

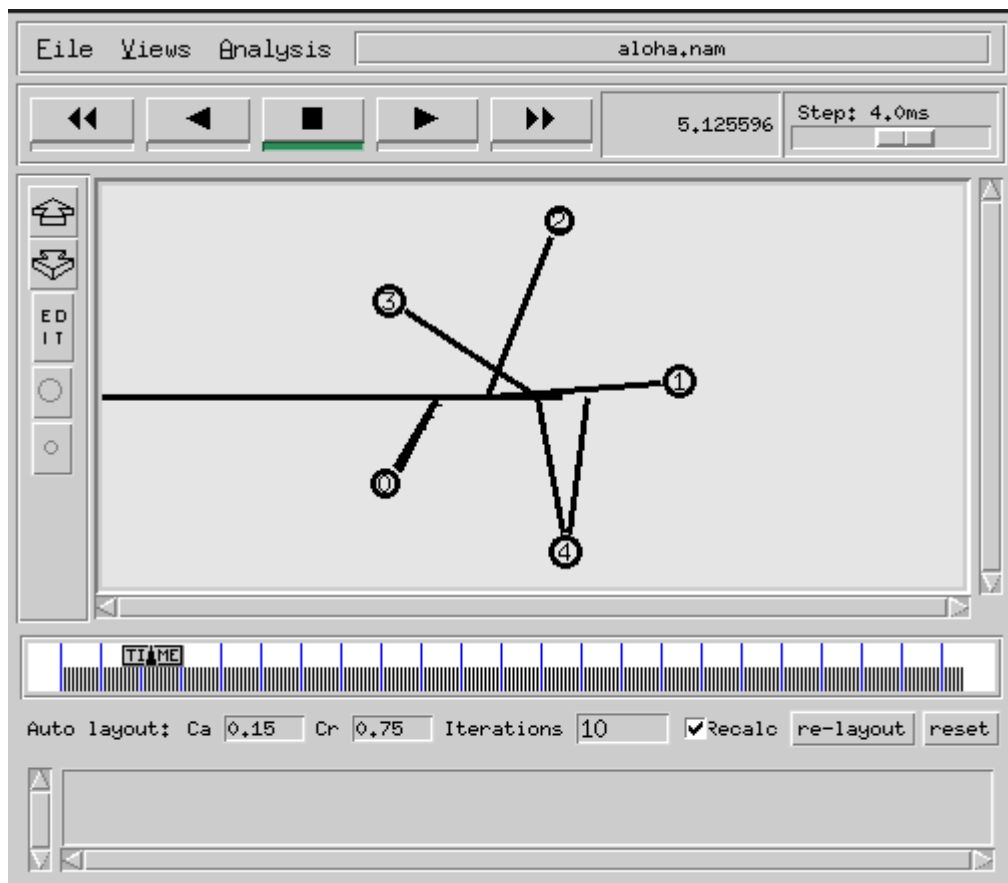
NETWORKS LABORATORY - 8

To analyse the performance of Pure Aloha, Slotted Aloha and CSMA MAC Protocols against increasing frame sizes (200,300,400) in a given network.

Pure Aloha:

```
set ns [new Simulator]
set nf [open aloha.nam w]
$ns namtrace-all $nf
set nftr [open aloha.tr w]
$ns trace-all $nftr
LanRouter set debug_ 0
proc finish {} {
    global ns nf nftr
    $ns flush-trace
    close $nf
}
set time 50
set packet_size [lindex $argv 0]
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
lappend nodelist $n0
lappend nodelist $n1
lappend nodelist $n2
lappend nodelist $n3
lappend nodelist $n4
set lan0 [$ns newLan $nodelist .2Mb 1ms LL Queue/DropTail
Mac/Sat/UnslottedAloha ]
# Node A
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
$tcp0 set packetSize_ $packet_size
# Node B
set tcp1 [new Agent/TCP]
$ns attach-agent $n1 $tcp1
$tcp1 set packetSize_ $packet_size
# Node D
set sink0 [new Agent/TCPSink]
$ns attach-agent $n3 $sink0
$ns connect $tcp0 $sink0
# Node E
```

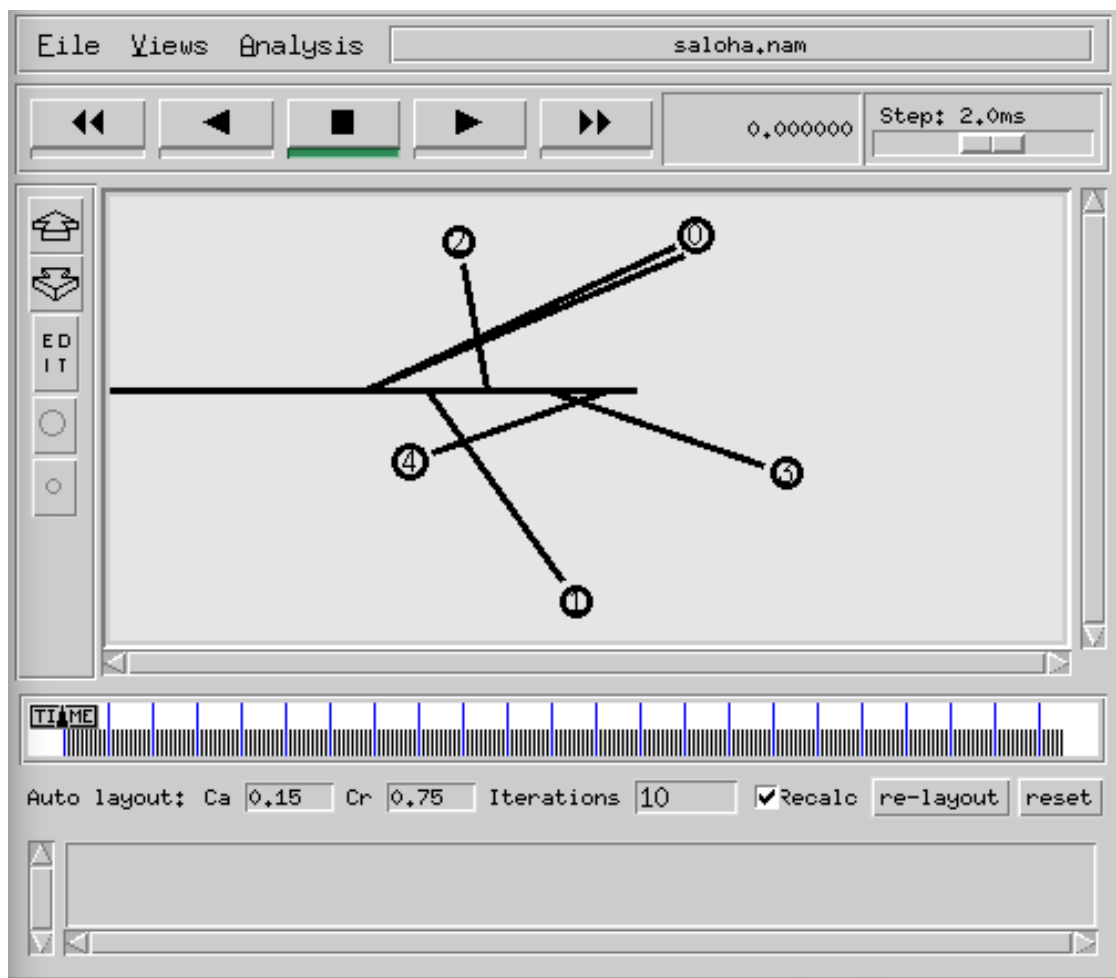
```
set sink1 [new Agent/TCPSink]
$ns attach-agent $n4 $sink1
$ns connect $tcp1 $sink1
set ftp0 [new Application/FTP]
$ftp0 set interval_ .02
$ftp0 attach-agent $tcp0
$ns at 5 "$ftp0 start"
$ns at $time "$ftp0 stop"
set ftp1 [new Application/FTP]
$ftp1 set packetSize_ $packet_size
$ftp1 set interval_ .02
$ftp1 attach-agent $tcp1
$ns at 15 "$ftp1 start"
$ns at $time "$ftp1 stop"
$ns at $time "finish"
$ns run
```



Slotted aloha:

```
set ns [new Simulator]
set nf [open saloha.nam w]
$ns namtrace-all $nf
set nftr [open saloha.tr w]
$ns trace-all $nftr
LanRouter set debug_ 0
proc finish {} {
    global ns nf nftr
    $ns flush-trace
    close $nf
}
set time 50
set packet_size [lindex $argv 0]
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
lappend nodelist $n0
lappend nodelist $n1
lappend nodelist $n2
lappend nodelist $n3
lappend nodelist $n4
set lan0 [$ns newLan $nodelist .2Mb 1ms LL Queue/DropTail
Mac/Sat/SlottedAloha]
# Node A
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
$tcp0 set packetSize_ $packet_size
# Node B
set tcp1 [new Agent/TCP]
$ns attach-agent $n1 $tcp1
$tcp1 set packetSize_ $packet_size
# Node D
set sink0 [new Agent/TCPSink]
$ns attach-agent $n3 $sink0
$ns connect $tcp0 $sink0
# Node E
set sink1 [new Agent/TCPSink]
$ns attach-agent $n4 $sink1
$ns connect $tcp1 $sink1
set ftp0 [new Application/FTP]
$ftp0 set packetSize_ $packet_size
$ftp0 set interval_ .02
$ftp0 attach-agent $tcp0
$ns at 5 "$ftp0 start"
$ns at $time "$ftp0 stop"
```

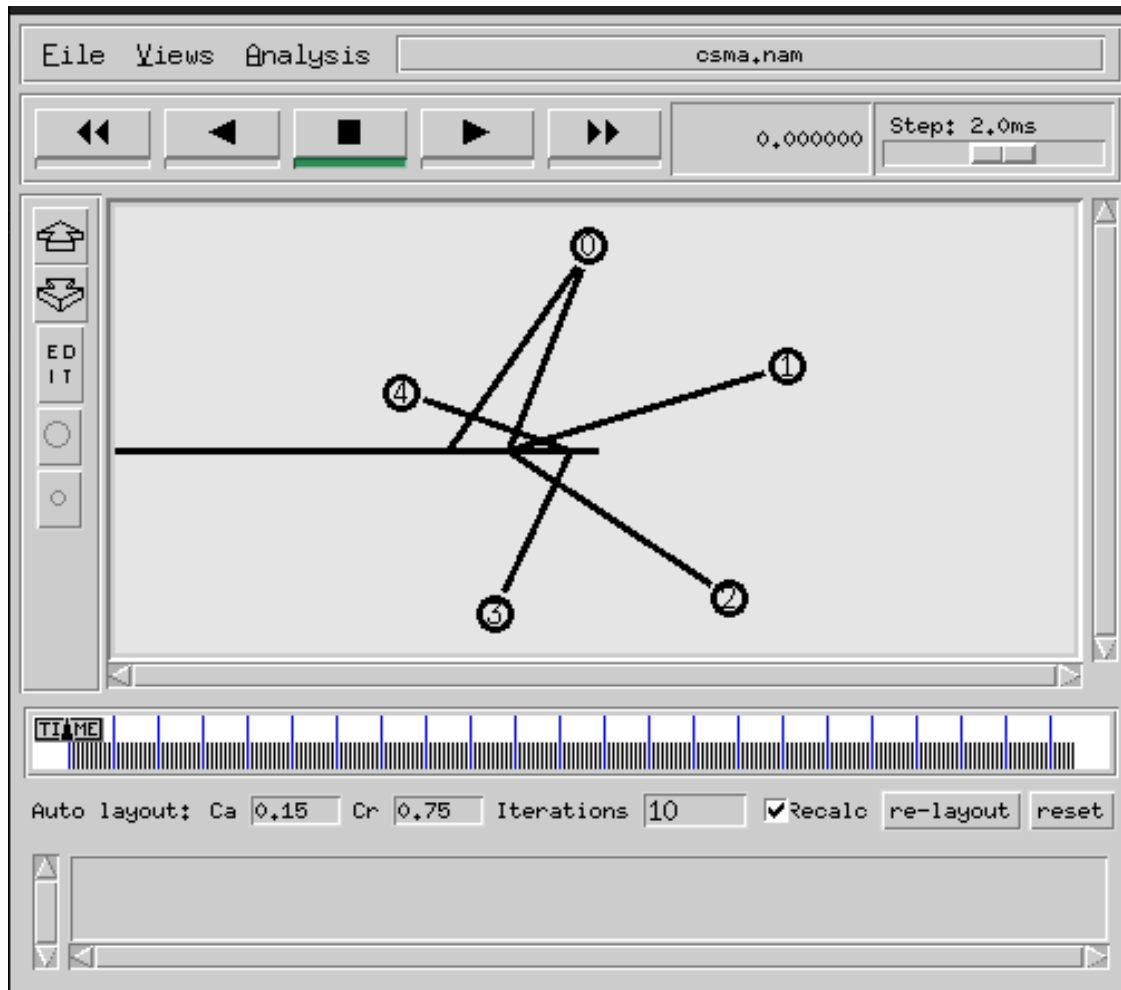
```
set ftp1 [new Application/FTP]
$ftp1 set packetSize_ $packet_size
$ftp1 set interval_ .02
$ftp1 attach-agent $tcp1
$ns at 15 "$ftp1 start"
$ns at $time "$ftp1 stop"
$ns at $time "finish"
$ns run
```



CSMA:

```
set ns [new Simulator]
set nf [open csma.nam w]
$ns namtrace-all $nf
set nftr [open csma.tr w]
$ns trace-all $nftr
LanRouter set debug_ 0
proc finish {} {
    global ns nf nftr
    $ns flush-trace
    close $nf
}
set time 50
set packet_size [lindex $argv 0]
#Create six nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
lappend nodelist $n0
lappend nodelist $n1
lappend nodelist $n2
lappend nodelist $n3
lappend nodelist $n4
set lan0 [$ns newLan $nodelist 0.2Mb 1ms LL Queue/DropTail Mac/Csma/Cd ]
# Node A
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
$tcp0 set packetSize_ $packet_size
#Node B
set tcp1 [new Agent/TCP]
$ns attach-agent $n1 $tcp1
$tcp1 set packetSize_ $packet_size
# Node D
set sink0 [new Agent/TCPSink]
$ns attach-agent $n3 $sink0
$ns connect $tcp0 $sink0
# Node E
set sink1 [new Agent/TCPSink]
$ns attach-agent $n4 $sink1
$ns connect $tcp1 $sink1
set ftp0 [new Application/FTP]
$ftp0 set packetSize_ $packet_size
$ftp0 set interval_ 0.02
$ftp0 attach-agent $tcp0
$ns at 5 "$ftp0 start"
$ns at $time "$ftp0 stop"
```

```
set ftp1 [new Application/FTP]
$ftp1 set packetSize_ $packet_size
$ftp1 set interval_ 0.02
$ftp1 attach-agent $tcp1
$ns at 15 "$ftp1 start"
$ns at $time "$ftp1 stop"
$ns at $time "finish"
$ns run
```



#plot.py

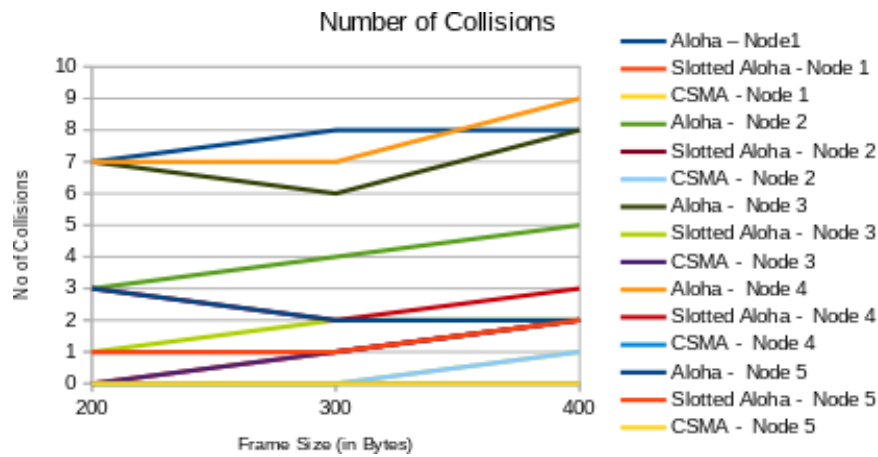
```
def calculate_collision(trace_filename):
    List = {}
    with open(trace_filename, 'r') as trace_file:
        for idx, line in enumerate(trace_file):
            line = line.strip().split()
            packet_id = line[-1]
            time1 = float(line[1])
            for idx, line2 in enumerate(trace_file):
                line2 = line2.strip().split()
                packet_id2 = line2[-1]
                time2 = float(line2[1])
                if (len(line2) <= 4 or len(line) <= 4):
                    continue
                if packet_id2 != packet_id:
                    if abs(time1 - time2) <= 0.02:
                        List[int(line2[2])] += 1
                        List[int(line2[3])] += 1
            return List

def calculate_packet_loss(trace_filename):
    packets_sent = 0 # Total packets sent
    packets_received = 0 # Total packets received
    packet_loss = 0
    with open(trace_filename, 'r') as trace_file:
        for line in trace_file:
            dataList = line.split()
            if dataList[0] == '-':
                packets_sent += 1
            if dataList[0] == 'r':
                packets_received += 1
            packet_loss += packets_sent - packets_received
    return packet_loss

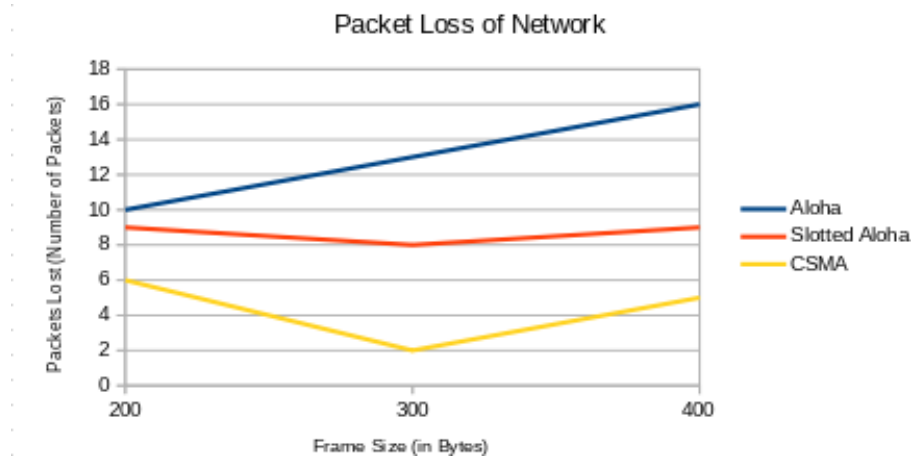
def calculate_throughput(trace_filename):
    packets_received = 0 # Total packets received
    packet_size = 1024
    time = 100
    with open(trace_filename, 'r') as trace_file:
        for line in trace_file:
            dataList = line.split()
            if dataList[0] == 'r':
                packets_received += packet_size
            throughput = packets_received / time
    return throughput
```

Output:

1) Number of Collisions:



2) Packet Loss:



3) Throughput:

