NETWORKINHG

NS3 REPORT

DEFENDING BLACK HOLE ATTAK

2021/2022

Supervised By

Dr. Md. Shohrab Hossain

SUBMITTED BY

1705087

FAHMID- AL- RIFAT

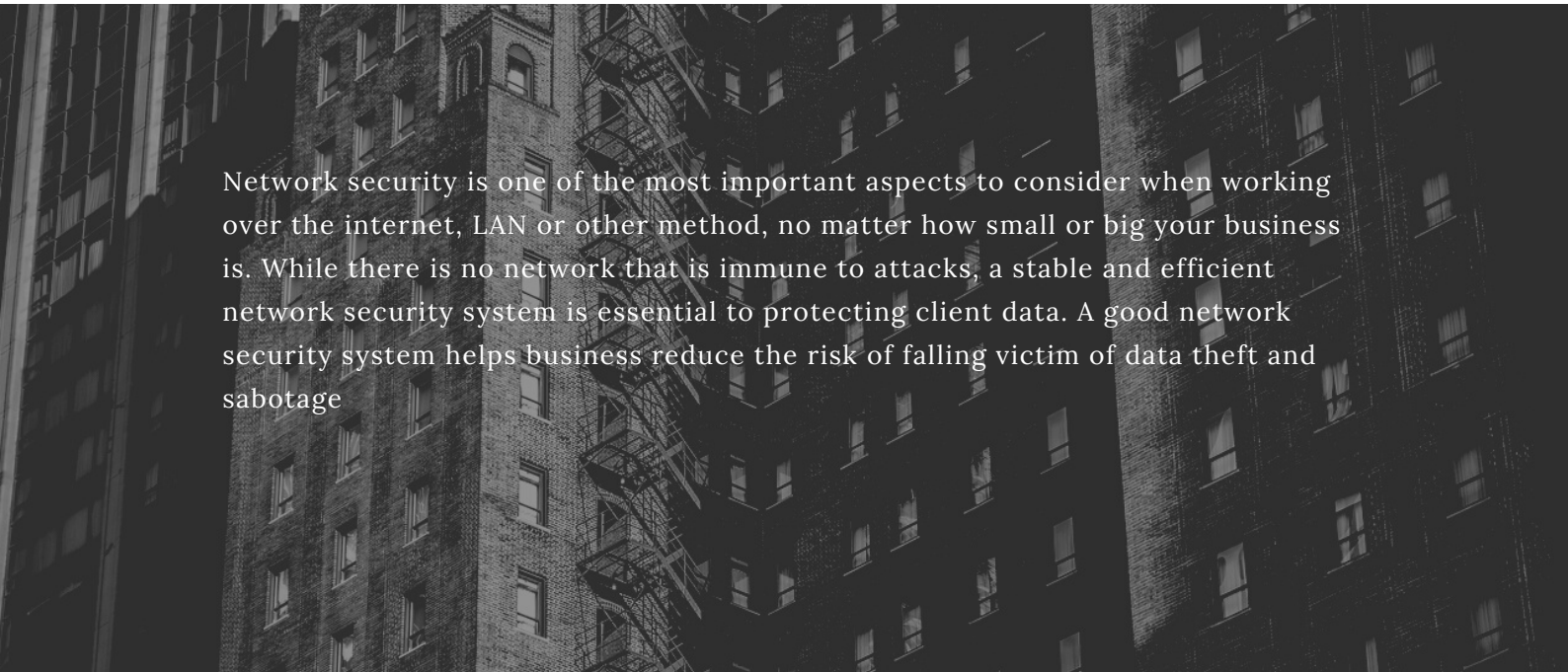
SHORT SUMMARY

Simulate and Defending black hole attack And Topology analysis

Task A:
Here we build two topologies with different type of components and measure performance varying different type of metrices and compare them with help of graph plotting

Task B:
An ad-hoc network is a collection of wireless mobile hosts forming a temporary network without the assistance of any stand-alone infrastructure or centralized administration .These types of networks are generally open to attack to information and physical security threats than fixed wired networks. The black hole attack is an active insider attack where the attacker consumes the intercepted packets without any forwarding .Our target is minimize the damage of black whole attack with our proposed Algorithm

Network security is one of the most important aspects to consider when working over the internet, LAN or other method, no matter how small or big your business is. While there is no network that is immune to attacks, a stable and efficient network security system is essential to protecting client data.



OUR REPORT FORMAT

Our report written mentioning the followings

Our report written mentioning the followings

a. Network **topologies** under simulation

b. **Parameters** under variation

c. Overview of the proposed **algorithm**

d. **Modifications** made in the simulator

e. Results with **graphs** (for Task A and B)

f. **Summary** findings explaining the results you found in Task A and Task B

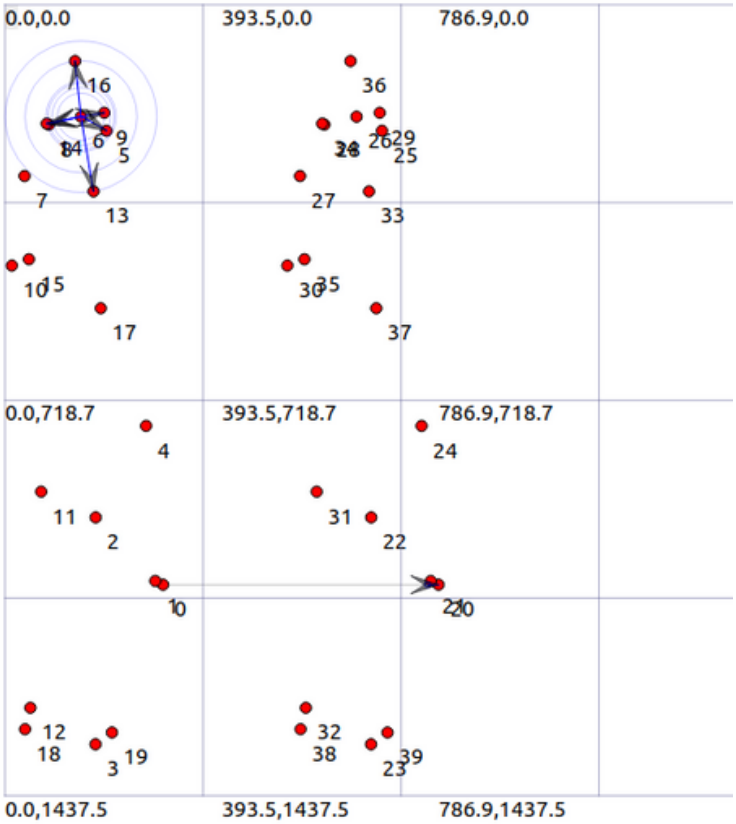
ALL OUR TOPOLOGIES

Task A - Part 1 and Task B

Topology 1 :

Here we use 2 manet network which are connected with a point to point network and AODV is the working routing protocol here and Thus three network created .

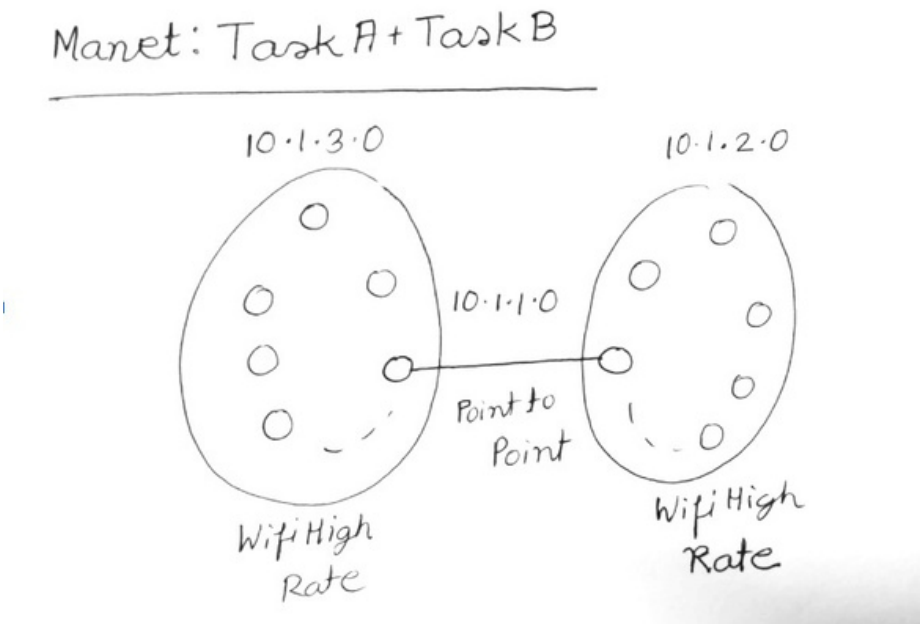
The performance metrices are good enough for moving manet nodes when communicate among themselves but it decrease slightly when they try to communicate between to manet with help of point to point helper .



ALL OUR TOPOLOGIES

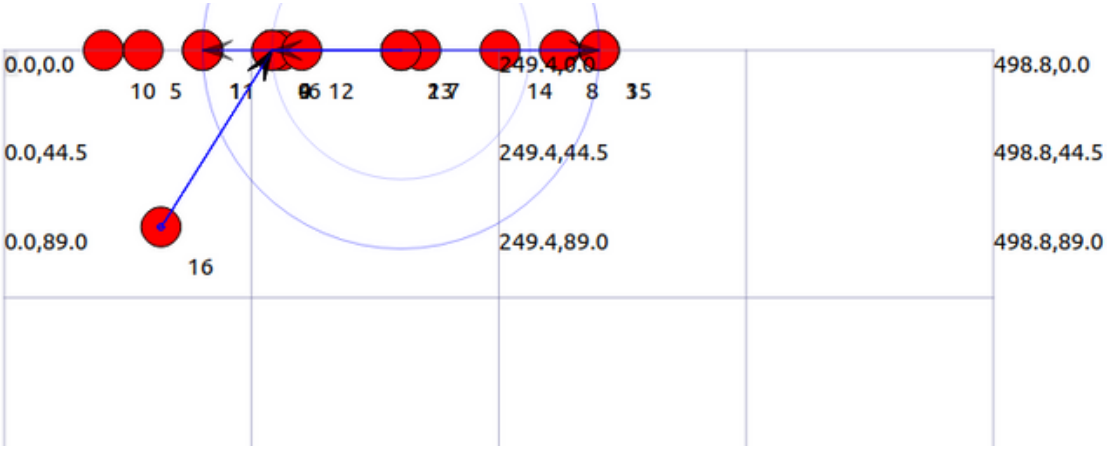
Task A - Part 1 and Task B

```
NS_LOG_INFO ("assigning ip address");


Ipv4AddressHelper addressAdhoc;

addressAdhoc.SetBase ("10.1.3.0", "255.255.255.0");
Ipv4InterfaceContainer p2pInterfaces;
p2pInterfaces = addressAdhoc.Assign (p2pDevices);


addressAdhoc.SetBase ("10.1.1.0", "255.255.255.0");
Ipv4InterfaceContainer adhocApInterfaces,adhocInterfaces;
adhocInterfaces = addressAdhoc.Assign (adhocDevices);
//adhocApInterfaces = addressAdhoc.Assign (wifiApDevices);

addressAdhoc.SetBase ("10.1.2.0", "255.255.255.0");
Ipv4InterfaceContainer adhocApInterfaces2,adhocInterfaces2;
adhocInterfaces2 = addressAdhoc.Assign (adhocDevices2);
//adhocApInterfaces2 = addressAdhoc.Assign (wifiApDevices2);
```



ALL OUR TOPOLOGIES

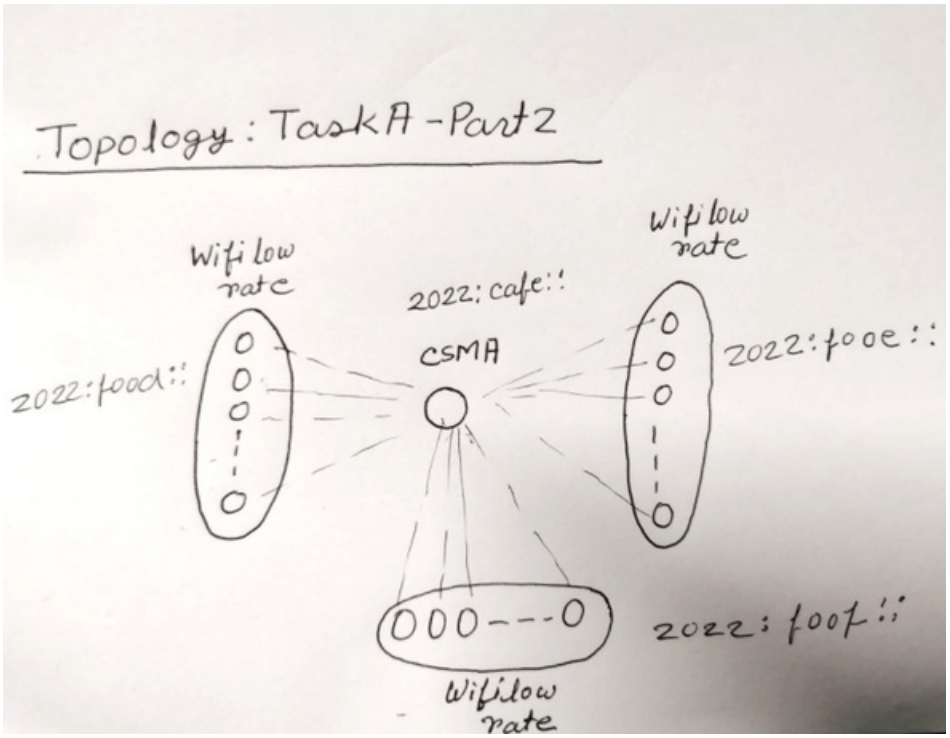
Task A - Part 2



Topology 2 :

Here we use 3 wireless mobile LR-Pan network which are connected with a CSMA network and Mesh is the working routing protocol here and Thus Four network created .

The performance metrices are good not up to the mark for this topology as Low rate W-Pan is used



ALL OUR TOPOLOGIES

Task A - Part 2

```
ipv6.SetBase (Ipv6Address ("2022:f00d::"), Ipv6Prefix (64));
Ipv6InterfaceContainer lr_interfaces;
lr_interfaces = ipv6.Assign (SixLow_devices);
lr_interfaces.SetForwarding (0, true);
lr_interfaces.SetDefaultRouteInAllNodes (0);

ipv6.SetBase (Ipv6Address ("2022:cafe::"), Ipv6Prefix (64));
Ipv6InterfaceContainer c_interfaces;
c_interfaces = ipv6.Assign (csmDevices);
c_interfaces.SetForwarding (1, true);
c_interfaces.SetDefaultRouteInAllNodes (1);
```

To get low rate Wpan we used LR Wpan Helper which work with Iv6 adressing

```
NetDeviceContainer lrDevices = lrWifi.Install (Lw_nodes);
lrWifi.AssociateToPan (lrDevices, 0);

LrWpanHelper lrWifi2;
NetDeviceContainer lrDevices2 = lrWifi2.Install (Lw_nodes2);
lrWifi2.AssociateToPan (lrDevices2, 0);

LrWpanHelper lrWifi3;
NetDeviceContainer lrDevices3 = lrWifi2.Install (Lw_nodes3);
lrWifi3.AssociateToPan (lrDevices3, 0);

InternetStackHelper ISv6;
ISv6.Install (Lw_nodes);
ISv6.Install (Lw_nodes2);
ISv6.Install (Lw_nodes3);
ISv6.Install (Csma_Nodes.Get (0));
```

PARAMETERS UNDER VARIATIONS

In the specifications :

a. The number of nodes needs to be varied as 20, 40, 60, 80, and 100

b. Besides, you need to vary the following parameters –

i. The number of flows (10, 20, 30, 40, and 50)

ii. The number of packets per second (100, 200, 300, 400, and 500)

iii. Speed of nodes (5 m/s, 10 m/s, 15 m/s, 20 m/s, and 25 m/s) [Only in case of having mobility]

iv. Coverage area (square coverage are varying one side as Tx_range, 2 x Tx_range, 3 x Tx_range, 4 x Tx_range, and 5 x Tx_range) [Only in case of having static nodes only]

In all cases, you need to measure the following metrics and plot graphs –

a. Network throughput

b. End-to-end delay

---

c. Packet delivery ratio (total # of packets delivered to end destination/total # of packets sent)

d. Packet drop ratio (total # of packets dropped / total # of packets sent)

Note that, show the results of (c) and (d) in the layer you are working on your project.

Task wise variation are given bellow :

1) Wireless high-rate (802.11) - (mobile)
- nodes , flows , mobility , packets per sec

2) Wireless low-rate (e.g., 802.15.4) (mobile)
- nodes , flows , mobility , packets per sec

3) In our proposed Algo - Nodes

Bonus : Jitter , Flows , Per sec and Per flow Thorghput

OVERVIEW OF THE PROPOSED ALGORITHM

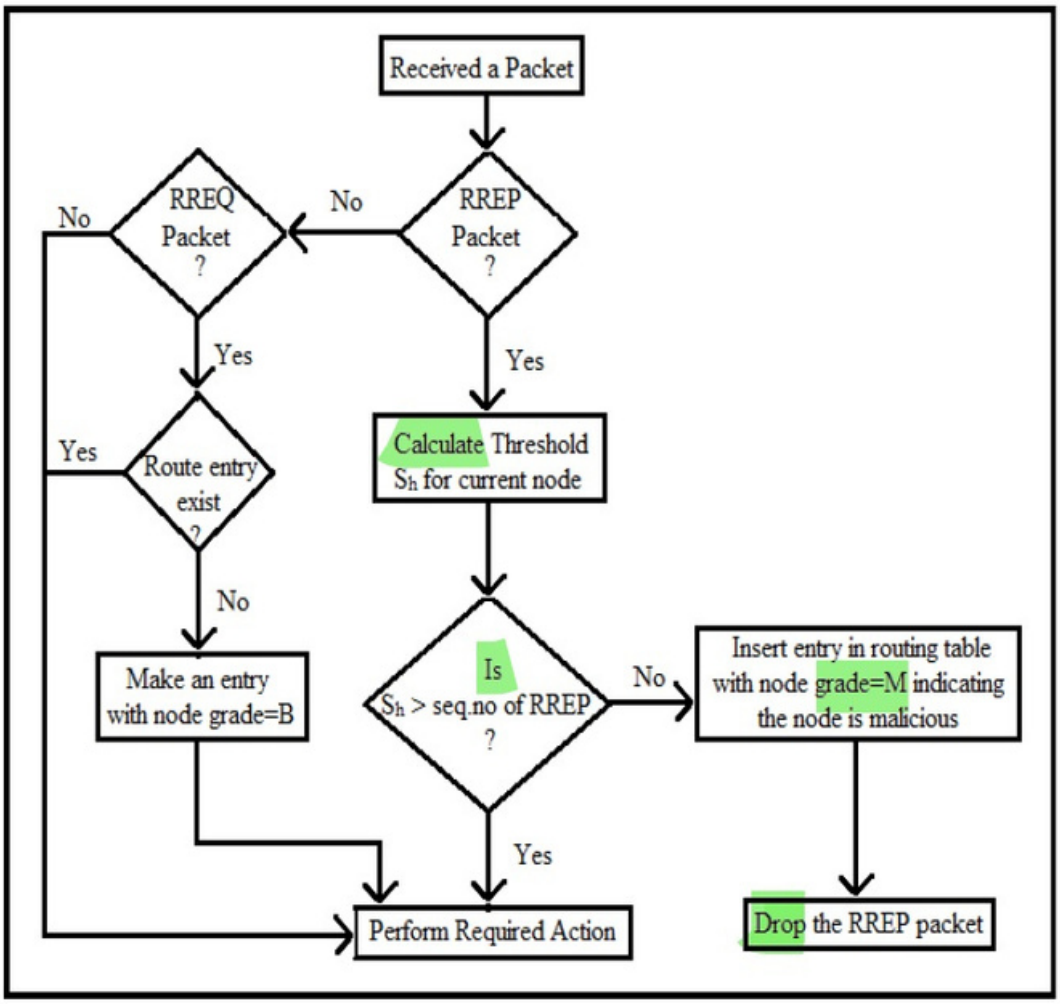


Fig. 1. Process Flow of Dynamic Sequence Number Thresholding Protocol

OVERVIEW OF THE PROPOSED ALGORITHM

The proposed protocol discussed in this section is Dynamic Sequence Number Thresholding (DSNT) Protocol for securing network against blackhole attack. For detecting the blackhole attack we use its large out of bound sequence number as a signature. We put a threshold on the sequence number from the RREP packet that is reply packet. Here, RREP is a feedback . message and sequence number decides whether it is benevolent or malicious. The threshold for sequence number 'Sh' is calculated as in Equation 1. To implement the DSNT protocol we modify AODV routing protocol. We add a new column named node grade into the routing table which represents whether a node is benevolent (B) or malicious (M). The Figure 1 represents the flow of DSNT protocol . Here, 'N' refers to number of nodes in network and 'maxof(seqno)' refers to the maximum of sequence number from route table entry of current node routing the packet excluding malicious node entries. The 'λ' is Sequence Number Increment Constant which represents amount of sequence numbers generated, that is packets routed, in the network in worst case scenario. The threshold calculation for Sh must be done from the maximum of sequence number in the route table of the current routing node. In addition the size of network must be consider in this calculation as for larger network, the sequence number will be generated at much faster rate as more number of node would want to send data to sink. Therefore at the worst case during the route discovery of a node, we may assume that all the nodes in network generated a RREQ and RREP packet increasing the sequence number [10]. Hence 2 times the number of nodes in network is added in the max(seqno). This brings λ value to 2. The value of λ may be fine tuned for different networks depending upon communication overhead and topology of the network. Hence Equation 1 is now reduced to Equation 2. $Sh = 2 * N + maxof(seqno)$ The flow of the DSNT protocol is represented by Figure 1. When RREQ packet is initiated by any node for route discovery, the route entry along with node grade value as benevolent by default is inserted into every nodes route table about the previous node from which it receives RREQ packet. If route entry already exists it is updated. When black hole node receives this packet it sends a RREP reply packet to its source with manipulated large sequence number. When node running DSNT protocol receives a RREP packet it checks packets sequence number against sequence number threshold Sh. If sequence number of RREP if lesser that threshold Sh then packet is accepted and required further processing is done. Otherwise the node grade of node from which RREP is received is changed to malicious 'M' then RREP packet is dropped.

MODIFICATIONS MADE IN THE SIMULATOR

Our Simulations contains the followings

a. Network **topologies** simulation plot Results with
. **graphs** in Task A for 2 Type of **Networks** Task A

b. Aodv **Manet Network** Simulation for Task B

c. **Black hole** Attack Simulations

d. **Improve Aodv** to Defend the attack

MODIFICATIONS MADE FOR ATTACK SIMULATION

Malicious Black Hole Node Creating :

```
 //Added by 1705087
 .AddAttribute ("IsBlackHoleNode",
                "Set malicious Node",
                BooleanValue (false),
                MakeBooleanAccessor (&RoutingProtocol::SetBlackHoleNode,
                                     &RoutingProtocol::GetBlackHoleNode),
                MakeBooleanChecker ())
;
```

Dropping Packet by malicious node:

```
bool
RoutingProtocol::Forwarding (Ptr<const Packet> p, const Ipv4Header & header,
                             UnicastForwardCallback ucb, ErrorCallback ecb)
{

  //Added by 1705087
  //std :: cout <<"Black Hole Node Attack !!! Not Forwarding , Dropping Packet

  if(m_IsBlackHoleNode)
  {
    return false;
  }
```

MODIFICATIONS MADE FOR ATTACK SIMULATION

Black hole node give false RREP reply (consist of higher sequence number) to RREQ request of Sender and Make false routing Table Entry :

```
{
  //Added By 1705087
  //False routing table entry having RREQ back message and hop count as 1
  //AT RecvRequest Funtion

  if(m_IsBlackHoleNode)
  {

    Ptr<NetDevice> dev = m_ipv4->GetNetDevice (m_ipv4->GetInterfaceForAddress (receiver));
    //Like newEnter
    RoutingTableEntry falseToDst(dev,dst,true,rreqHeader.GetDstSeqno()+100,
    m_ipv4->GetAddress (m_ipv4->GetInterfaceForAddress(receiver),0),1,dst,m_activeRouteTimeout);

    SendReplyByIntermediateNode (falseToDst, toOrigin, rreqHeader.GetGratuitousRrep ());
    return;
  }


  m_routingTable.LookupRoute (origin, toOrigin);
  SendReplyByIntermediateNode (toDst, toOrigin, rreqHeader.GetGratuitousRrep ());
```

To make Higher sequence number we add 100 to victim fake seq reply and and made next hop count 1 to make sure future data from AODV protocol will come to malicious node .So all the future packet will come to Malicious node instead of real victim node and thus attack happens.

Note : We do not show Header File Changes Here

Main Aodv Model File :
-Improve-aodv-routing-protocol.cc
-Improve-aodv-rtable
-Improve-aodv-rqueue

MODIFICATIONS MADE FOR DEFEND ATTACK

```
RoutingTableEntry::RoutingTableEntry (Ptr<NetDevice> dev, Ipv4Address dst, bool vS
                                      Ipv4InterfaceAddress iface, uint16_t hops, I
                                      ,uint32_t malNode)
  : m_ackTimer (Timer::CANCEL_ON_DESTROY),
    m_validSeqNo (vSeqNo),
    m_seqNo (seqNo),
    m_hops (hops),
    m_lifeTime (lifetime + Simulator::Now ()),
    m_iface (iface),
    m_flag (VALID),
    m_reqCount (0),
    m_blackListState (false),
    m_blackListTimeout (Simulator::Now ()),
    node_stat(malNode)
{
  m_ipv4Route = Create<Ipv4Route> ();
  m_ipv4Route->SetDestination (dst);
  m_ipv4Route->SetGateway (nextHop);
```

Adding New Routing Table Entry For Malicious Node detection , It will 1 when node threshold is above sh and No one will make connection if 1 is true and every node keep is entry

```
Ptr<Ipv4Route> m_ipv4Route;
/// Output interface address
Ipv4InterfaceAddress m_iface;
/// Routing flags: valid, invalid or in search
RouteFlags m_flag;

/// List of precursors
std::vector<Ipv4Address> m_precursorList;
/// When I can send another request
Time m_routeRequestTimout;
/// Number of route requests
uint8_t m_reqCount;
/// Indicate if this entry is in "blacklist"
bool m_blackListState;
/// Time for which the node is put into the blacklist
Time m_blackListTimeout;

uint16_t node_stat; //1705087
```

MODIFICATIONS MADE FOR DEFEND ATTACK

Traverse Routing table and Calculate Threshold value Sh :

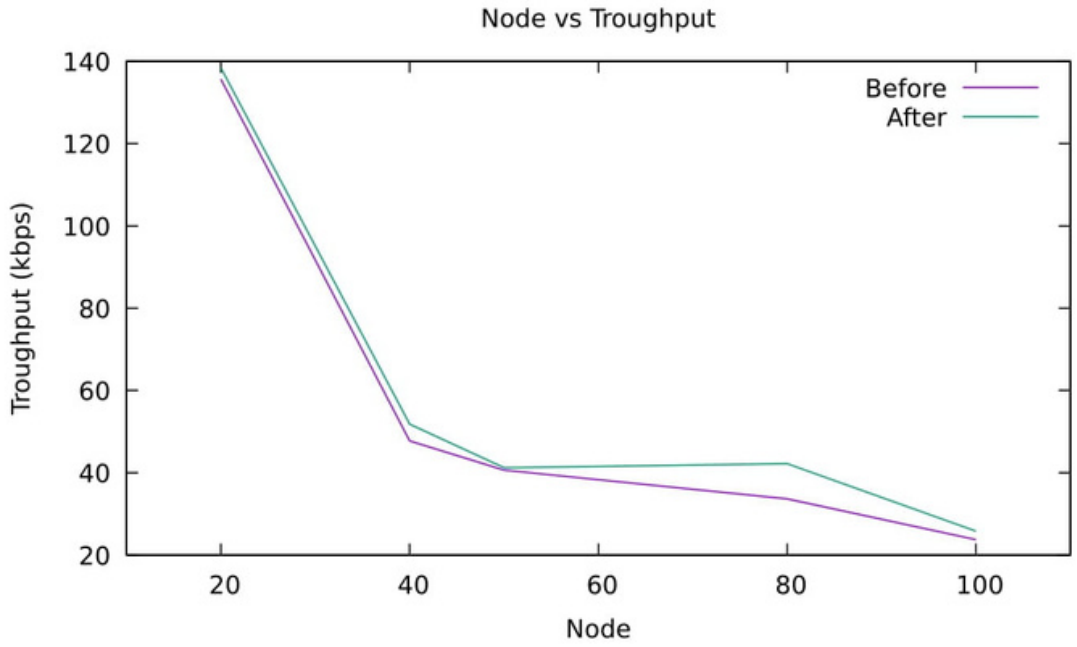
```
//1705087 - Sh calculation
int
RoutingTable::CalculateSh ()
{
  NS_LOG_FUNCTION ("Sh");

  int max_seq_no = 0;

  for (std::map<Ipv4Address, RoutingTableEntry>::iterator i =
        m_ipv4AddressEntry.begin (); i != m_ipv4AddressEntry.end (); ++i)
    {
      if(i->second.GetFlag ()>max_seq_no)
      {
        max_seq_no = i->second.GetSeqNo ();
      }

    }

  NS_LOG_LOGIC ("Max seq " << max_seq_no << " in Routing Table ");
  return 2*max_seq_no; //1705087 change as pie*lamda = max_seq as seq increase one by one
}
```

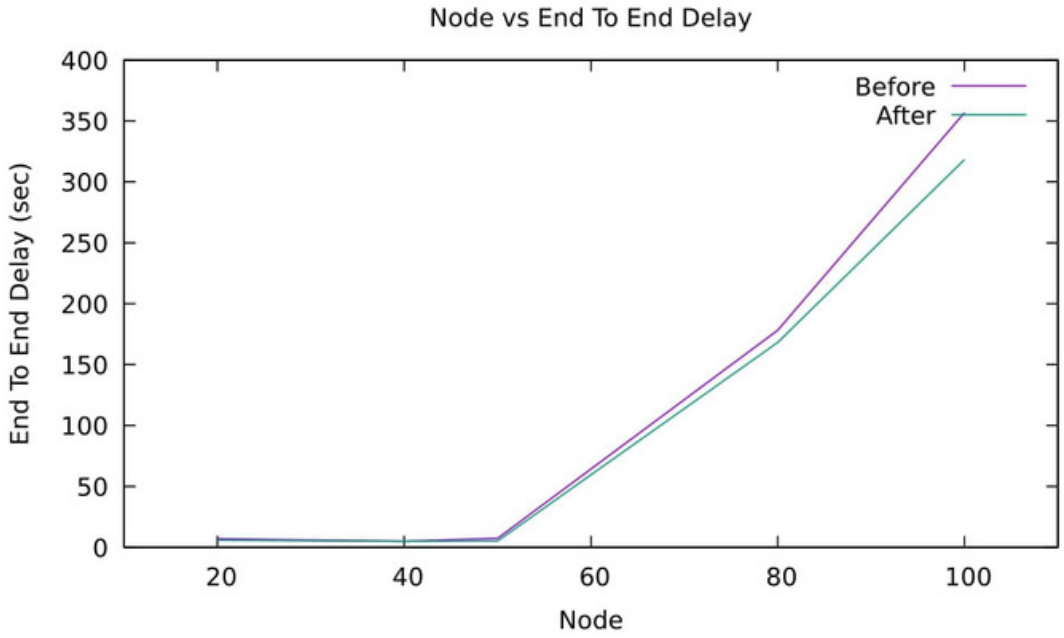
If In Sender RREP seq no is higher then Sh we will give value 1 in node state .Which mean the node is melicious and make Jop Count High :

```
//1705087 - At RecvReply -Thresold Check
uint sh = m_routingTable.CalculateSh ();

if(rrepHeader.GetDstSeqno()>sh)
{
  RoutingTableEntry newEntry (
    /*device=*/ dev, /*dst=*/ dst, /*validSeqNo=*/ true, /*seqno=*/ rrepHeader.GetDstSeqno (),
    /*iface=*/ m_ipv4->GetAddress (m_ipv4->GetInterfaceForAddress (receiver), 0),/*hop=*/ 120, // Increase Ho
    /*nextHop=*/ sender, /*lifeTime=*/ rrepHeader.GetLifeTime (),1);
}
```

RESULTS WITH GRAPHS (FOR TASK B)

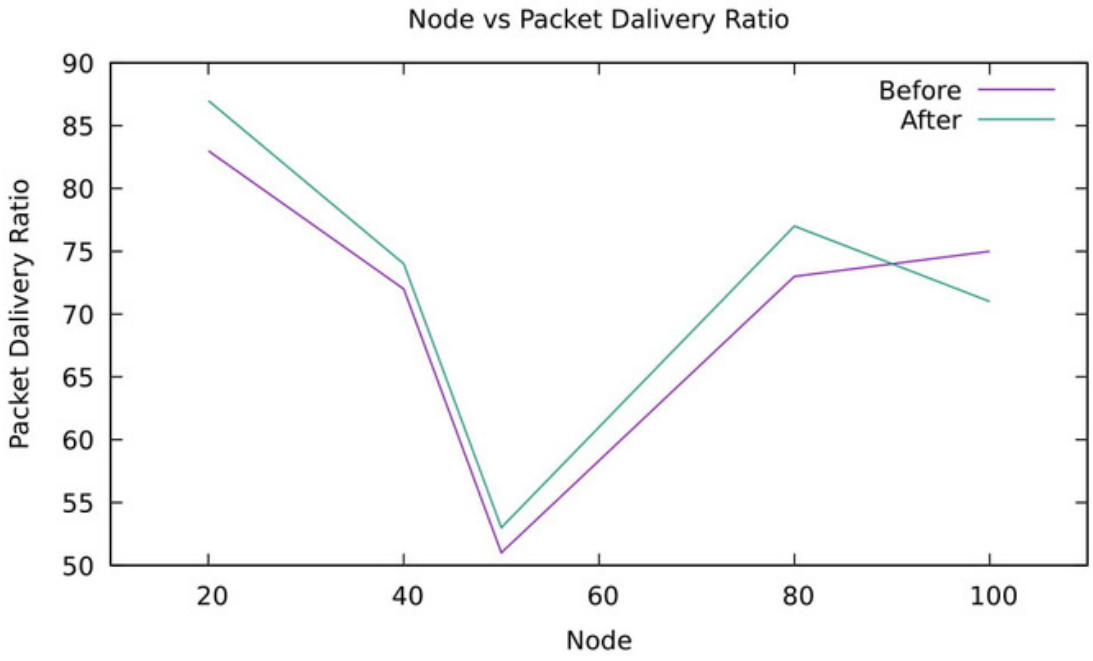
DEFENDING BLACK HOLE ATTAK

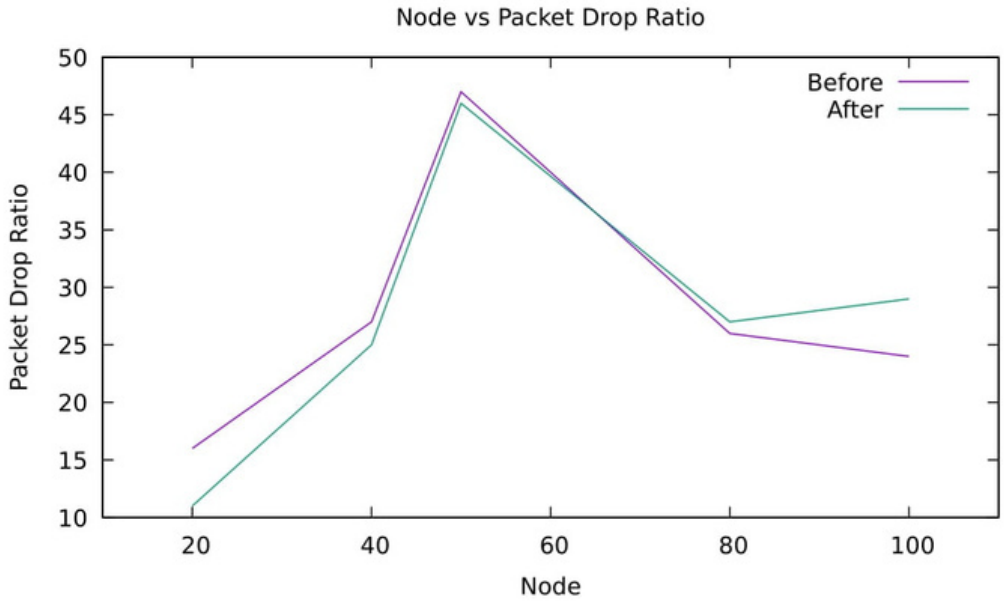




RESULTS WITH GRAPHS (FOR TASK B)

DEFENDING BLACK HOLE ATTAK

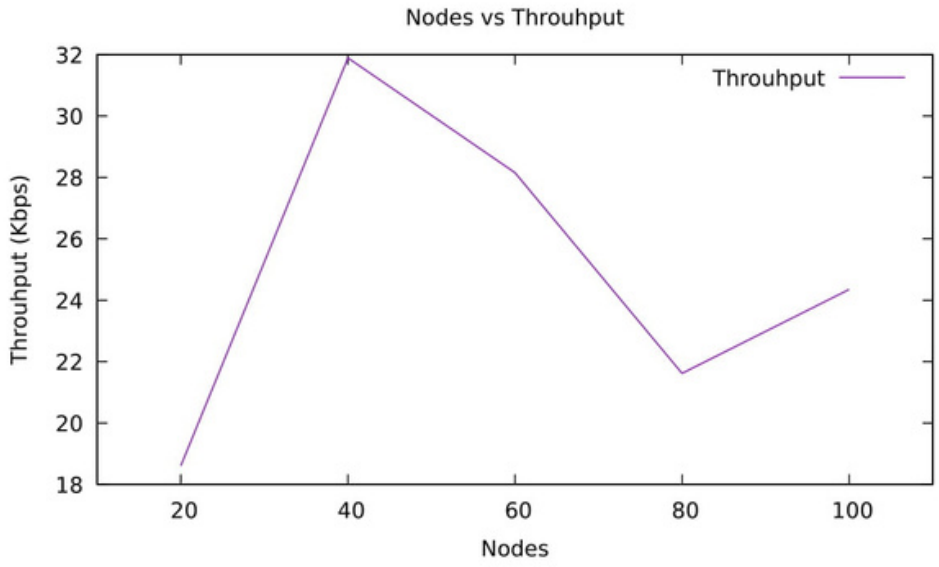


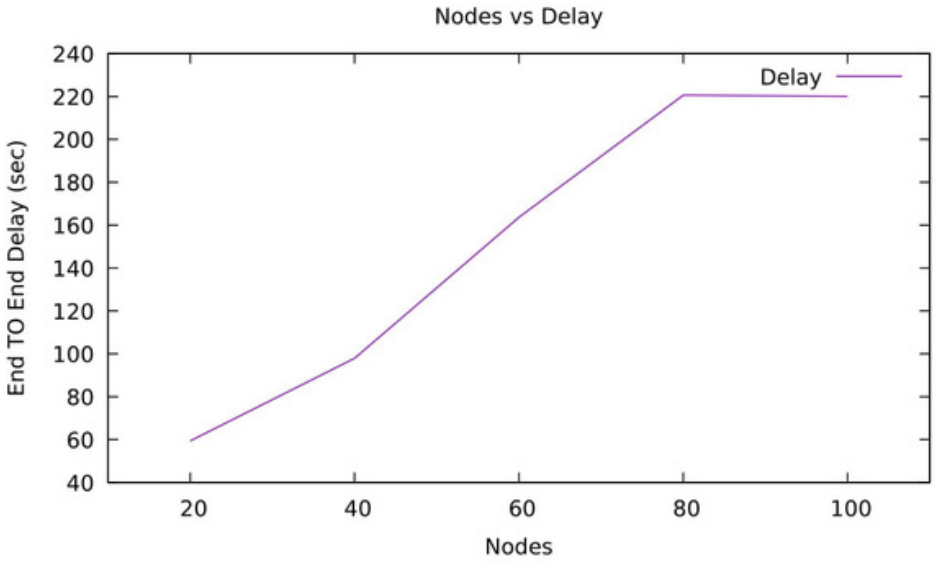


RESULTS WITH (NODES)

GRAPHS (FOR TASK A1)

WIRELESS HIGH-RATE (802.11) (MOBILE

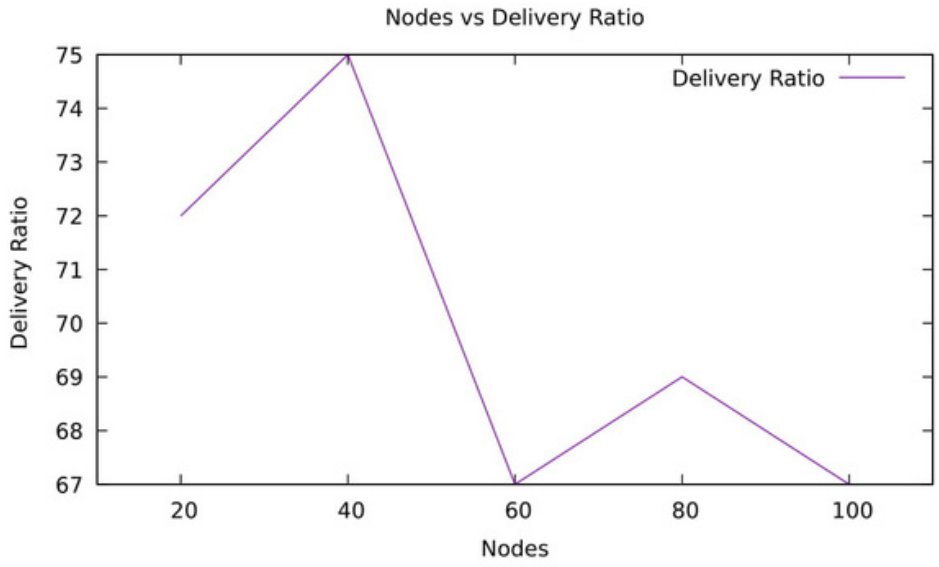


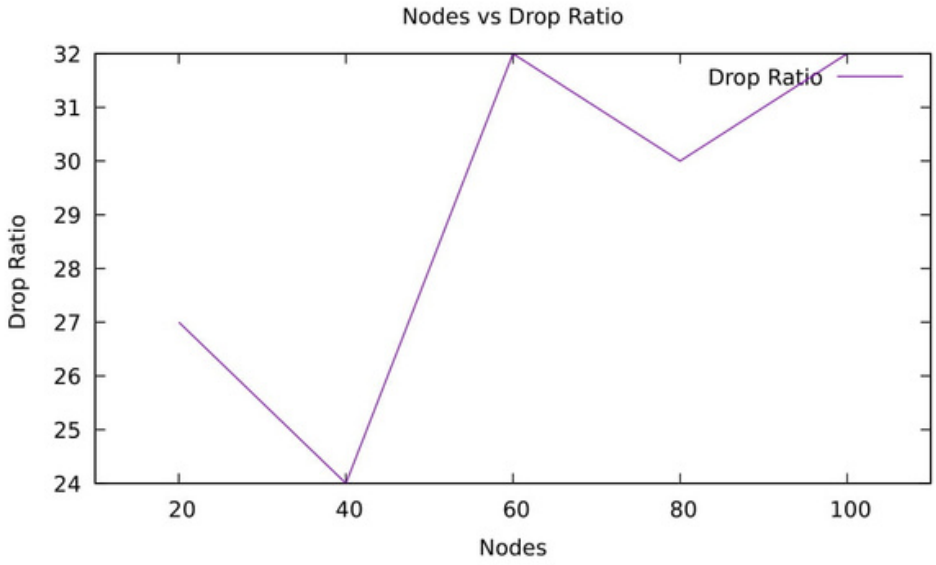


RESULTS WITH (NODES)

GRAPHS (FOR TASK A1)

WIRELESS HIGH-RATE (802.11) (MOBILE

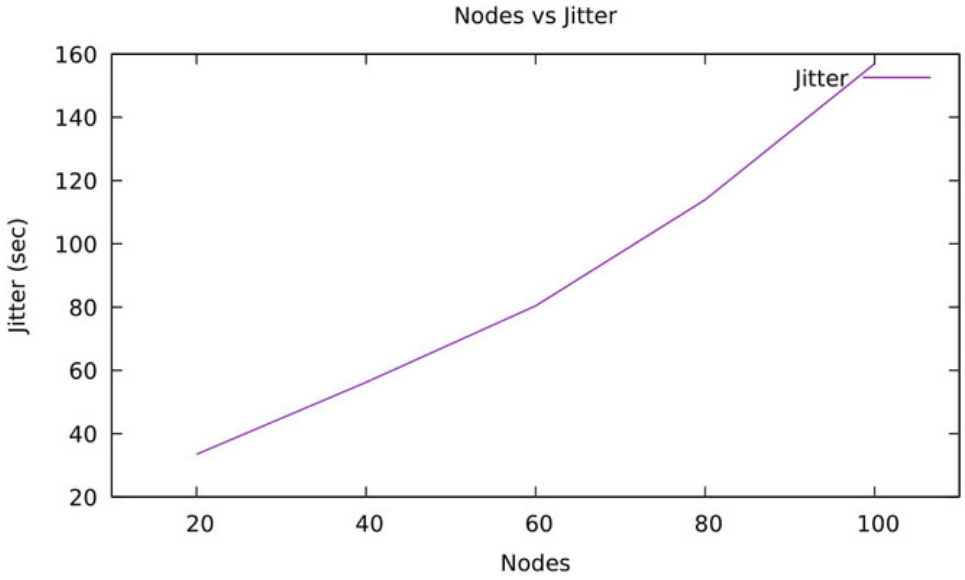


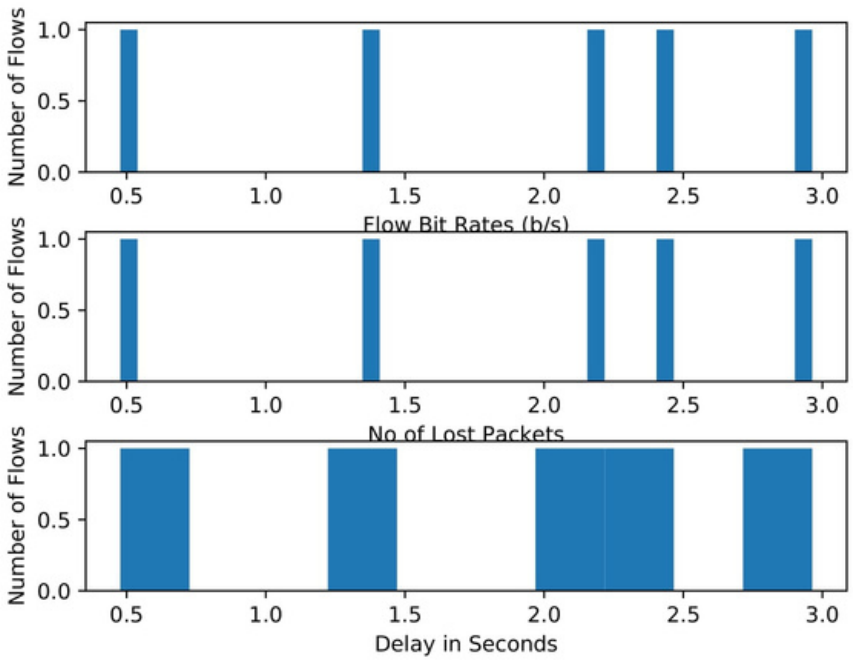


RESULTS BONUS (NODES)

GRAPHS (FOR TASK A1)

WIRELESS HIGH-RATE (802.11) (MOBILE

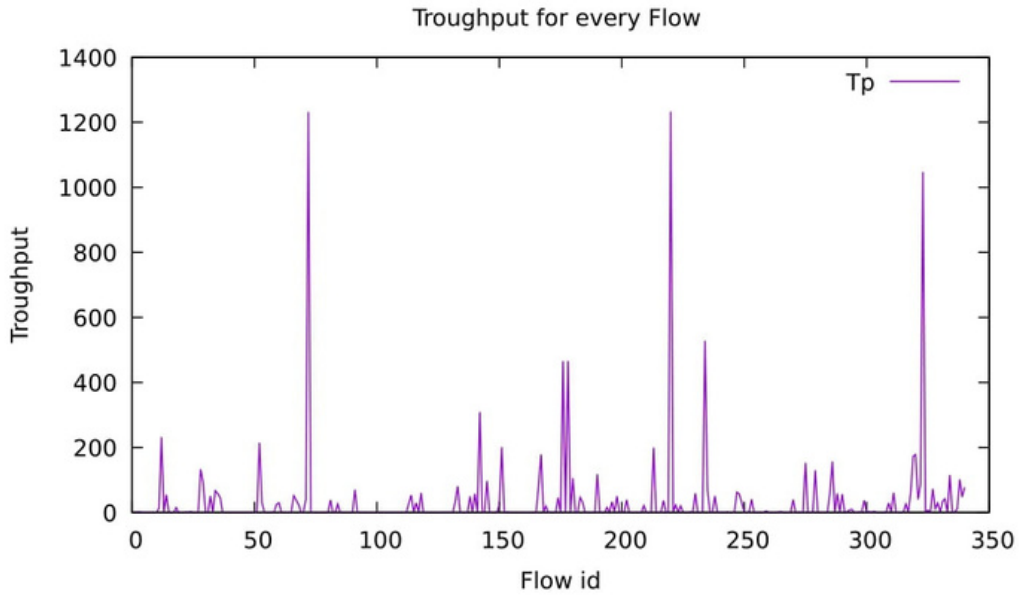


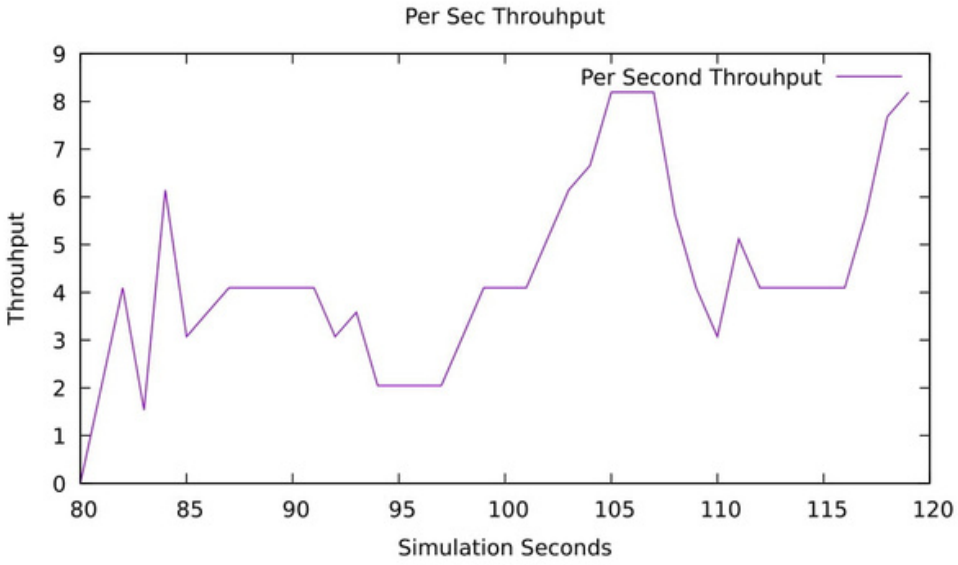


RESULTS BONUS (NODES)

GRAPHS (FOR TASK A1)

WIRELESS HIGH-RATE (802.11) (MOBILE

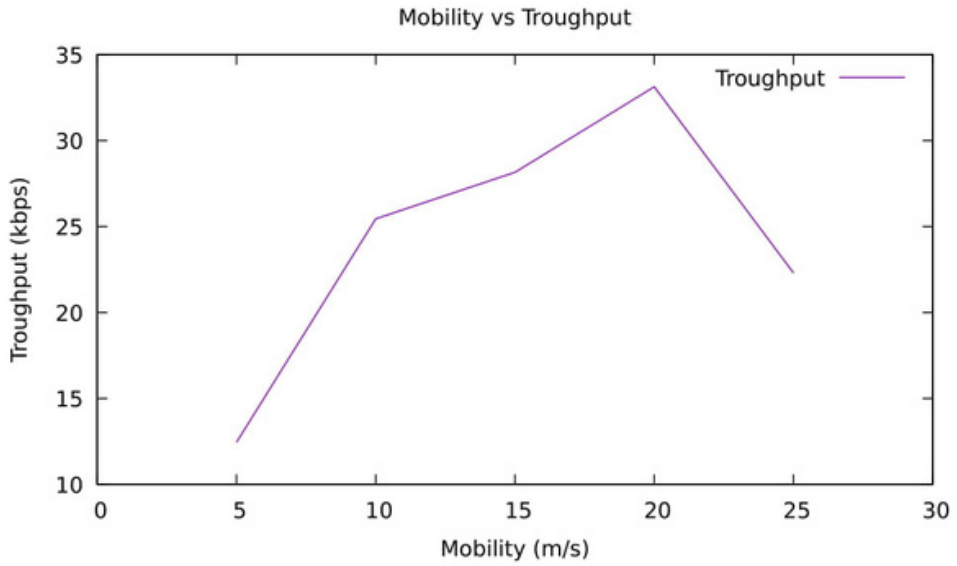


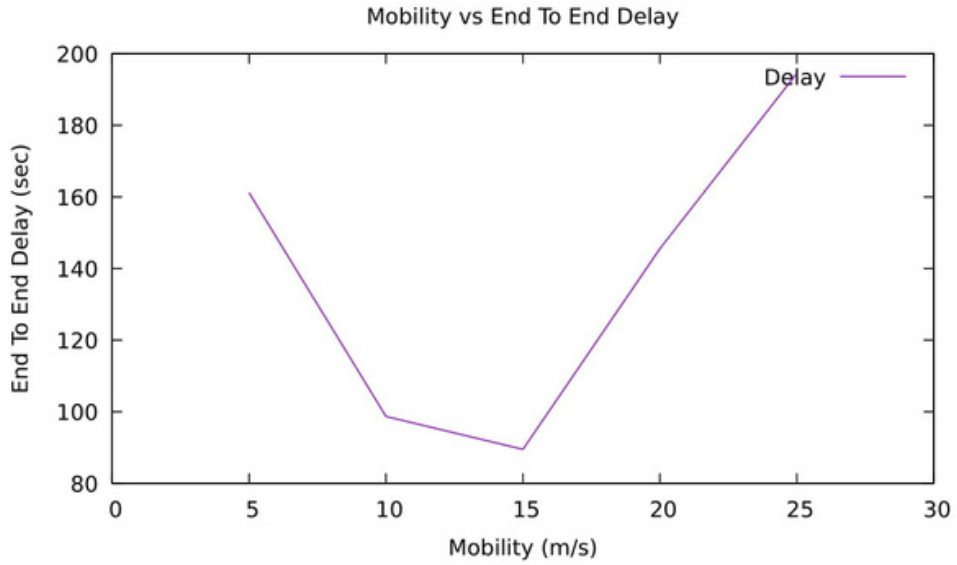


RESULTS WITH (SPEED)

GRAPHS (FOR TASK A1)

WIRELESS HIGH-RATE (802.11) (MOBILE

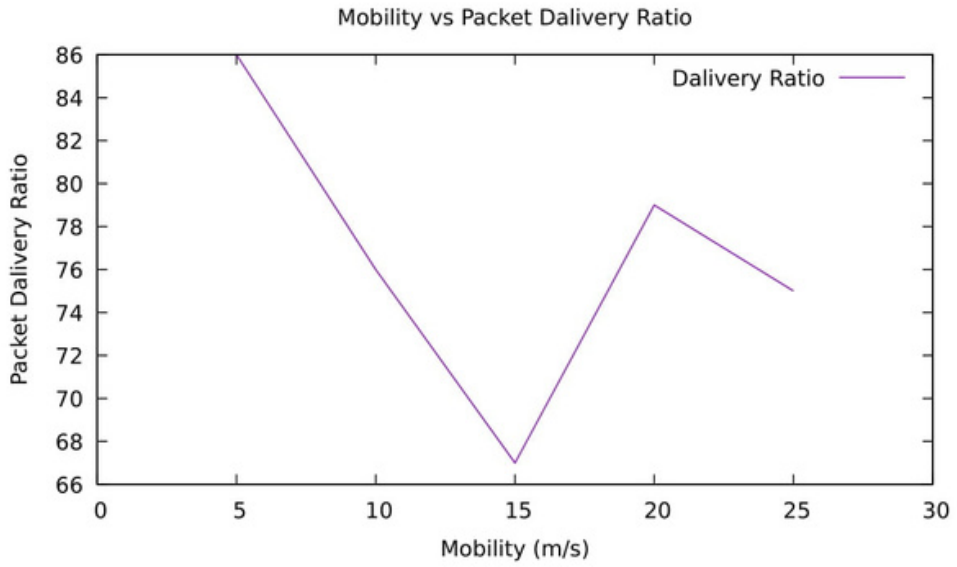


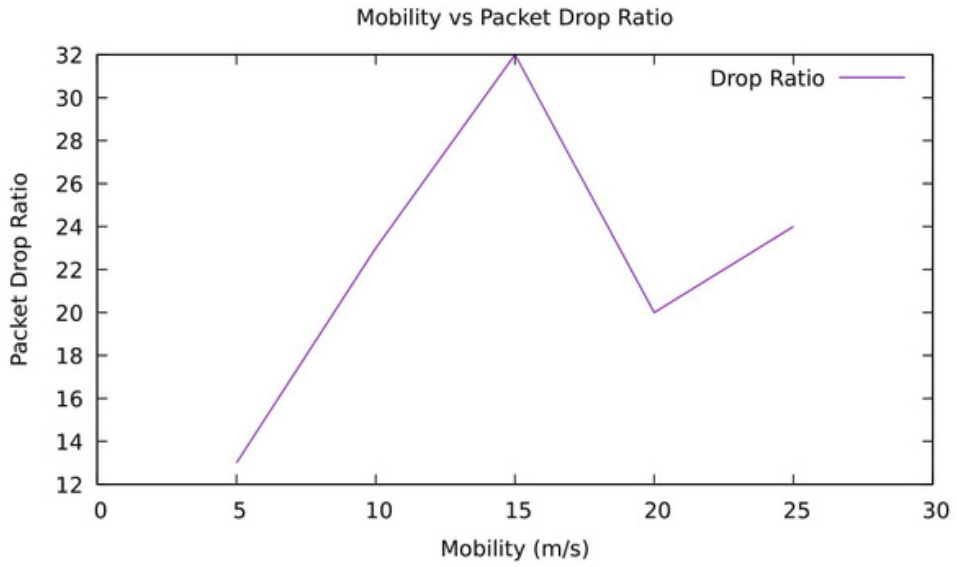


RESULTS WITH (SPEED)

GRAPHS (FOR TASK A1)

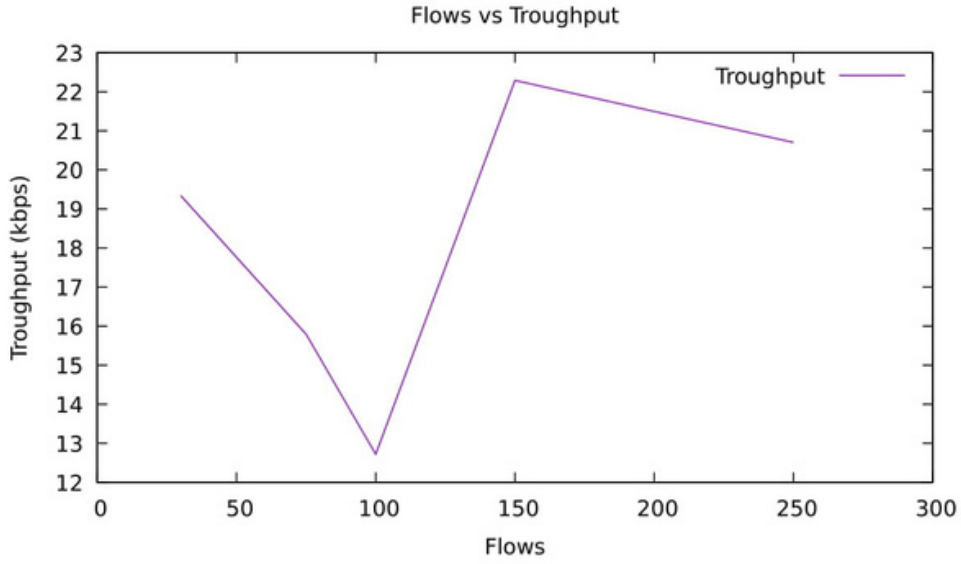
WIRELESS HIGH-RATE (802.11) (MOBILE

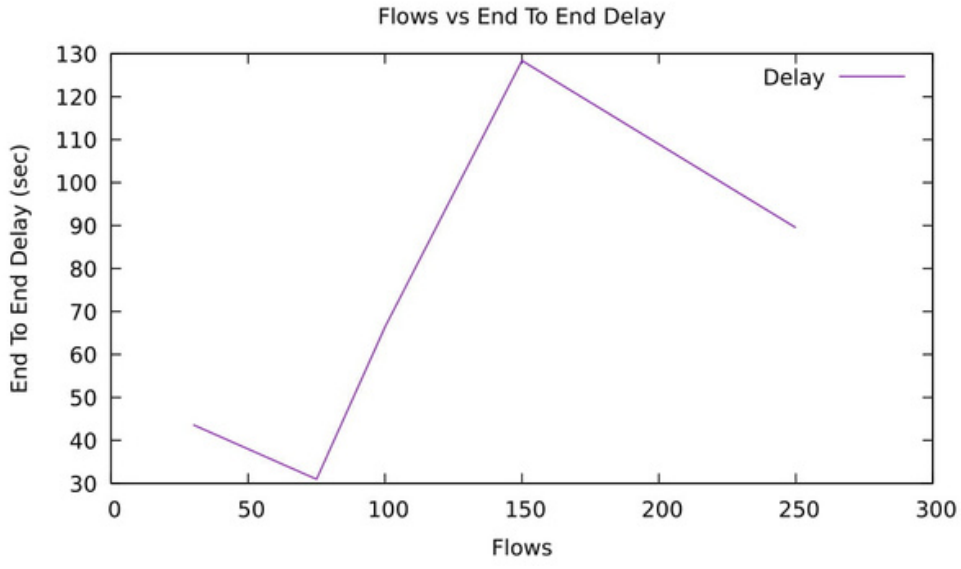




RESULTS WITH (FLOW) GRAPHS (FOR TASK A1)

WIRELESS HIGH-RATE (802.11) (MOBILE

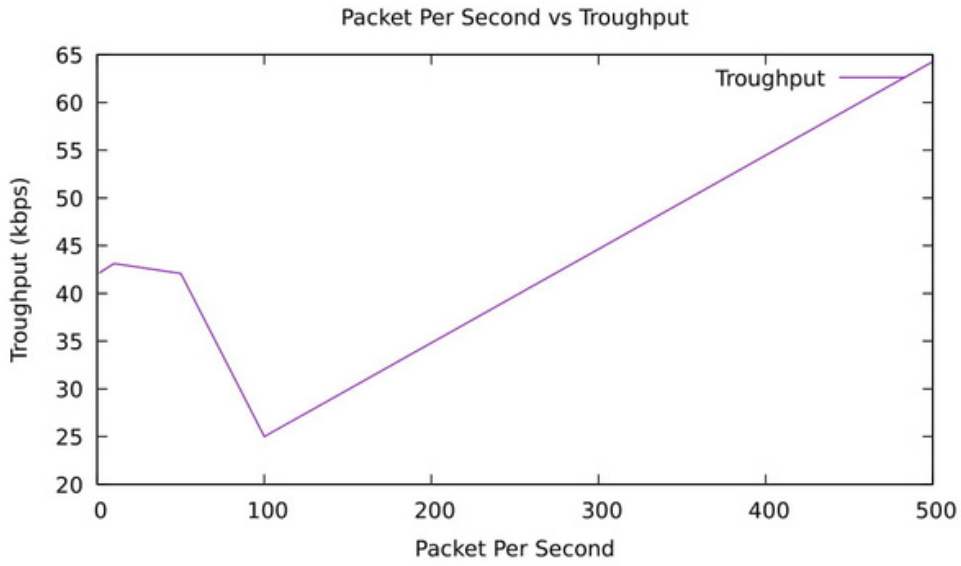


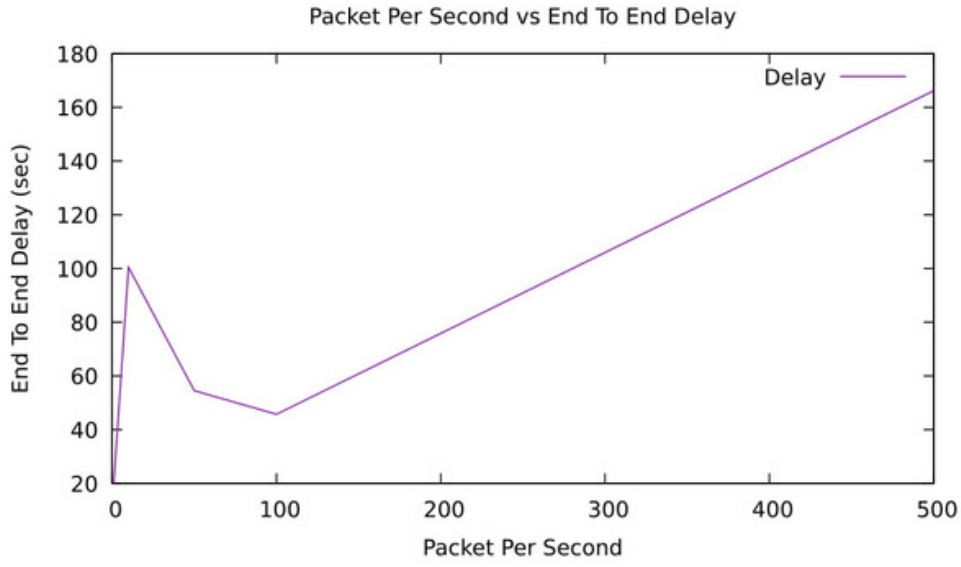


RESULTS WITH (PPS)

GRAPHS (FOR TASK A1)

WIRELESS HIGH-RATE (802.11) (MOBILE

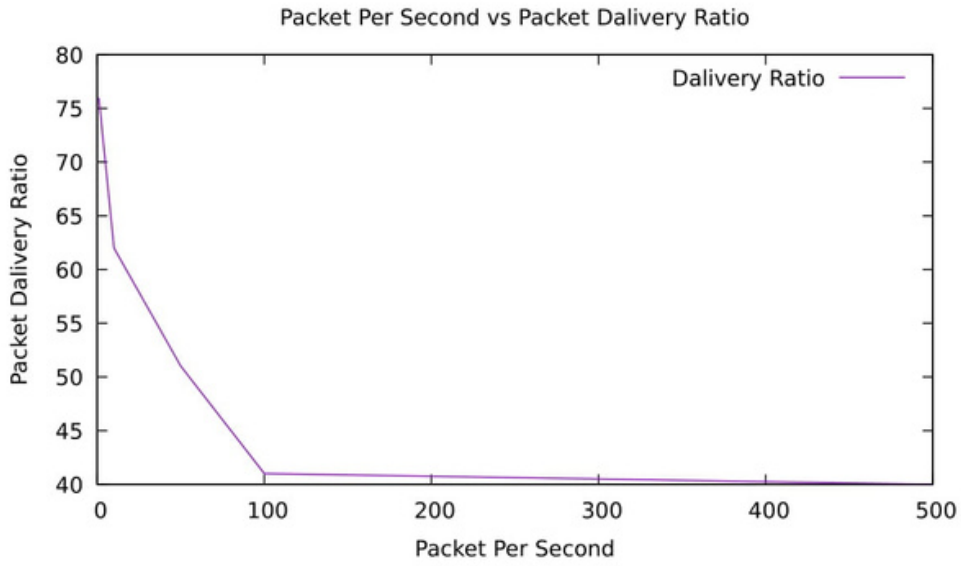


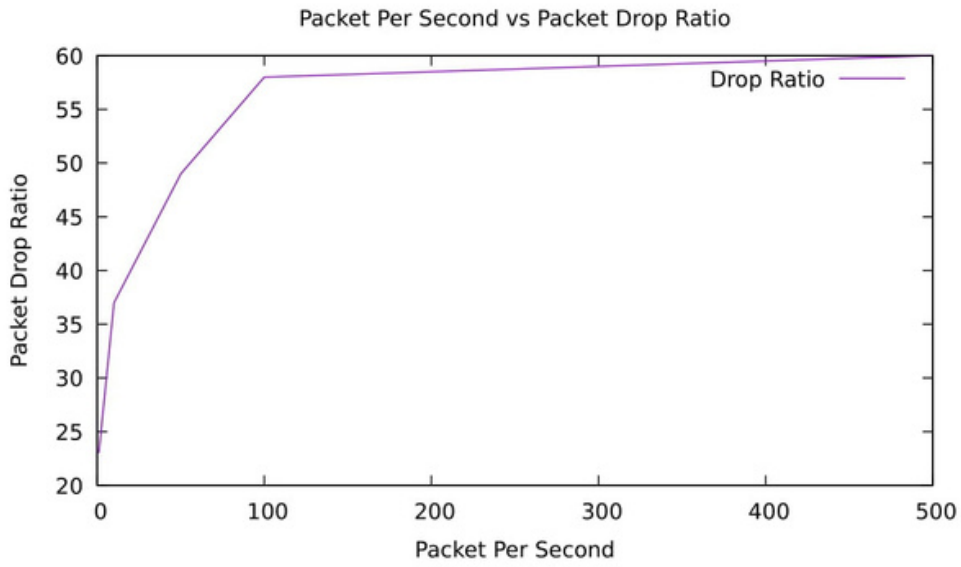


RESULTS WITH (PPS)

GRAPHS (FOR TASK A1)

WIRELESS HIGH-RATE (802.11) (MOBILE

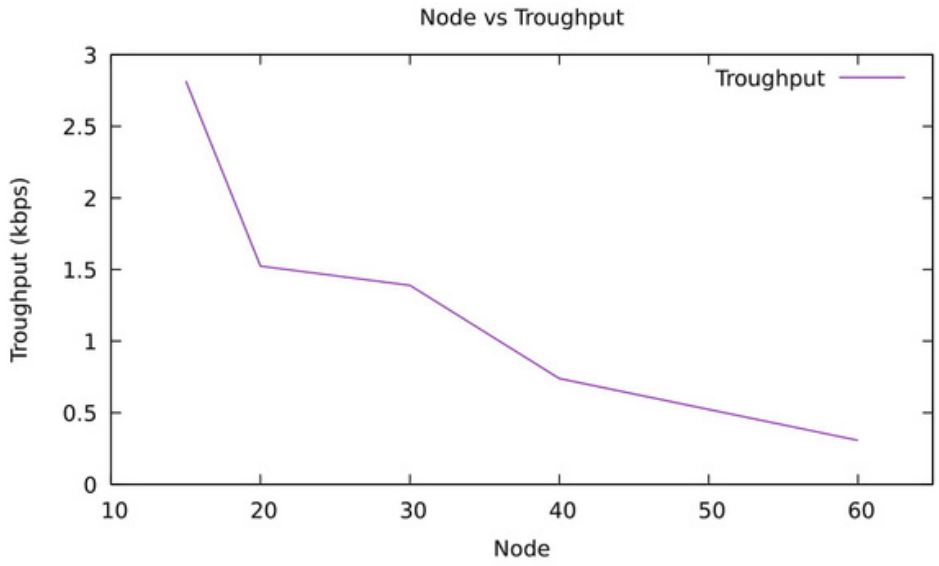


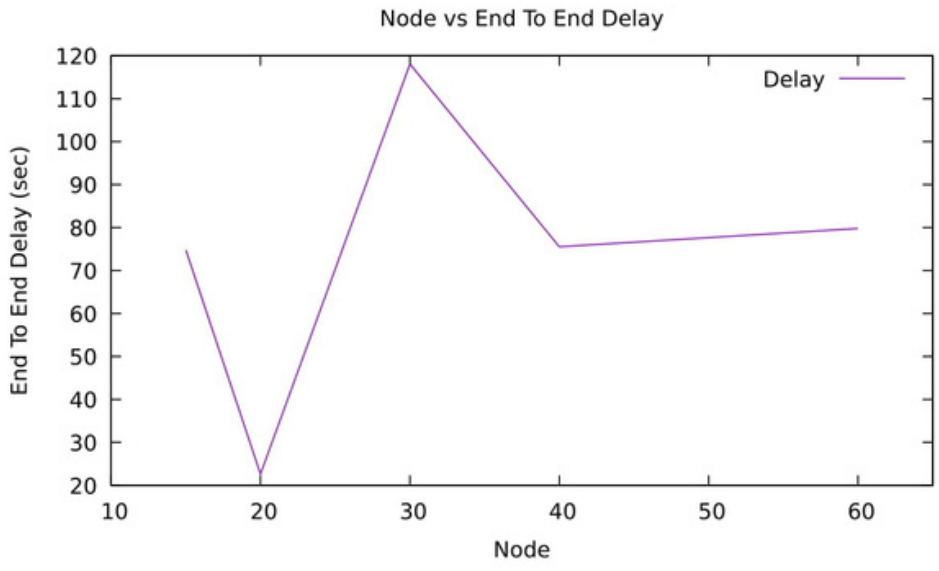


RESULTS WITH (NODES)

GRAPHS (FOR TASK A2)

WIRELESS LOW-RATE (E802.15.4) (MOBILE

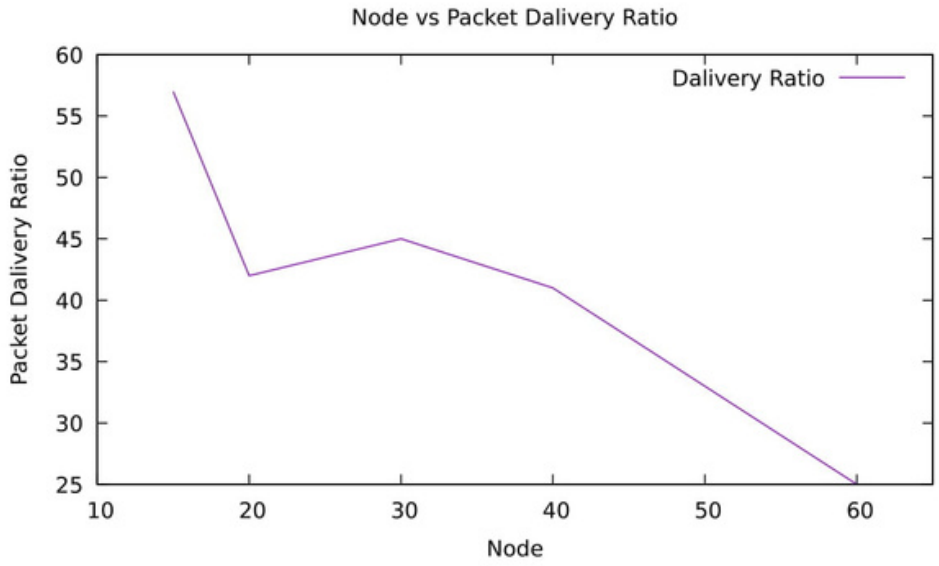


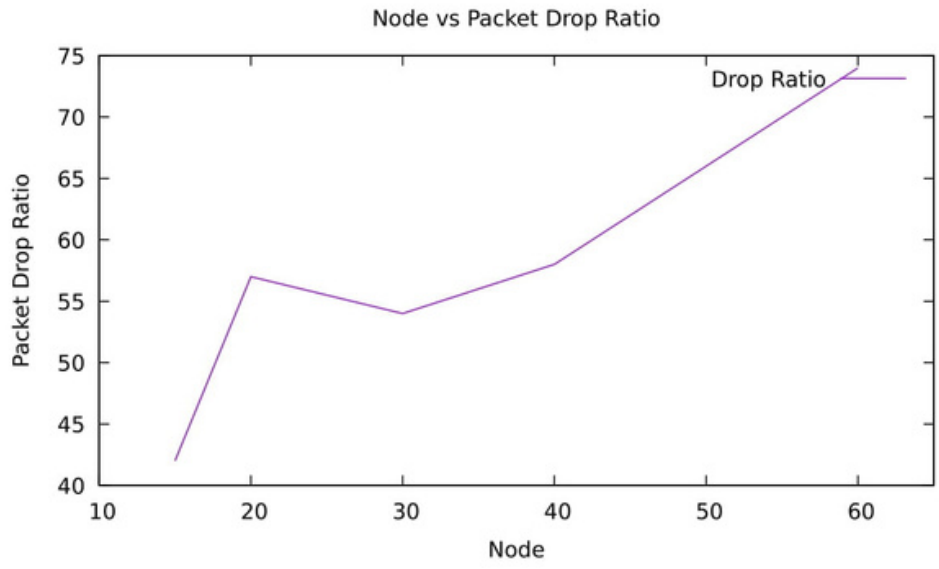


RESULTS WITH (NODES)

GRAPHS (FOR TASK A2)

WIRELESS LOW-RATE (E802.15.4) (MOBILE

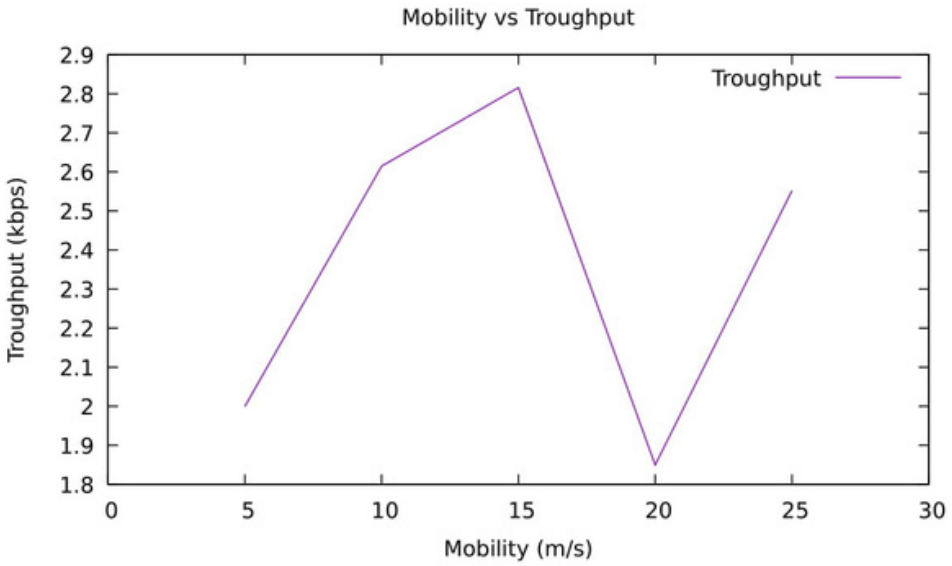


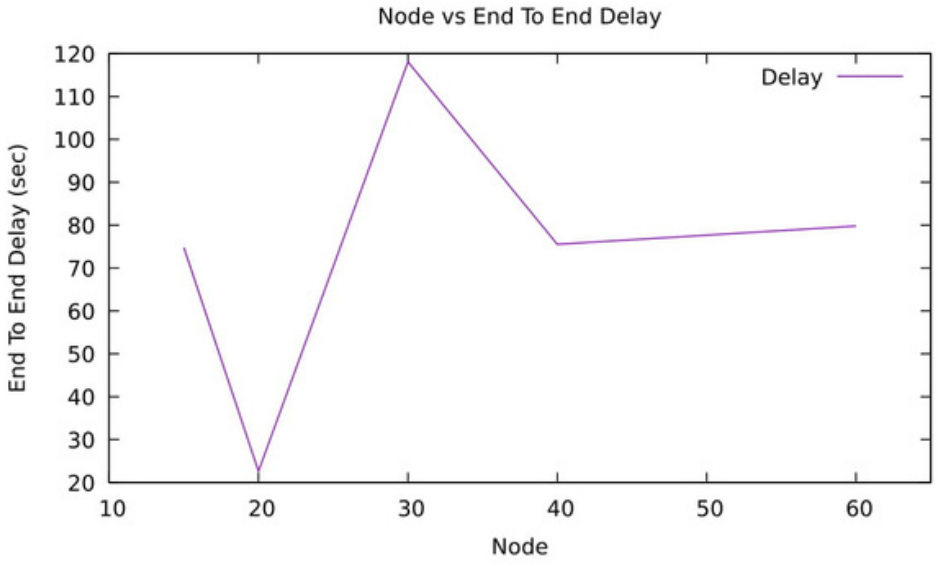


RESULTS WITH (SPEED)

GRAPHS (FOR TASK A2)

WIRELESS LOW-RATE (E802.15.4) (MOBILE

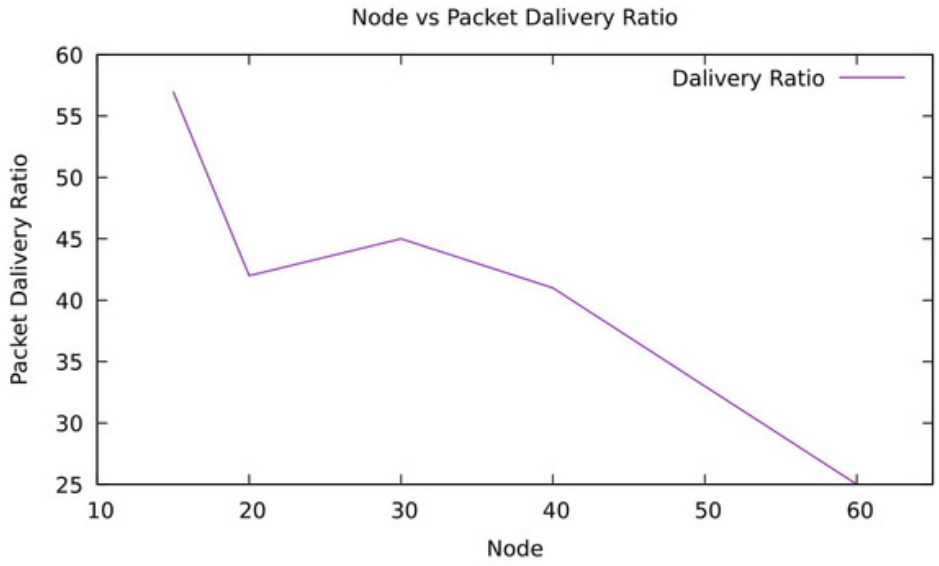


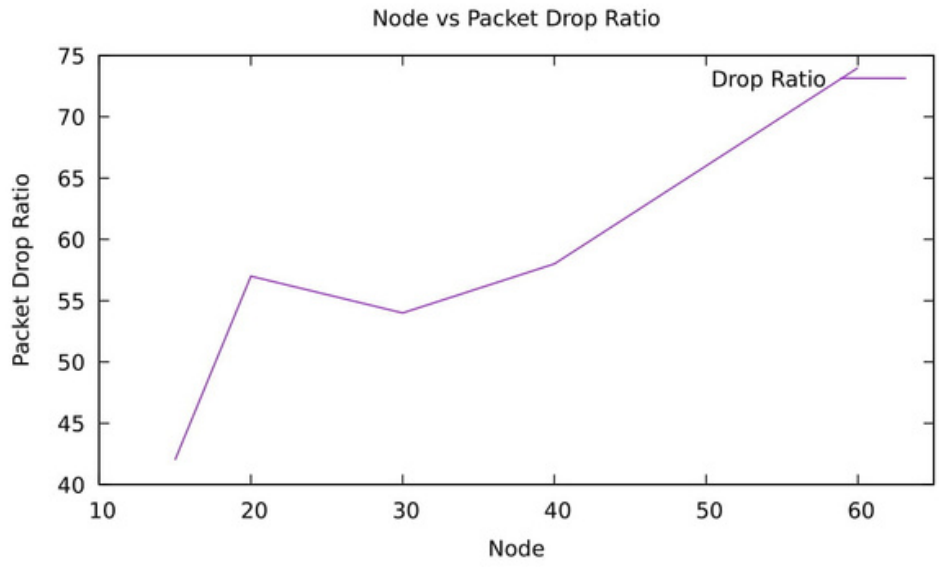


RESULTS WITH (SPEED)

GRAPHS (FOR TASK A2)

WIRELESS LOW-RATE (E802.15.4) (MOBILE

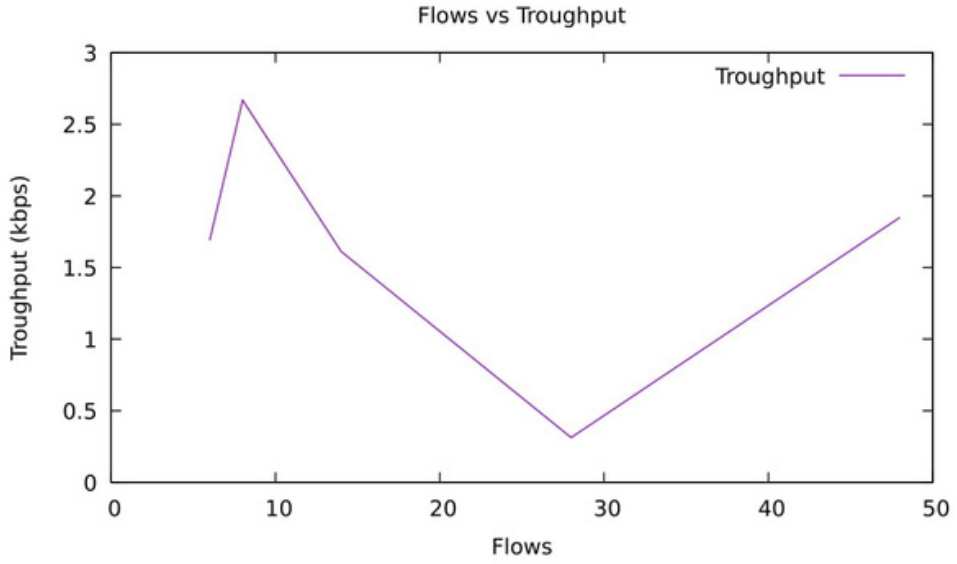


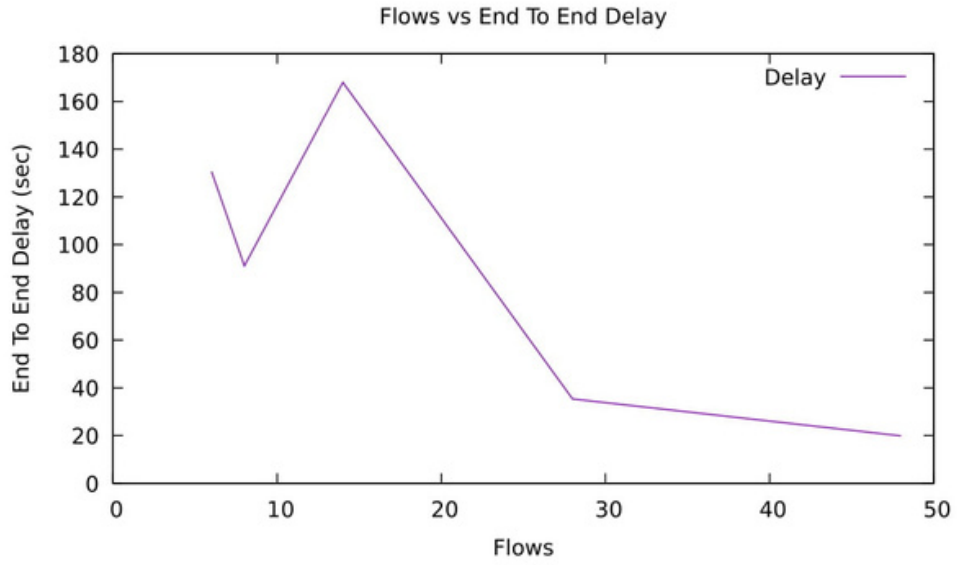


RESULTS WITH (FLOWS)

GRAPHS (FOR TASK A2)

WIRELESS LOW-RATE (E802.15.4) (MOBILE

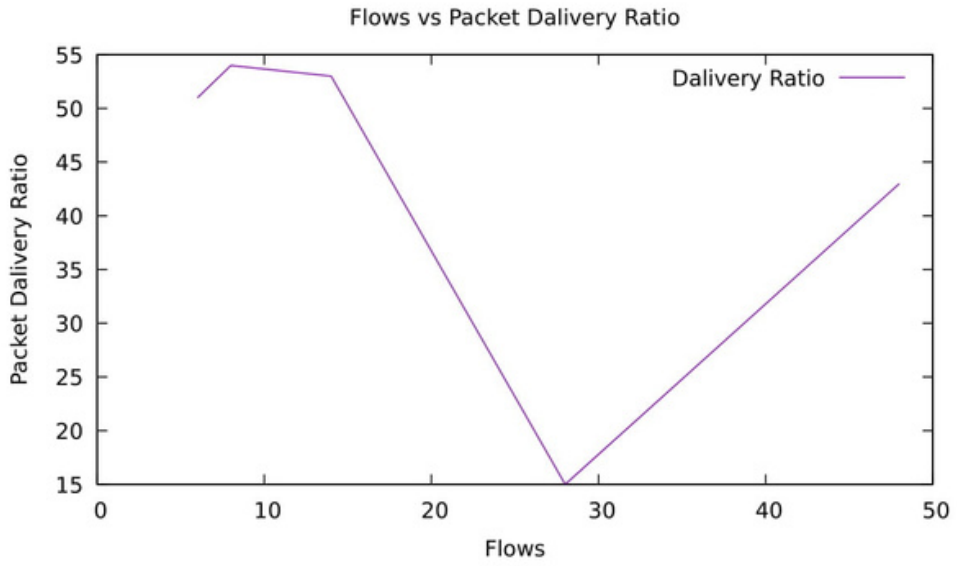


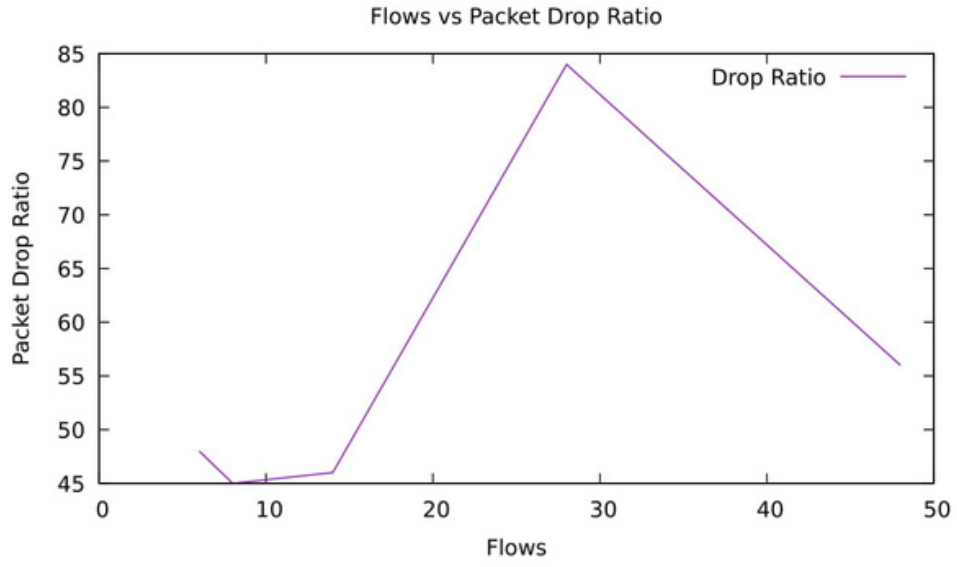


RESULTS WITH (FLOWS)

GRAPHS (FOR TASK A2)

WIRELESS LOW-RATE (E802.15.4) (MOBILE

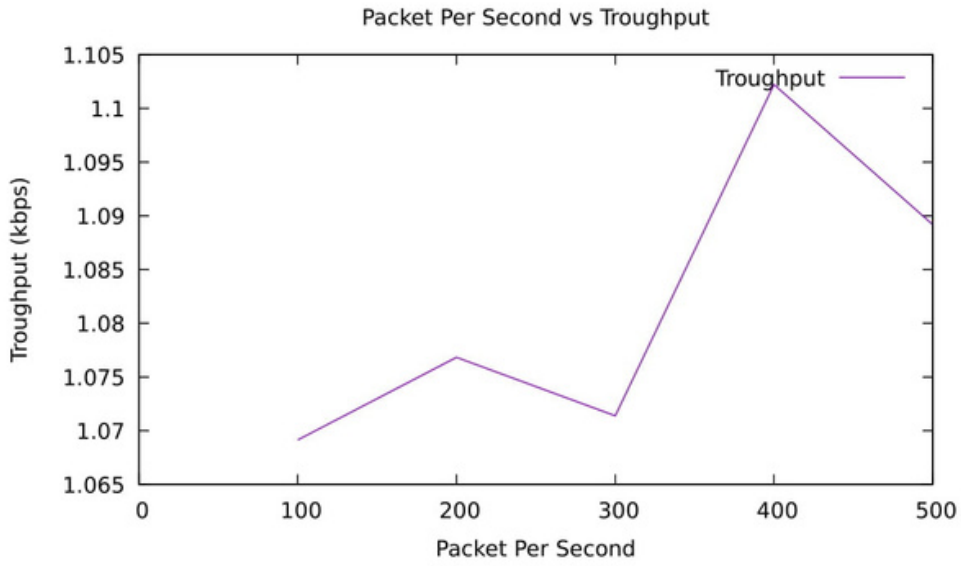


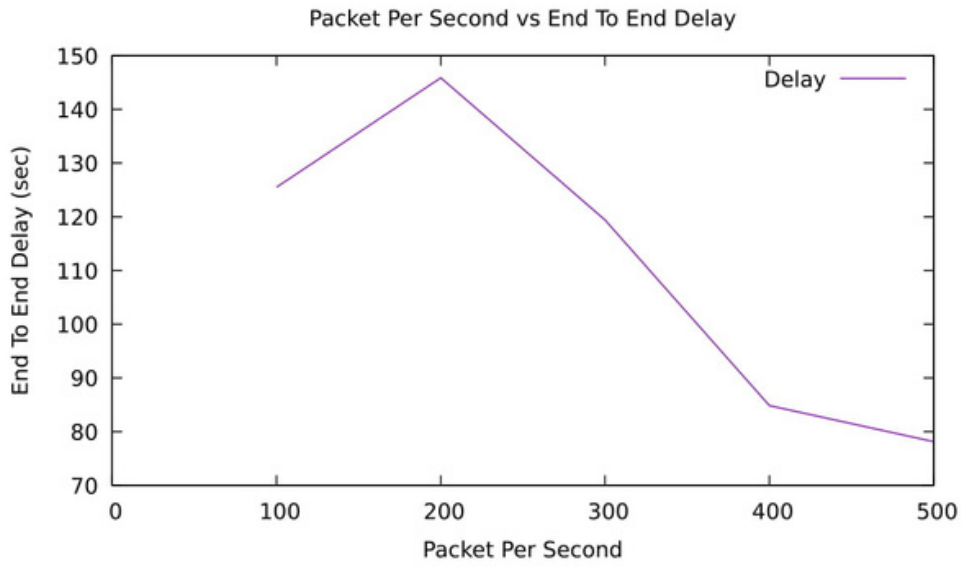


RESULTS WITH (PPS)

GRAPHS (FOR TASK A2)

WIRELESS LOW-RATE (E802.15.4) (MOBILE

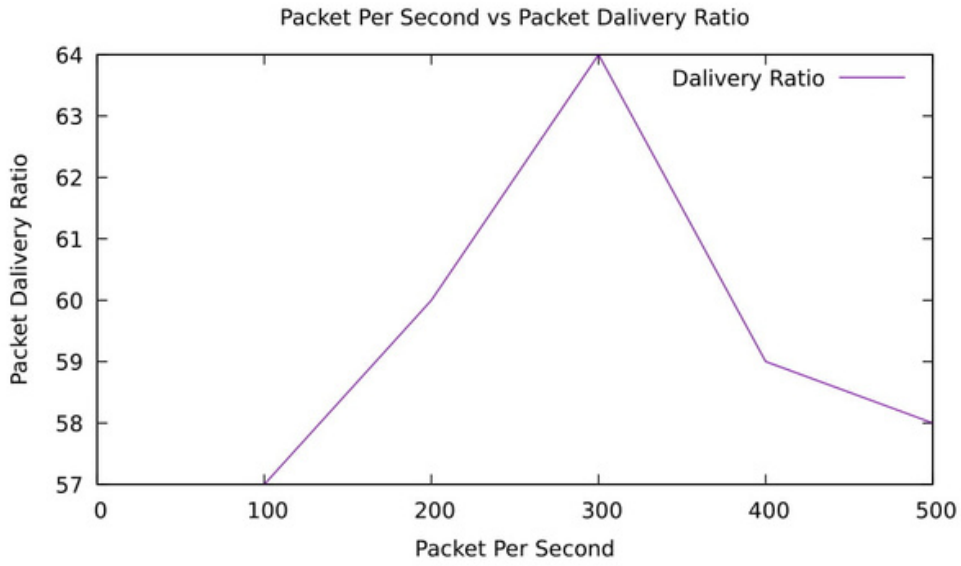


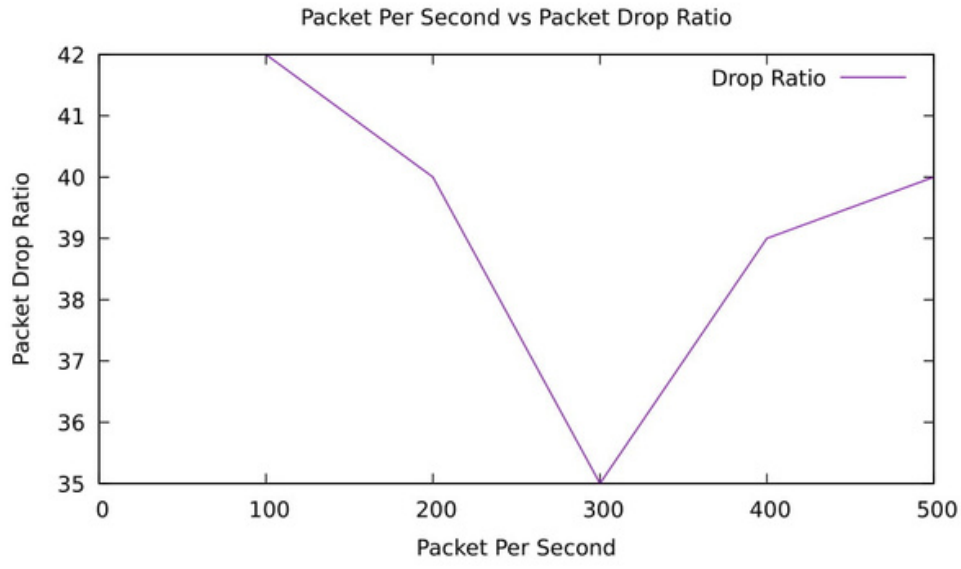


RESULTS WITH (PPS)

GRAPHS (FOR TASK A2)

WIRELESS LOW-RATE (E802.15.4) (MOBILE





SUMMARY FINDINGS EXPLAINING THE RESULTS FOUND IN TASK B

The performance metrices Decreases when Black Hole Attack introduced in our networks .But there are improvements in performance metrices due to improving ADOV protocol . Sadly , the improvements are not up to the mark . The reasons behind it given bellow :

-Many simplifications applied
-Ideal Environment not set up
-Not applying proper threshold value

SUMMARY FINDINGS EXPLAINING THE RESULTS FOUND IN TASK A

In task A, we have to simulate two different kinds of networks - Wireless high rate (mobile) and Wireless low rate (mobile) .The performance metrices Increases when node , mobility increase but it decrease when flows and PPS increases .This scenario reverses if we reverse the order of the parameters. But there are less improvements in performance metrices because it send the dates at low rate . Sadly , the model for Low rate in Ns3 are not up to the mark , The model-802.15.4 has not been validated against real hardware. So, there can be several loophole that is not known yet . There seems to be a lot packet drop in case of low rate data transfer. They may be dropped due to excessive transmission retries or channel access failure. Data taking or human error also can occurs here