open ex4.tr w]
set nf [open ex4.nam w]
$ns trace-all $tf
$ns namtrace-all $nf

set s [$ns node]
set c [ns node]
$s label "Server"
$c label "Client"
$ns color 1 Blue

$ns duplex-link $s $c 10Mb 22ms DropTail
$ns duplex-link-op $s $c orient right

set tcp0 [new Agent/TCP]
set sink0 [new Agent/TCPSink]
set ftp0 [new Application/FTP]
$ns attach-agent $s $tcp0
$ns attach-agent $c $sink0
$ns connect $tcp0 $sink0
$ftp0 attach-agent $tcp0

$tcp0 set packetsize_ 1500
$tcp0 set fid_ 1

proc finish {} {
    global ns tf nf
    $ns flush-trace
    close tf
    close nf
    exec nam ex4.nam &
    exec awk -f transfer.awk ex4.tr &
    exec awk -f convert.awk ex4.tr &
    exec xgragh convert.tr -geometry 800*400 -t "bytes_received_at_client" -x "time_in_secs" -y "bytes-in-bps" &
}

$ns at 0.01 "$ftp0 start"
$ns at 15.0 "$ftp0 stop"
$ns at 15.1 "finish"

$ns run

#transfer.awk

BEGIN {
    count = 0;
    time = 0;
    total_bytes_sent = 0;
    total_bytes_received = 0;
}
{
    if($1=="r" && $5=="tcp" && $4==1)
        total_bytes_received += $6;
    if($1=="+" && $5=="tcp" && $3==0)
        total_bytes_sent += $6;
}

END{
    system("clear");
    printf("Tranmission time required to transfer the file = %f", $2);
    printf("Actual data sent form the server is %f Mbps", (total_bytes_sent)/1000000);
    printf("Data recieved by the client is %f Mbps", (total_bytes_received)/1000000);
}

#convert.awk

BEGIN {
    count = 0;
    time = 0;
}
{
    if ($1=="r" && $5=="tcp" && $4==1) {
        count += $6;
        time = $2;
        printf("%f %f", time,(count)/1000000);
    }
}
END{
}


#5.............................................................................

#multicast routing protocol
#own parameters for bandwidth delay

set ns [new Simulator -multicast on]
set tf [open mcast.tr w]
set nf [open mcast.nam w]
$ns trace-all $tf
$ns namtrace-all $nf

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]

$n0 color Blue
$n1 color Blue
$n5 color Purple
$n6 color Purple
$n7 color Purple

$n0 label "Source 1"
$n1 label "Source 2"
$n5 label "Reciever 1"
$n6 label "Reciever 2"
$n7 label "Reciever 3"

$ns duplex-link $n0 $n2 1.5Mb 10ms DropTail
$ns duplex-link $n1 $n2 1.5Mb 10ms DropTail
$ns duplex-link $n2 $n3 1.5Mb 10ms DropTail
$ns duplex-link $n3 $n7 1.5Mb 10ms DropTail
$ns duplex-link $n3 $n4 1.5mb 10ms DropTial
$ns duplex-link $n4 $n5 1.5Mb 10ms DropTail
$ns duplex-link $n4 $n6 1.5Mb 10ms DropTail

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

set group1 [Node allocaddr]
set group2 [Node allocaddr]
set udp0 [new Agent/UDP]
set cbr1 [new Appliaction/Traffic/CBR]
$ns attach-agent $n0 $udp0
$udp0 set dst_addr_ $group1
$udp0 dst_port_ 0
$cbr1 attach-agent $udp0

set udp1 [new Agent/UDP]
set cbr2 [new Application/Traffic/CBR]
$ns attach-agent $n1 $udp1
$udp1 set dst_addr_ $group2
$udp1 set dst_port_ 0
$cbr2 attach-agent $udp1


set rcvr1 [new Agent/Null]
$ns attach-agent $n5 $rcvr1
$ns at 1.0 "$n5 join-group $rcvr1 $group1"

set rcvr2 [new Agent/Null]
$ns attach-agent $n6 $rcvr2
$ns at 1.5 "$n6 join-group $rcvr2 $group1"

set rcvr3 [new Agent/Null]
$ns attach-agent $n7 $rcvr3
$ns at 2.0 "$n7 join-group $rcvr3 $group1"
#
set rcvr4 [new Agent/Null]
$ns attach-agent $n5 $rcvr4
$ns at 2.5 "$n5 join-group $rcvr4 $group2"

set rcvr5 [new Agent/Null]
$ns attach-agent $n6 $rcvr5
$ns at 3.0 "$n6 join-group $rcvr5 $group2"

set rcvr6 [new Agent/Null]
$ns atatch-agent $n7 $rcvr6
$ns at 3.5 "$n7 join-group $rcvr6 $group2"

$ns at 4.0 "$n5 leave-group $rcvr1 $group1"
$ns at 4.5 "$n6 leave-group $rcvr2 $group1"
$ns at 5.0 "$n7 leave-group $rcvr3 $group1"
$ns at 5.5 "$n5 leave-group $rcvr4 $group2"
$ns at 6.0 "$n6 leave-group $rcvr5 $group2"
$ns at 6.5 "$n7 leave-group $rcvr6 $group2"

$ns at 0.5 "$cbr1 start"
$ns at 9.5 "$cbr1 stop"
$ns at 0.5 "$cbr2 start"
$ns at 9.5 "$cbr2 stop"
$ns at 10.0 "finish"

proc finish {} {
    global ns tf nf
    $ns flush-trace
    close $tf
    close $nf
    exec nam mcat.nam &
    exit 0
}

$ns run
```