

Name - Abinash Gupta
Roll No- 120CS0157
Computer Networks LAB

Write a TCL script to create a topology as shown in Figure-1 where two TCP connections are created from S0 to D0 and S1 to D1 using FTP traffic and node S2 to D2 and S3 to D3 UDP connections are created, and data are sent using CBR traffic. Run the simulation for 60 seconds and find the followings:

- Compare the congestion window size for two connections where TCP connection1 uses TCP Tahoe and TCP connection2 uses TCP Reno algorithm.
- Also compare the TCP throughput received at D0 and D1 for these two connections.

TCL Script

```
#create a new Simulator
set ns [new Simulator]

$ns color 1 Blue
$ns color 2 Red
$ns color 3 Green
$ns color 4 Black

#create trace file
set tf [open q.tr w]
set nf [open q.nam w]

$ns trace-all $tf
$ns namtrace-all $nf

#create nodes
for {set i 0} {$i < 10} {incr i} {
    set n[expr $i] [$ns node]
}

#create links
$ns duplex-link $n0 $n4 100Mb 25ms DropTail
$ns duplex-link $n1 $n4 100Mb 25ms DropTail
$ns duplex-link $n2 $n4 100Mb 25ms DropTail
$ns duplex-link $n3 $n4 100Mb 25ms DropTail

$ns duplex-link $n5 $n6 100Mb 25ms DropTail
$ns duplex-link $n5 $n7 100Mb 25ms DropTail
```

```
$ns duplex-link $n5 $n8 100Mb 25ms DropTail
$ns duplex-link $n5 $n9 100Mb 25ms DropTail

$ns duplex-link $n4 $n5 100Mb 100ms DropTail
```

```
#create agents
```

```
set tcp0 [new Agent/TCP]
$tcp0 set fid_ 1
$ns attach-agent $n0 $tcp0
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
set sink0 [new Agent/TCPSink]
$ns attach-agent $n6 $sink0
$ns connect $tcp0 $sink0
```

```
set tcp1 [new Agent/TCP]
$tcp1 set fid_ 2
set ftp1 [new Application/FTP]
$ns attach-agent $n1 $tcp1
$ftp1 attach-agent $tcp1
set sink1 [new Agent/TCPSink]
$ns attach-agent $n7 $sink1
$ns connect $tcp1 $sink1
```

```
set udp0 [new Agent/UDP]
$udp0 set fid_ 3
```

```
set cbr0 [ new Application/Traffic/CBR]
$cbr0 set packetSize_ 1000
$cbr0 attach-agent $udp0
$ns attach-agent $n3 $udp0
set null0 [new Agent/Null]
$ns attach-agent $n8 $null0
$ns connect $udp0 $null0
```

```
set udp1 [new Agent/UDP]
$udp1 set fid_ 4
$ns attach-agent $n3 $udp1
set cbr1 [ new Application/Traffic/CBR]
$cbr1 set packetSize_ 1000
```

```
$cbr1 attach-agent $udp1
set null1 [new Agent/Null]
$ns attach-agent $n9 $null1
$ns connect $udp1 $null1

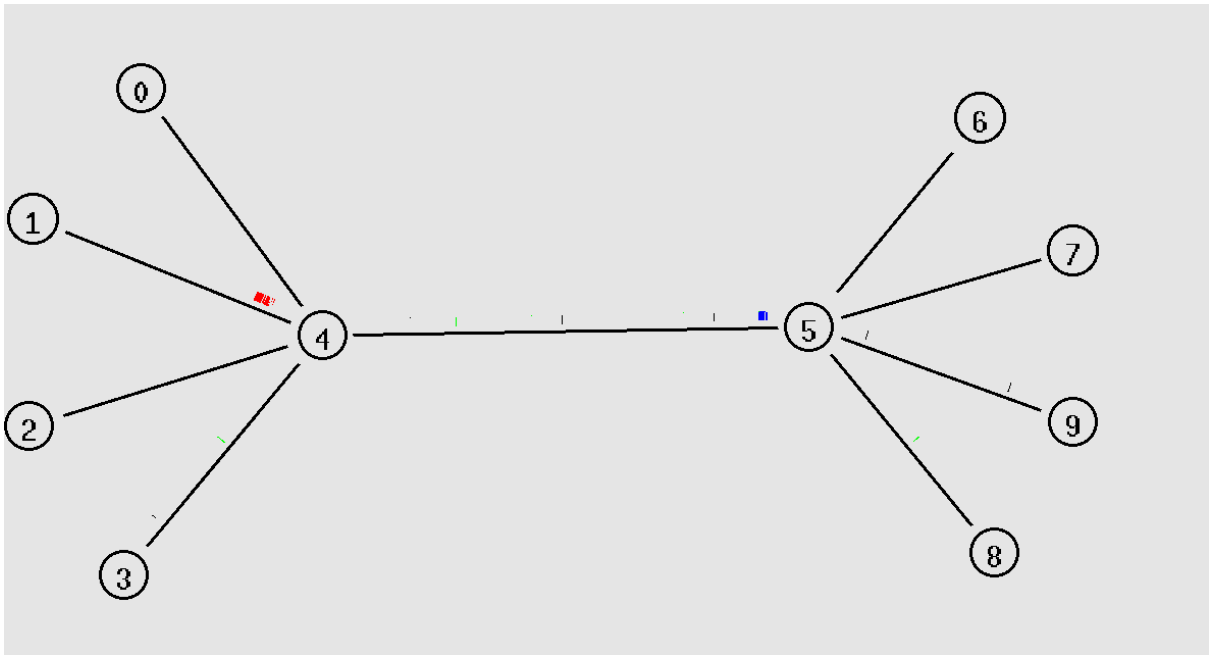
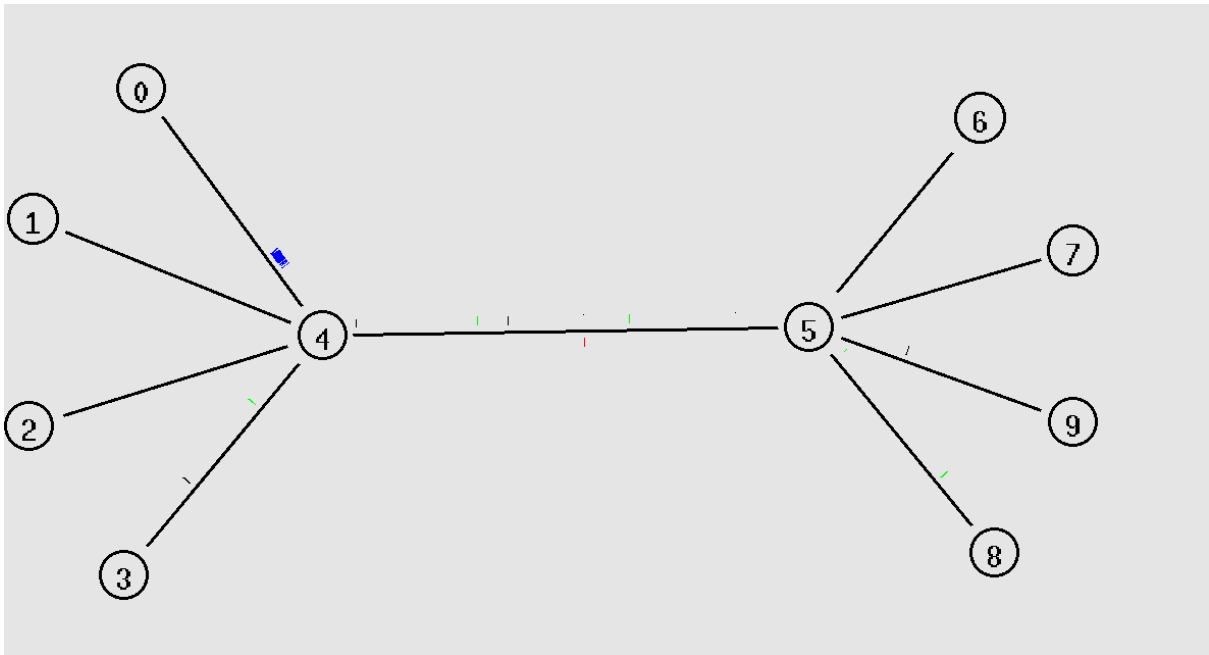
#start the simulation
$ns at 0.0 "$ftp0 start"
$ns at 0.1 "$ftp1 start"
$ns at 0.2 "$cbr0 start"
$ns at 0.3 "$cbr1 start"

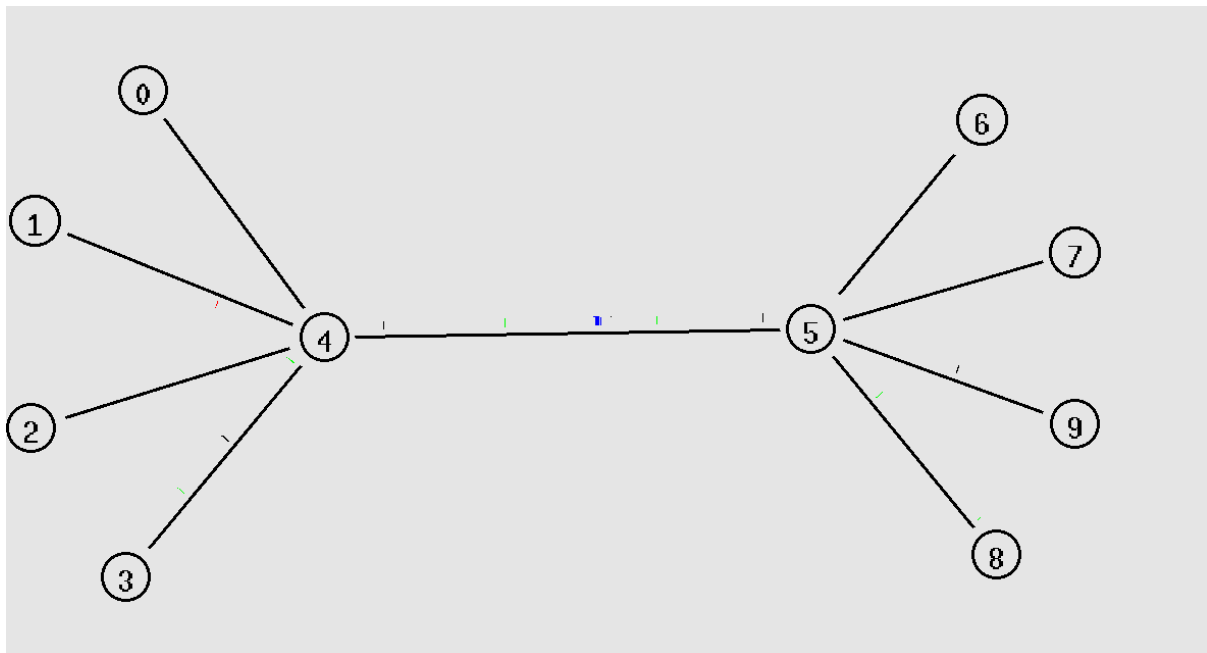
$ns at 54.0 "$ftp0 stop"
$ns at 55.0 "$ftp1 stop"
$ns at 56.0 "$cbr0 stop"
$ns at 57.0 "$cbr1 stop"

$ns at 60 "finish"

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

$ns run
```





THROUGHPUT AWK FILE FOR D0

```
#!/usr/bin/awk -f

BEGIN {
    # Set default values for variables
    interval = 0.1 # Time interval in seconds
    time = 0.1
    tcp_byte_count = 0
    tcp_throughput = 0
}

# Process each packet in the trace file
{
    # Check if the packet is TCP or UDP
    if ($5 == "tcp") {
        tcp_byte_count += $6
    }

    # Update time and calculate throughput every "interval" seconds
    if ($2 >= time)
    {
        tcp_throughput = tcp_byte_count * 8 / interval / 1000
        printf("%.2f %.2f\n", time, tcp_throughput)
        time += interval
        tcp_byte_count = 0
    }
}
```

```

    }
}

# Print final throughput if needed
END {
    ; #if (tcp_byte_count > 0 || udp_byte_count > 0) {
        #   tcp_throughput = tcp_byte_count * 8 / interval / 1000000
        #   udp_throughput = udp_byte_count * 8 / interval / 1000000
        #   printf("%.2f %.2f %.2f\n", time, tcp_byte_count,
udp_byte_count)
        #}
    }
}

```

FOR D1

```

#!/usr/bin/awk -f

BEGIN {
    # Set default values for variables
    interval = 0.1 # Time interval in seconds
    time = 0.1
    tcp_byte_count = 0
    tcp_throughput = 0
}

# Process each packet in the trace file
{
    # Check if the packet is TCP or UDP
    if ($5 == "udp") {
        tcp_byte_count += $6
    }

    # Update time and calculate throughput every "interval" seconds
    if ($2 >= time)
    {
        tcp_throughput = tcp_byte_count * 8 / interval / 1000
        printf("%.2f %.2f\n", time, tcp_throughput)
        time += interval
        tcp_byte_count = 0
    }
}

```

```

}

# Print final throughput if needed
END {
    ; #if (tcp_byte_count > 0 || udp_byte_count > 0) {
        #   tcp_throughput = tcp_byte_count * 8 / interval / 1000000
        #   udp_throughput = udp_byte_count * 8 / interval / 1000000
        #   printf("%.2f %.2f %.2f\n", time, tcp_byte_count,
udp_byte_count)
        #}
    }
}

```

OUTPUT

