BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY

Report
on

NS3 Simulation

Course: Computer Network (CSE 322)

Submitted from:
Tanjim Ahmed Khan
Student ID: 1705032

Supervised by:
Syed Md. Mukit Rashid
Lecturer, CSE, BUET

Submission Date: 23 February, 2022

Network Topologies under Simulation

1. Dumbbell topology [Task A (Wireless high-rate) and Task B]
2. Mesh topology [Task A (Wireless low-rate)]

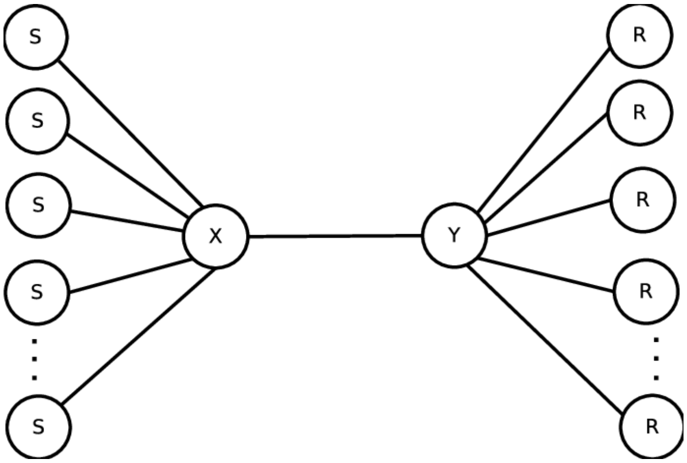


Fig 1: Dumbbell Topology

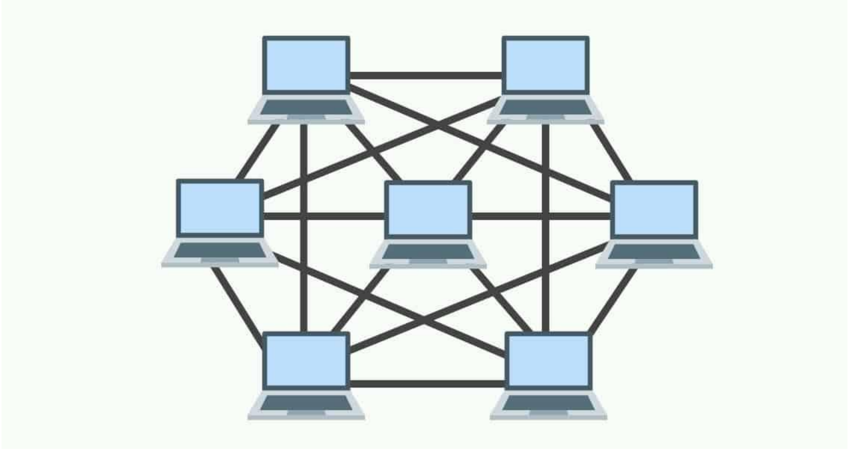


Fig 2: Mesh Topology

Parameters under Variation

- Task A
  - Number of nodes
  - Number of flows
  - Packets per second
  - Coverage range
- Task B
  - Number of nodes

Overview of Proposed Algorithm

The proposed algorithm is basically a modification of Jacobson's algorithm for calculating round trip time (RTT). The original algorithm is as follows:

$$PRTT_{n+1} = α * PRTT_n + (1 - α) * ARTT_n$$
$$PD_{n+1} = ß * PD_n + (1 - ß) * AD_n$$
$$\text{where, } AD_n = | PRTT_n - ARTT_n |$$
$$RTO_{n+1} = 4 * PD_{n+1} + PRTT_{n+1}$$

Here,
PRTT = predicted round trip time
ARTT = actual round trip time
PD = predicted deviation
AD = actual deviation
RTO = retransmission timeout
α, ß = smoothing factors

This algorithm takes into account the value of RTT, without considering the tendency of it. On the other hand, the changing trend of RTT reflects the traffic state of a network more.

In the original algorithm, when there is a sudden and substantial increase or decrease in RTT, RTO cannot follow RTT accordingly. So, we are going to make α and ß variable rather than constant. We will modify α and ß using the rate of change of PRTT according to the following equations:

$$k_{n+1} = (| PRTT_{n+1} - PRTT_n |) / PRTT_n$$
$$α_{n+1} = α_0 (1 + k_{n+1})$$
$$ß_{n+1} = ß_0 (1 - k_{n+1})$$

Modifications made in NS3

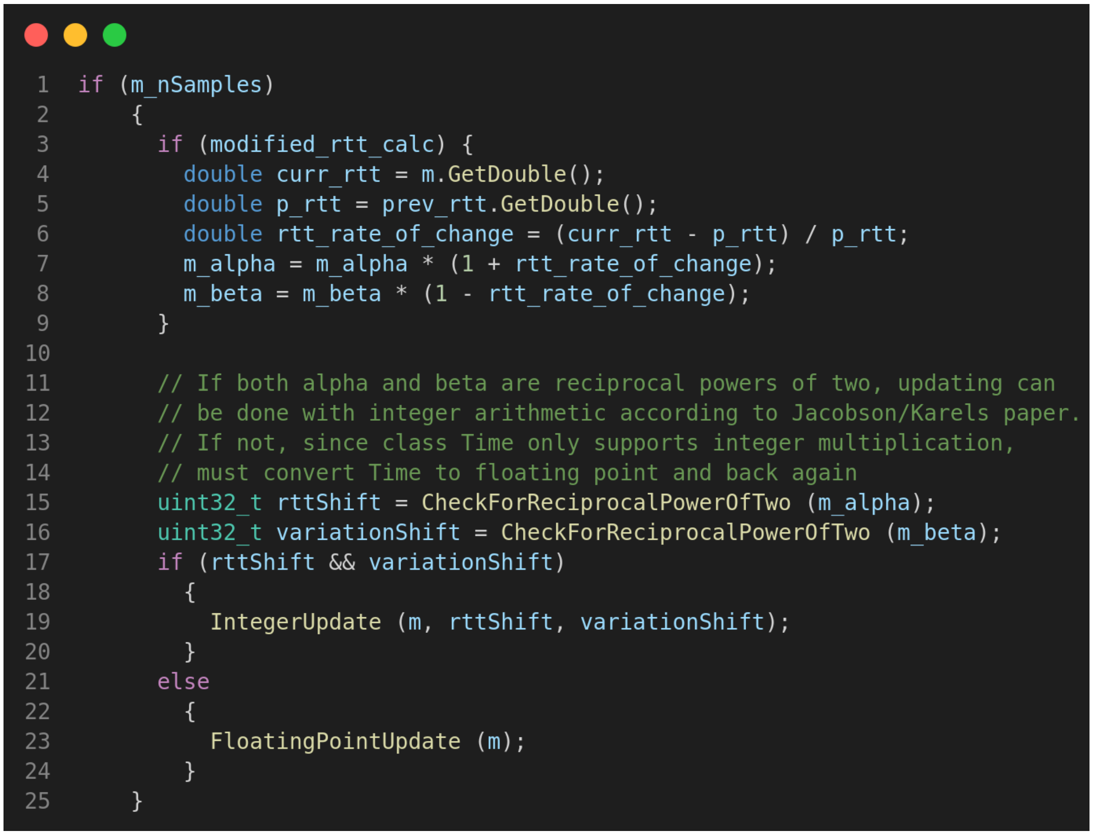
To modify the algorithm, changes have been made in the following files:

- rtt-estimator.h
- rtt-estimator.cc

Added attributes:

name	data type	description
prev_rtt	Time	for tracking last actual RTT to calculate RTT's rate of change
modified_rtt_calc	bool	for choosing between original algo and modified algo

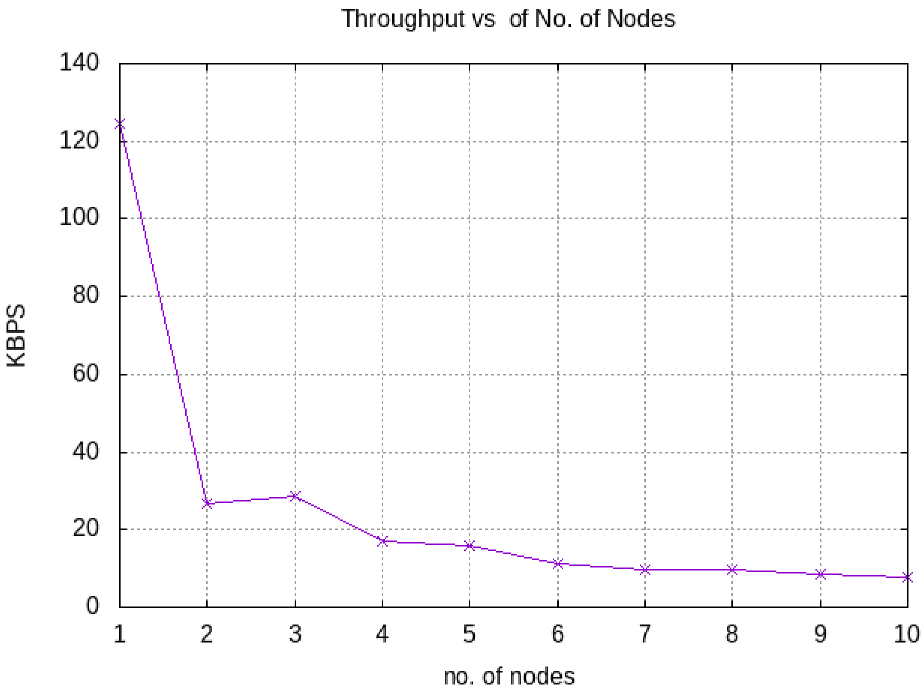
Calculation for the modification is done in `Measurement` function of `RTTMeanDeviation` class. Here is the code snippet:

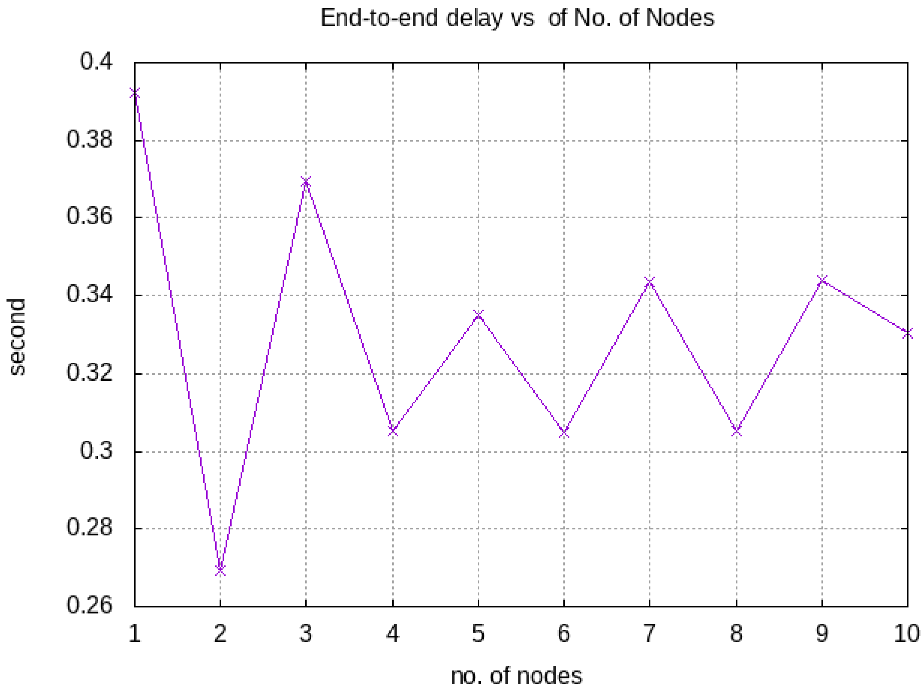


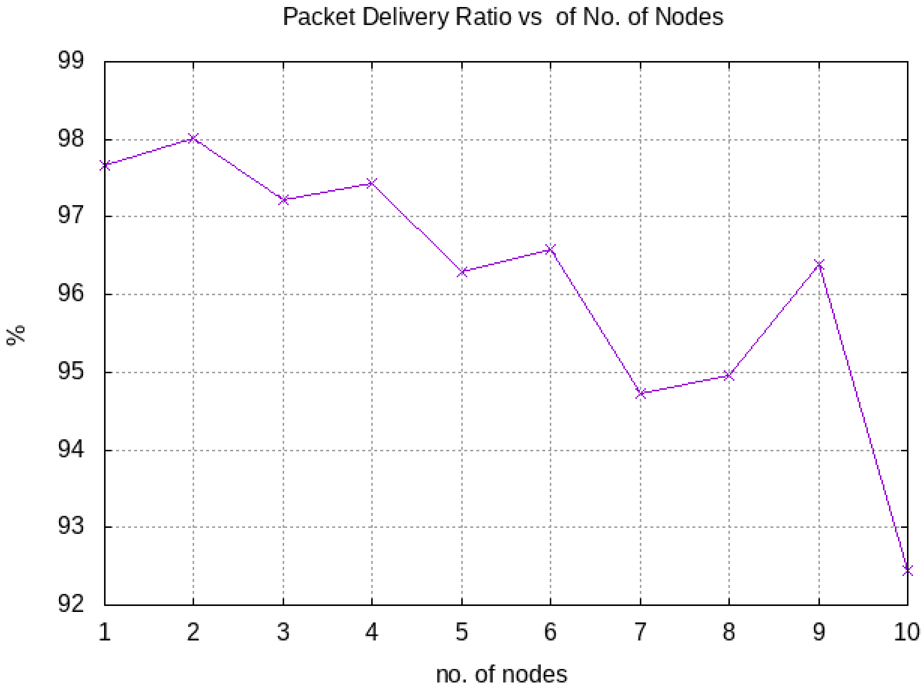
```
if (m_nSamples)
    {
      if (modified_rtt_calc) {
        double curr_rtt = m.GetDouble();
        double p_rtt = prev_rtt.GetDouble();
        double rtt_rate_of_change = (curr_rtt - p_rtt) / p_rtt;
        m_alpha = m_alpha * (1 + rtt_rate_of_change);
        m_beta = m_beta * (1 - rtt_rate_of_change);
      }

      // If both alpha and beta are reciprocal powers of two, updating can
      // be done with integer arithmetic according to Jacobson/Karels paper.
      // If not, since class Time only supports integer multiplication,
      // must convert Time to floating point and back again
      uint32_t rttShift = CheckForReciprocalPowerOfTwo (m_alpha);
      uint32_t variationShift = CheckForReciprocalPowerOfTwo (m_beta);
      if (rttShift && variationShift)
        {
          IntegerUpdate (m, rttShift, variationShift);
        }
      else
        {
          FloatingPointUpdate (m);
        }
    }
```

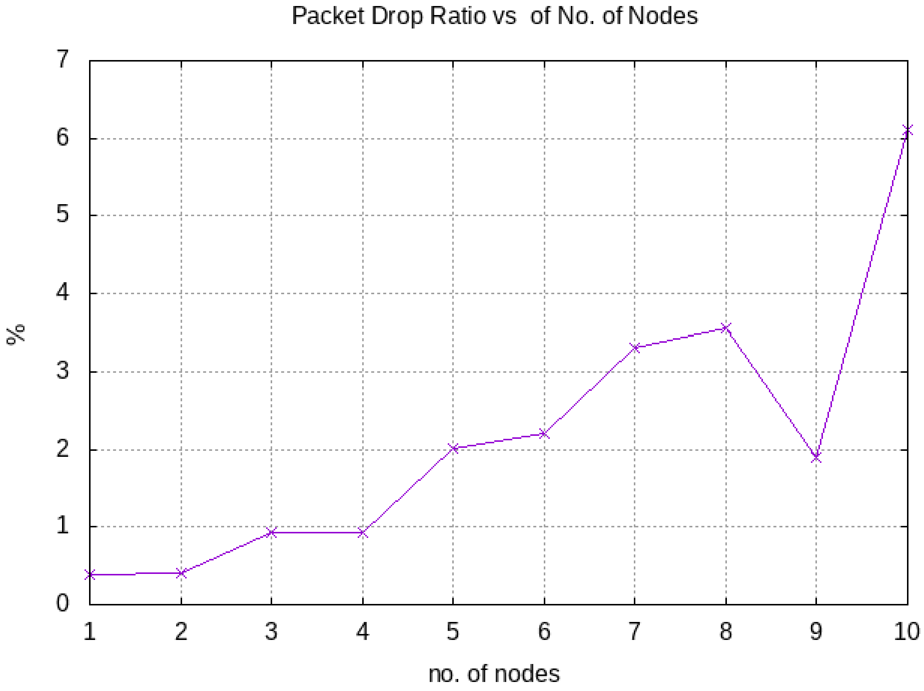
Graphs (Task A - Wireless high-rate)

Varying number of nodes:

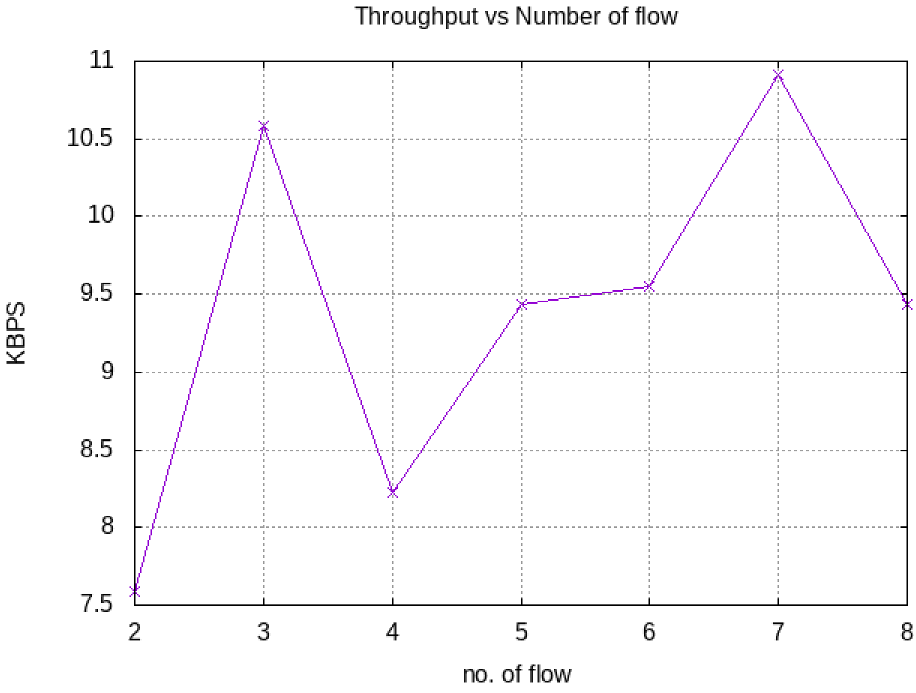


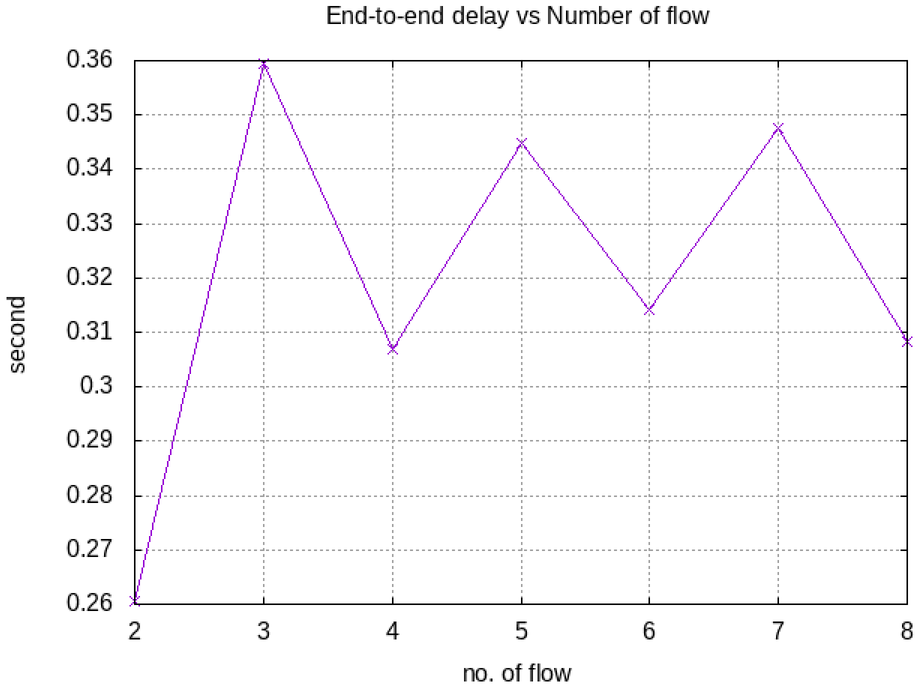


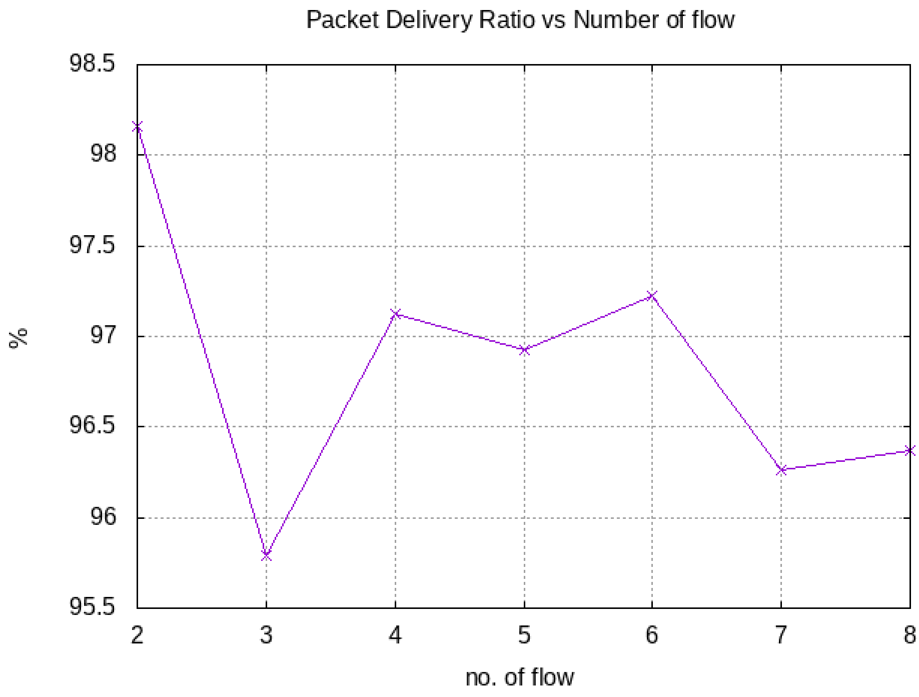

Packet Delivery Ratio vs of No. of Nodes
99
98
97
96
95
94
93
92
%
1
2
3
4
5
6
7
8
9
10
no. of nodes

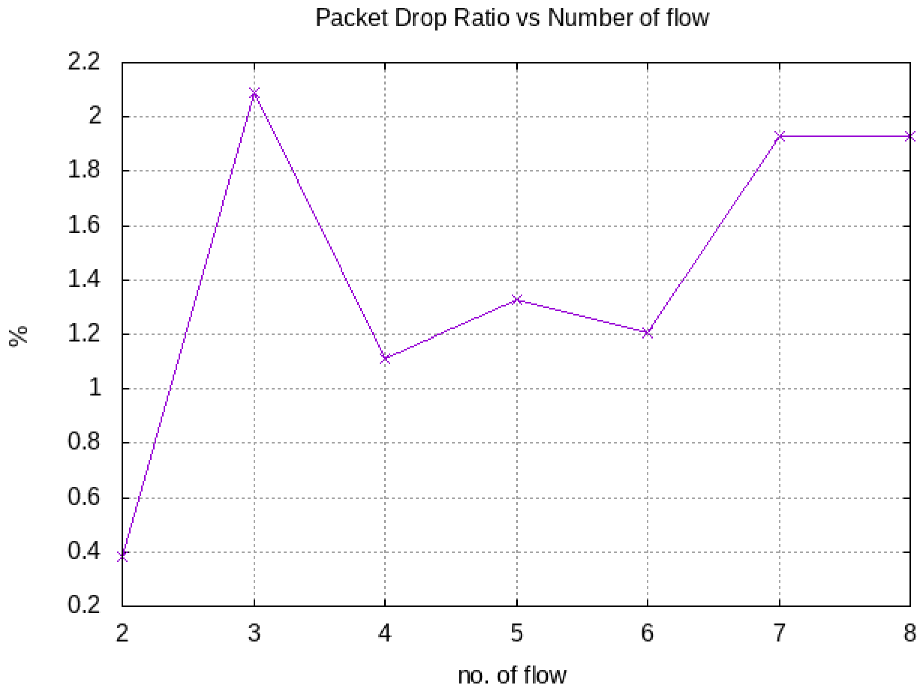

Packet Drop Ratio vs of No. of Nodes
7
6
5
4
3
2
1
0
%
1
2
3
4
5
6
7
8
9
10
no. of nodes

Varying number of flows:

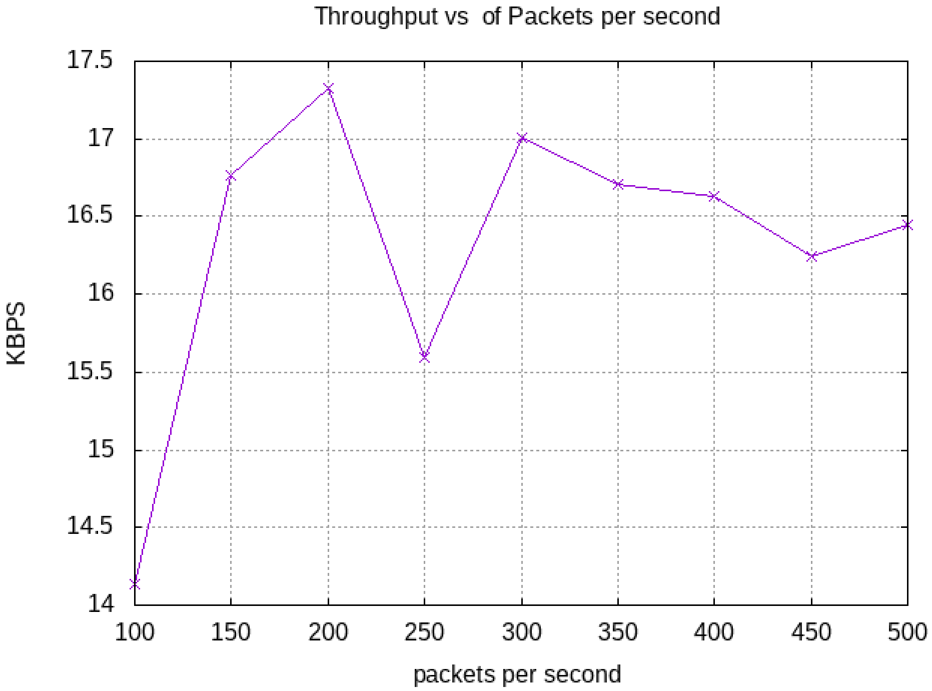


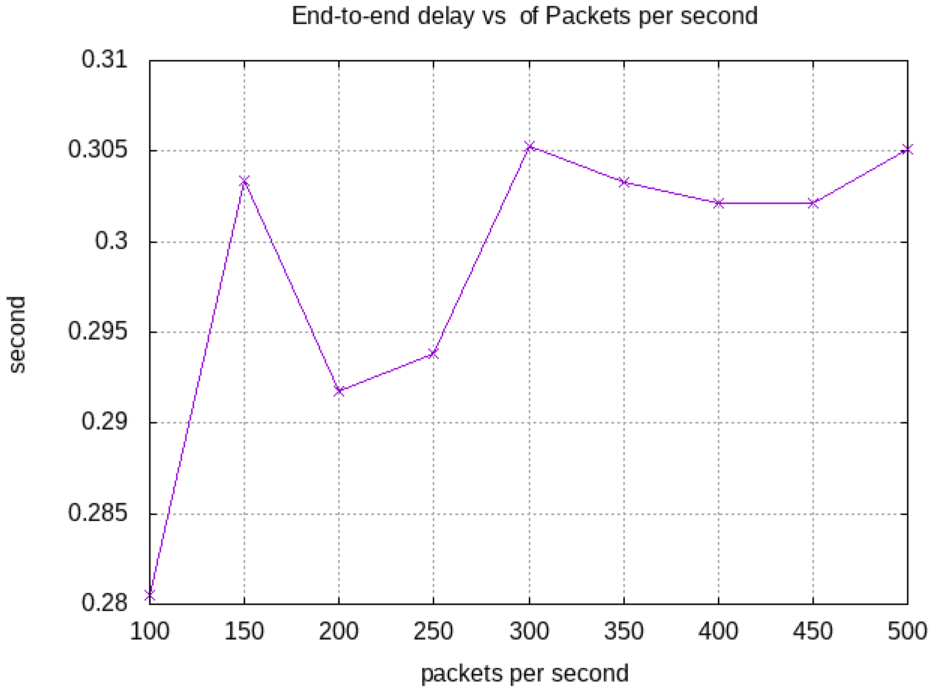


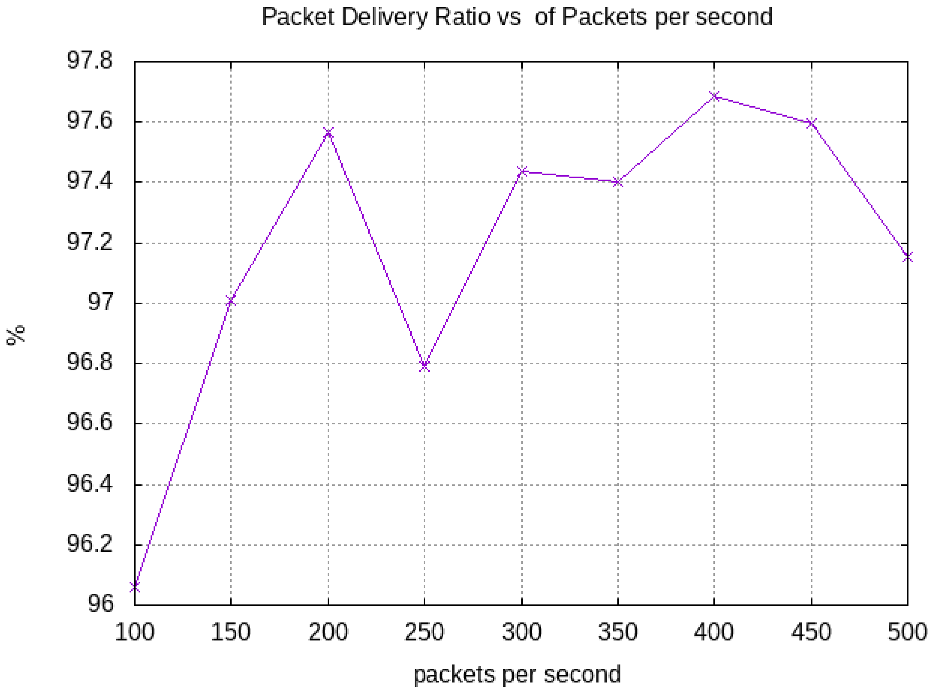

Packet Delivery Ratio vs Number of flow
98.5
98
97.5
97
96.5
96
95.5
%
2
3
4
5
6
7
8
no. of flow

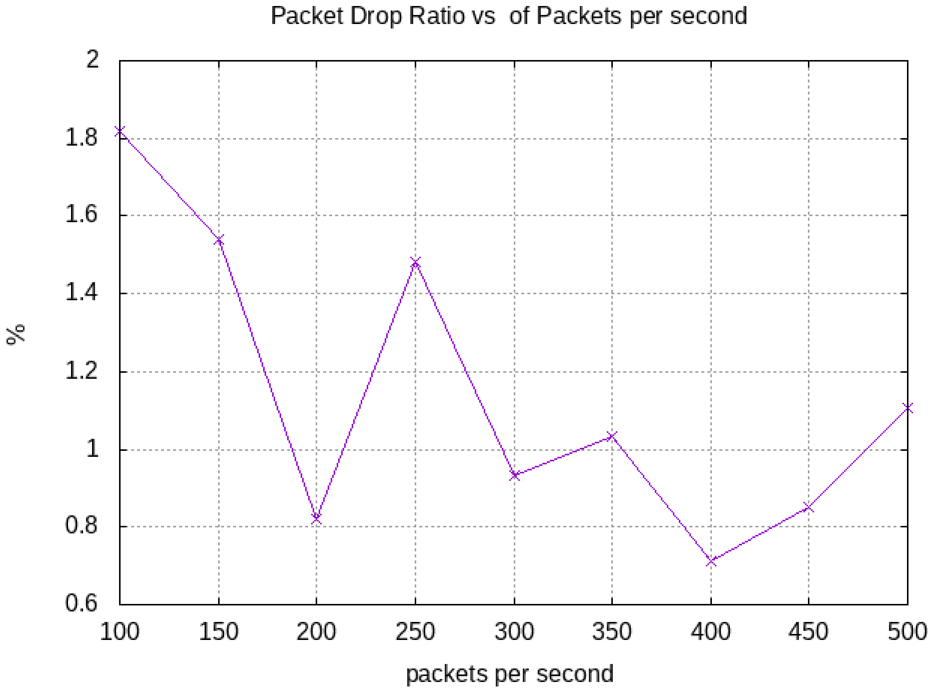

Packet Drop Ratio vs Number of flow
2.2
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
%
2
3
4
5
6
7
8
no. of flow

Varying packets per second:

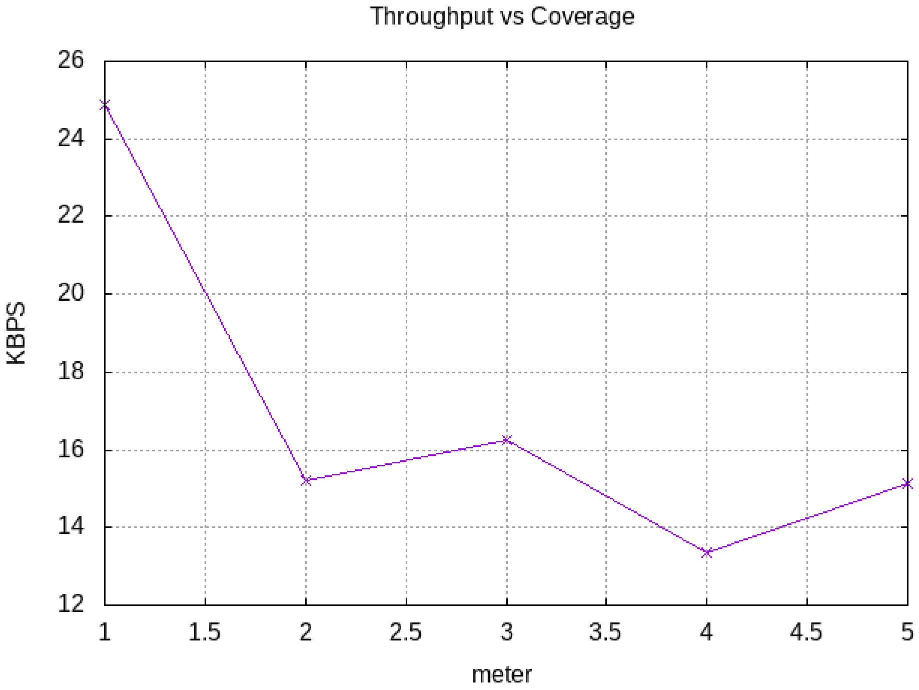


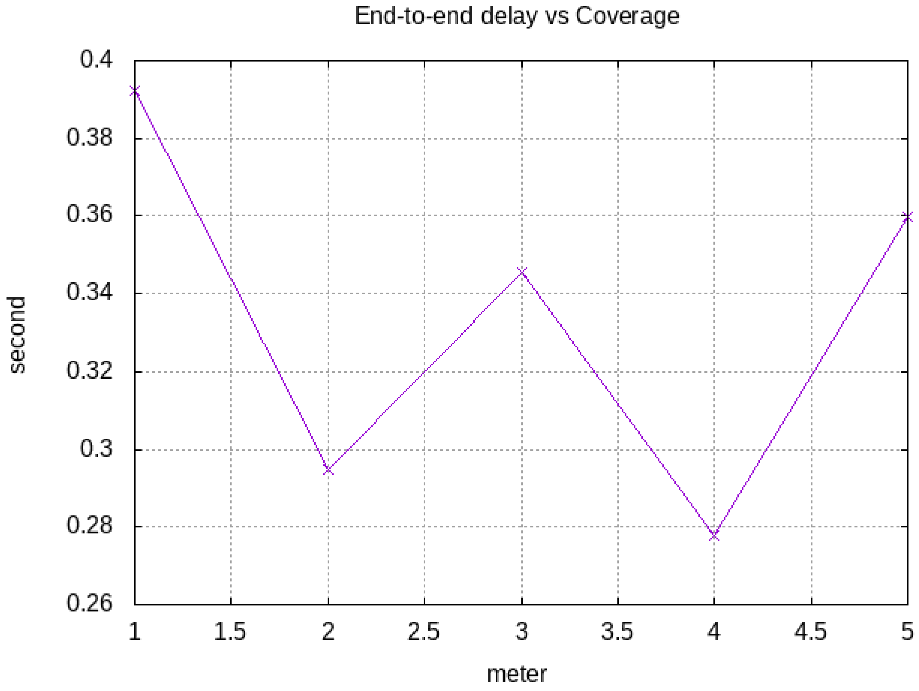


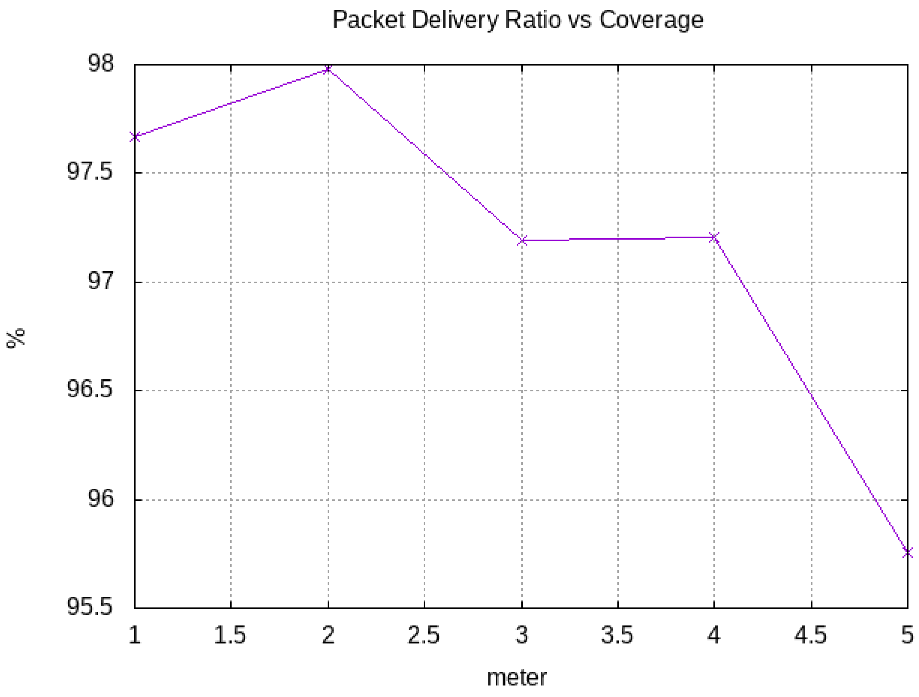

Packet Delivery Ratio vs of Packets per second
97.8
97.6
97.4
97.2
97
96.8
96.6
96.4
96.2
96
%
100
150
200
250
300
350
400
450
500
packets per second

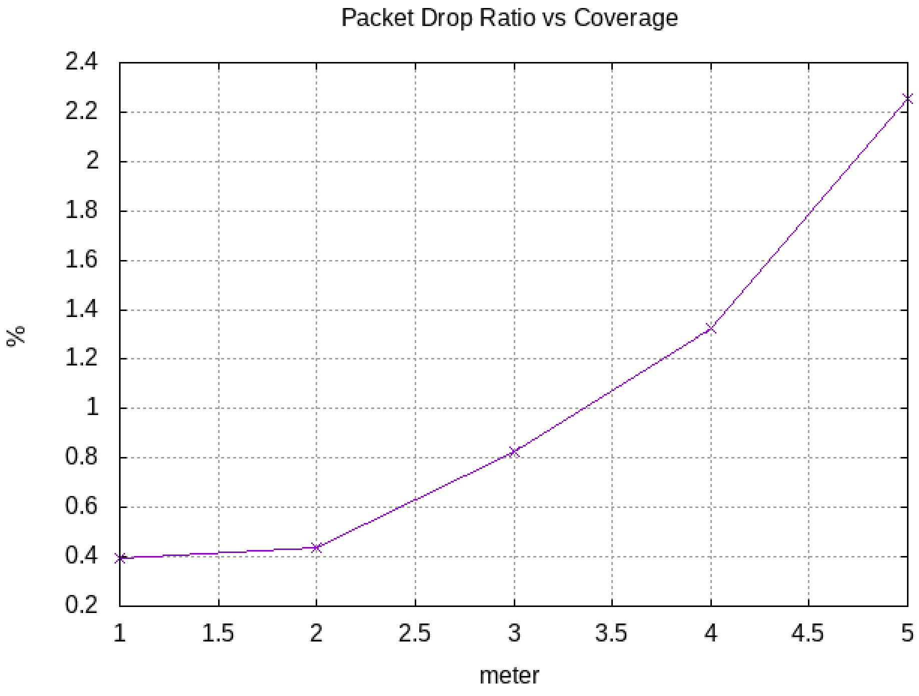

Packet Drop Ratio vs of Packets per second
2
1.8
1.6
1.4
1.2
1
0.8
0.6
%
100
150
200
250
300
350
400
450
500
packets per second

Varying coverage:



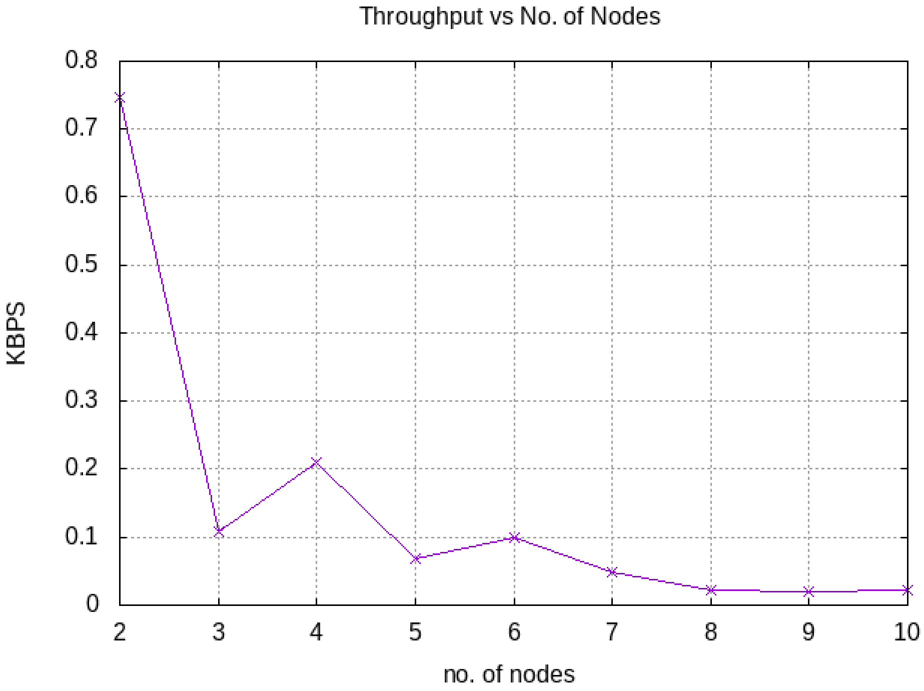


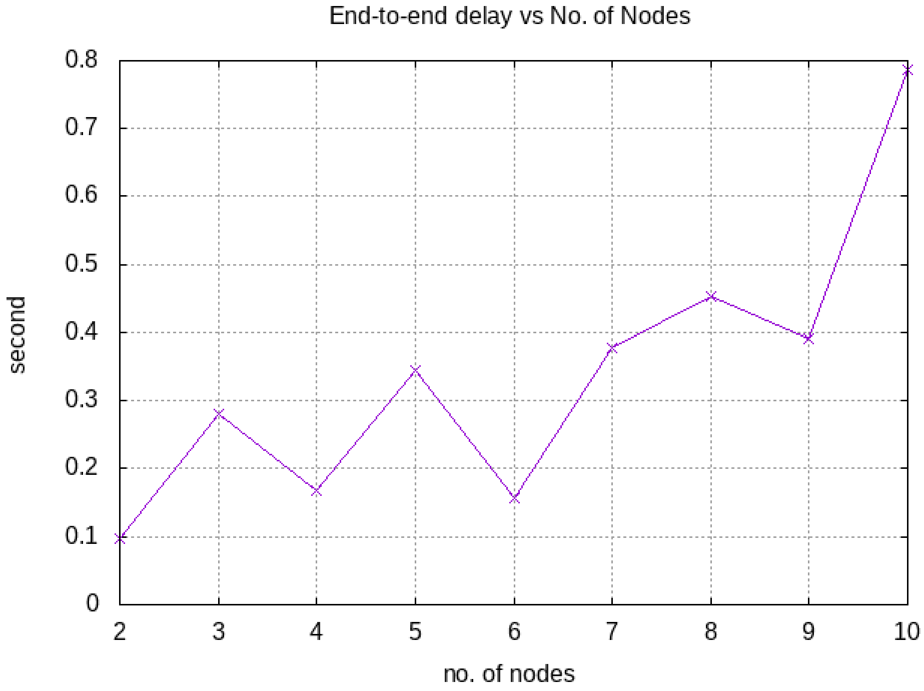

Packet Delivery Ratio vs Coverage
98
97.5
97
96.5
96
95.5
%
1
1.5
2
2.5
3
3.5
4
4.5
5
meter


Packet Drop Ratio vs Coverage
2.4
2.2
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
%
1
1.5
2
2.5
3
3.5
4
4.5
5
meter

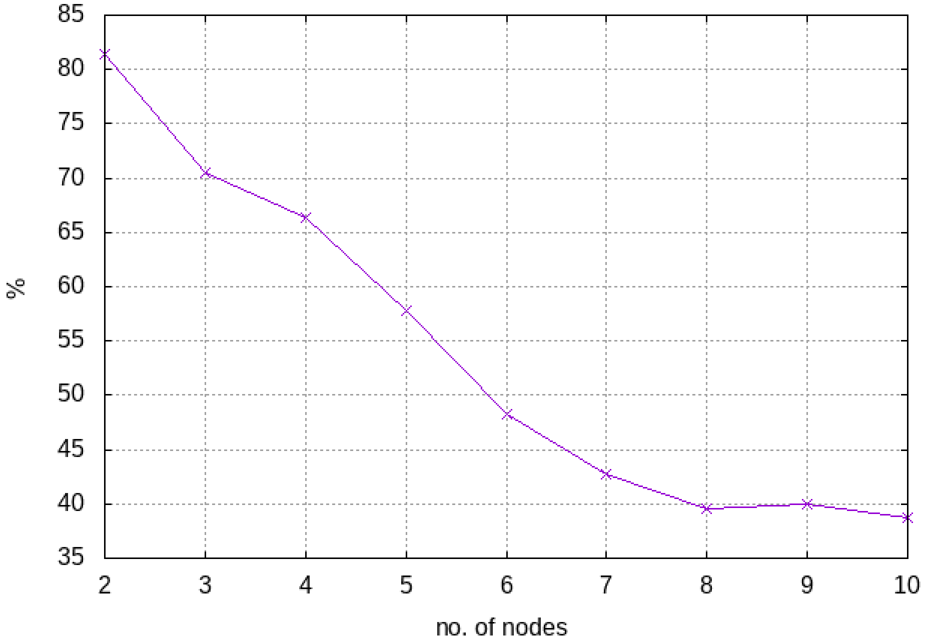
Graphs (Task A - Wireless low-rate)

Varying number of nodes:

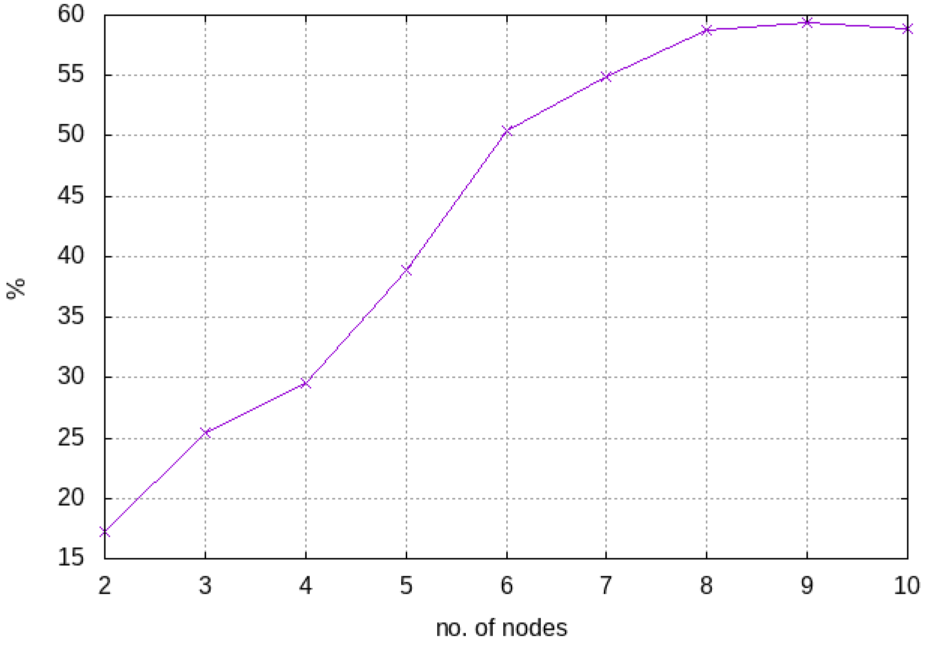




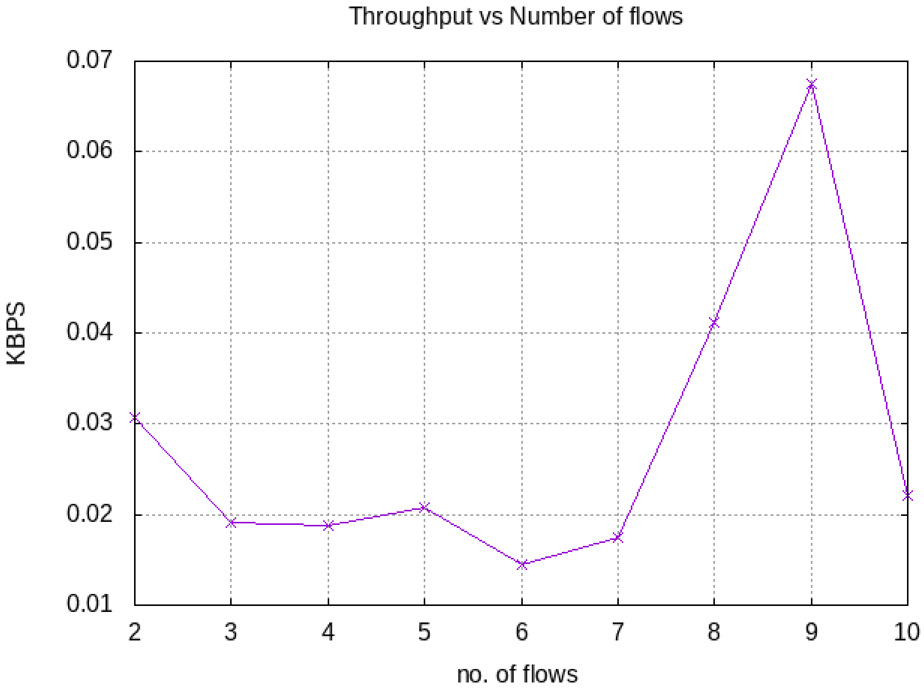
Packet Delivery Ratio vs No. of Nodes

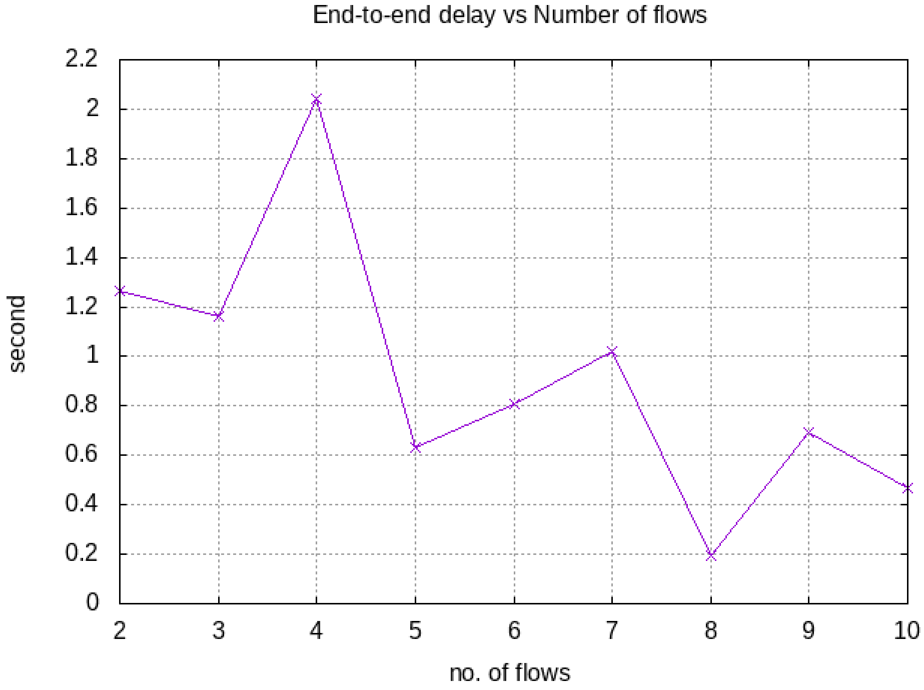


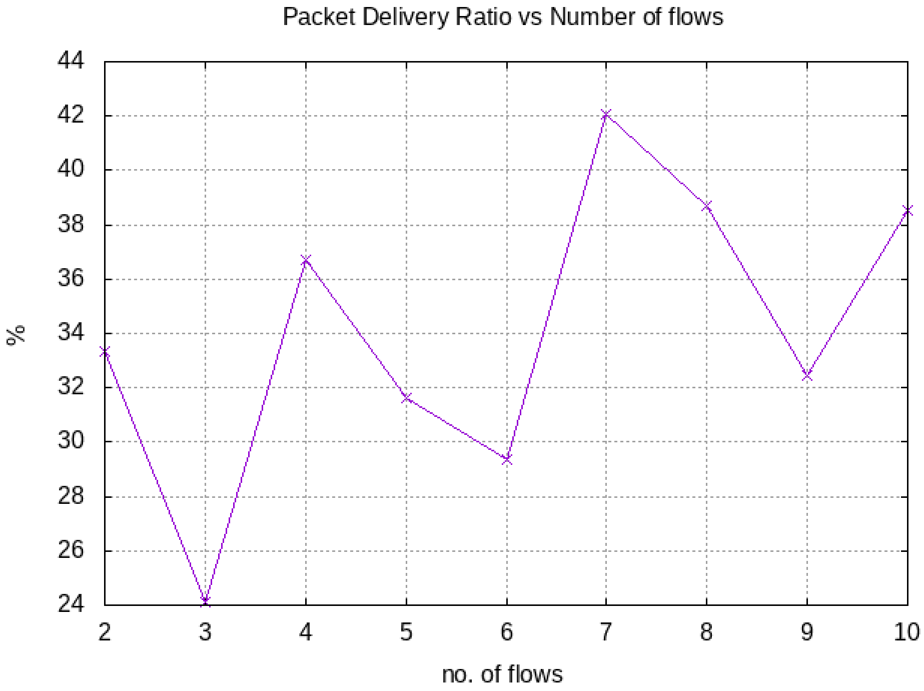
Packet Drop Ratio vs No. of Nodes

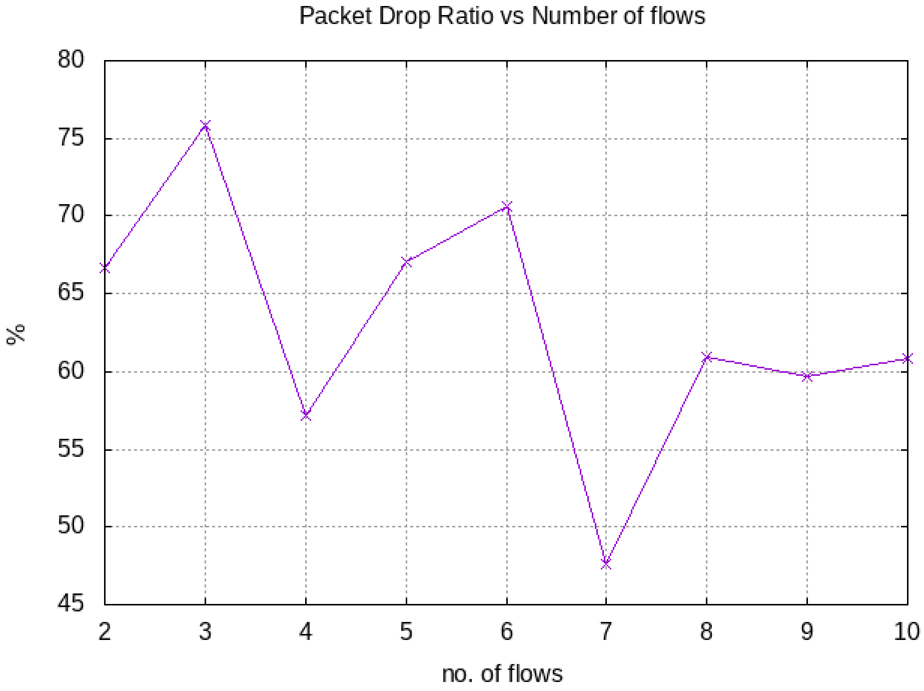


Varying number of flows:

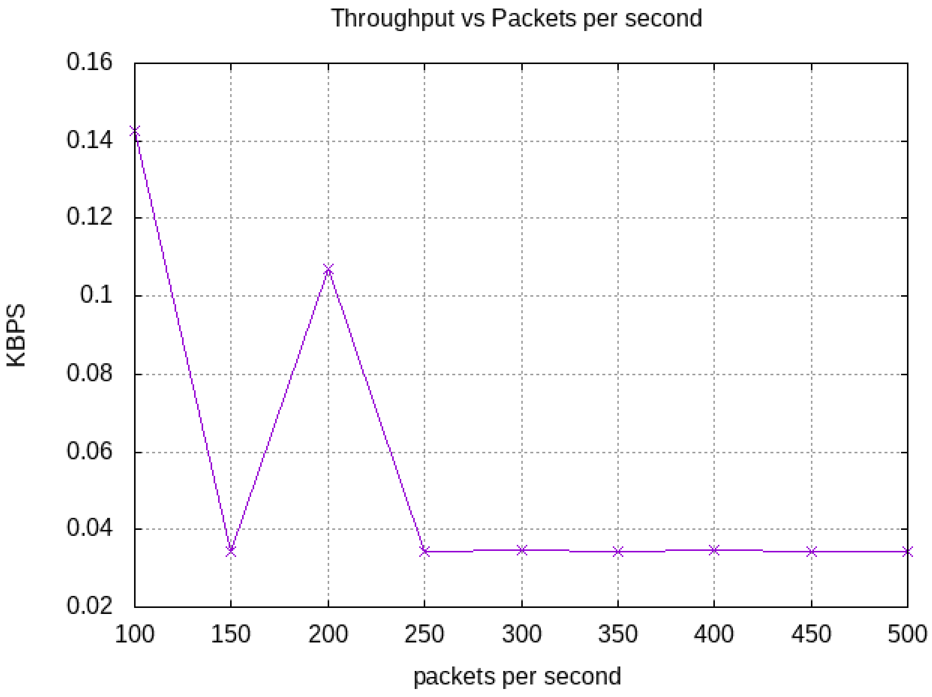


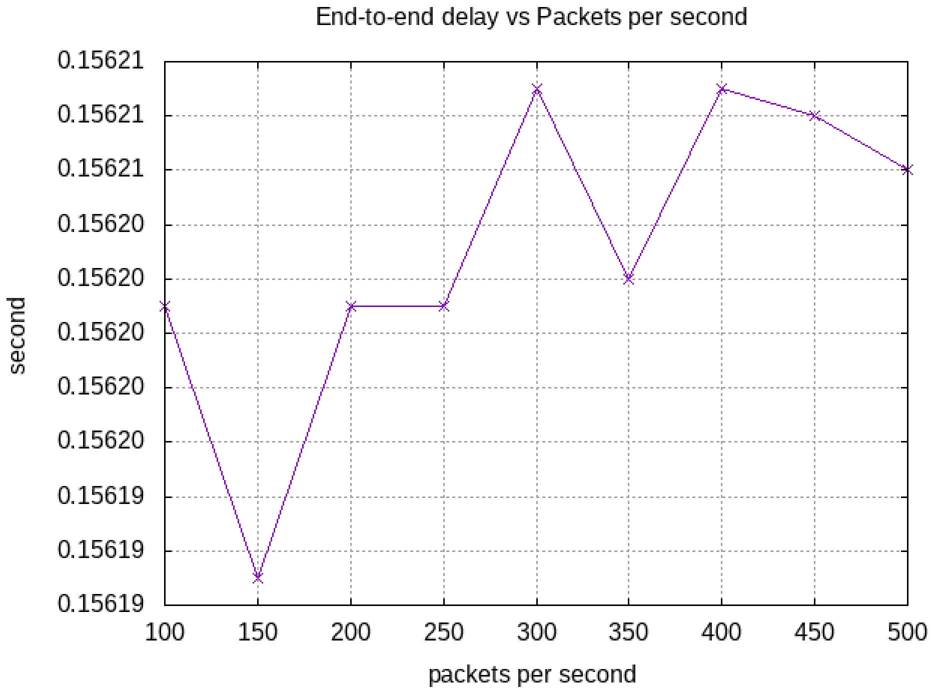


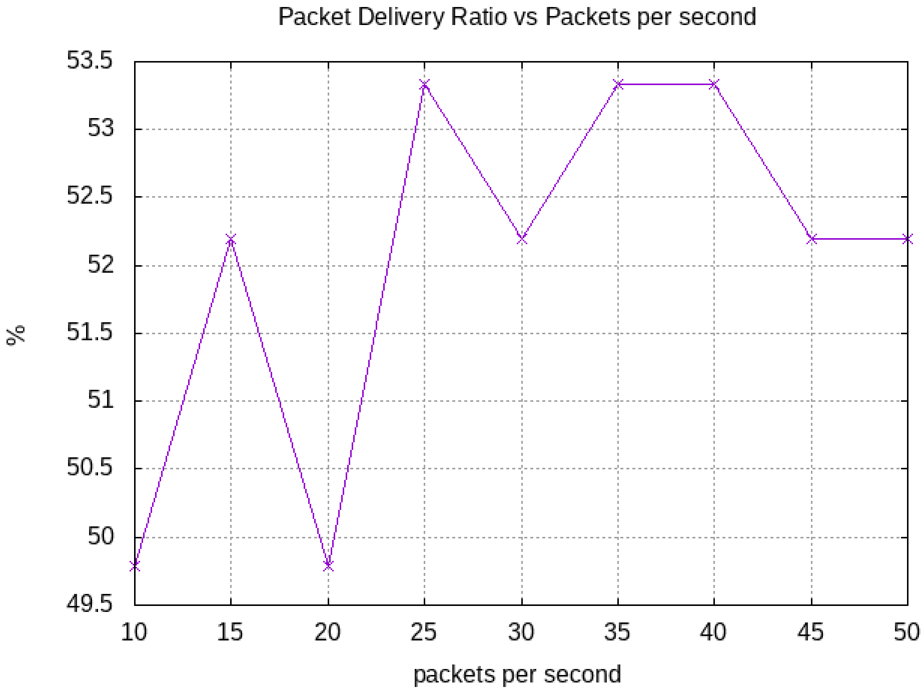

Packet Delivery Ratio vs Number of flows
44
42
40
38
36
34
32
30
28
26
24
%
2
3
4
5
6
7
8
9
10
no. of flows

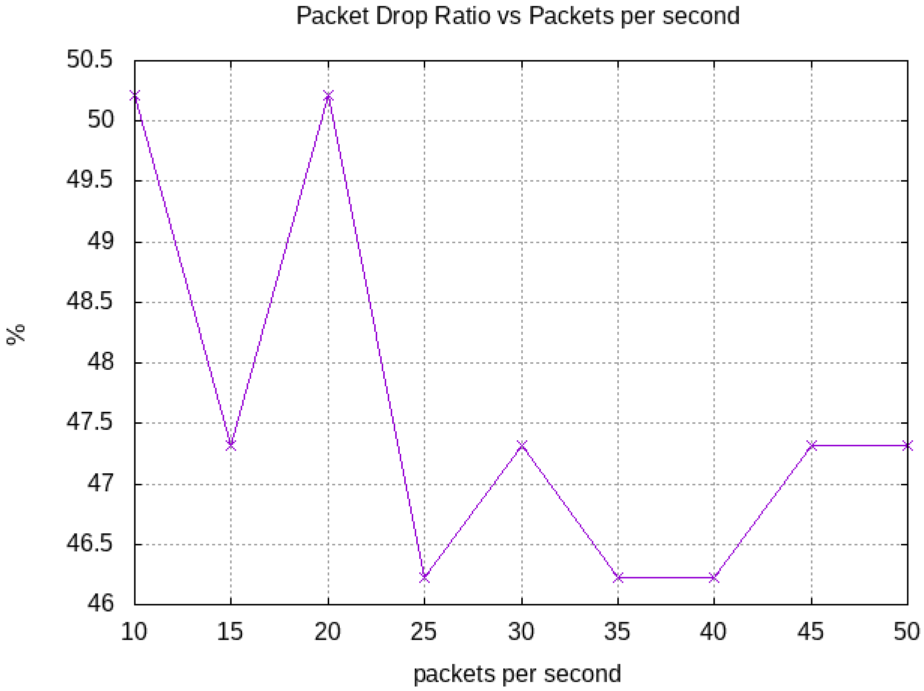

Packet Drop Ratio vs Number of flows
80
75
70
65
60
55
50
45
%
2
3
4
5
6
7
8
9
10
no. of flows

Varying packets per second:

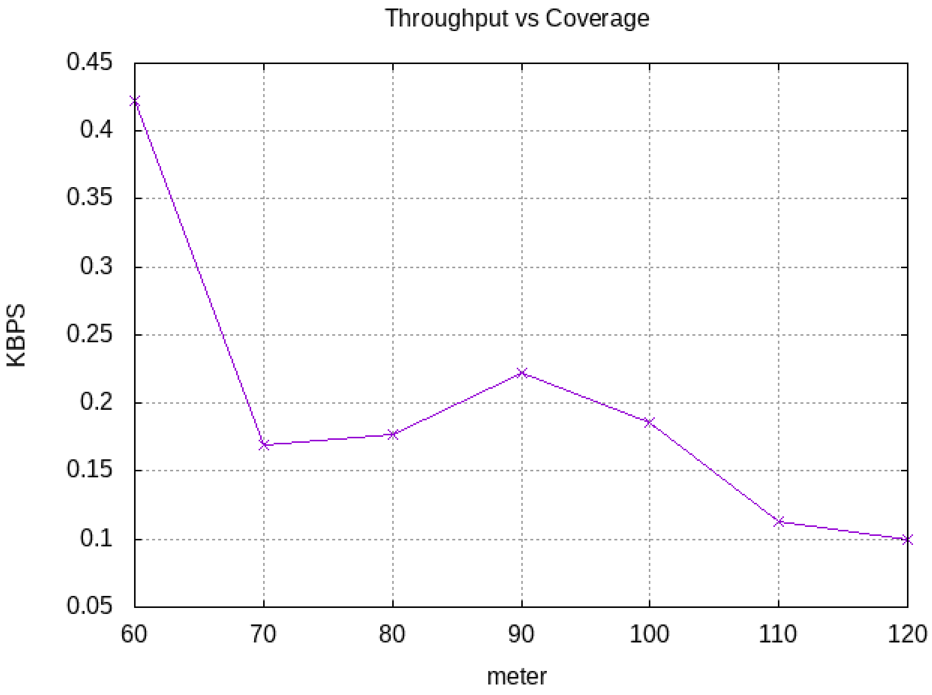


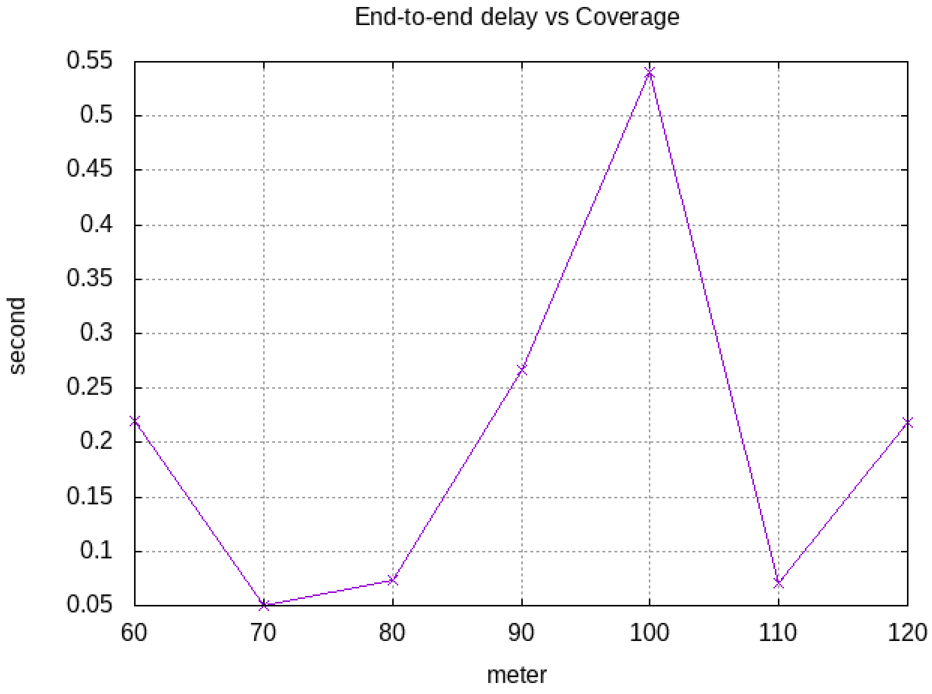


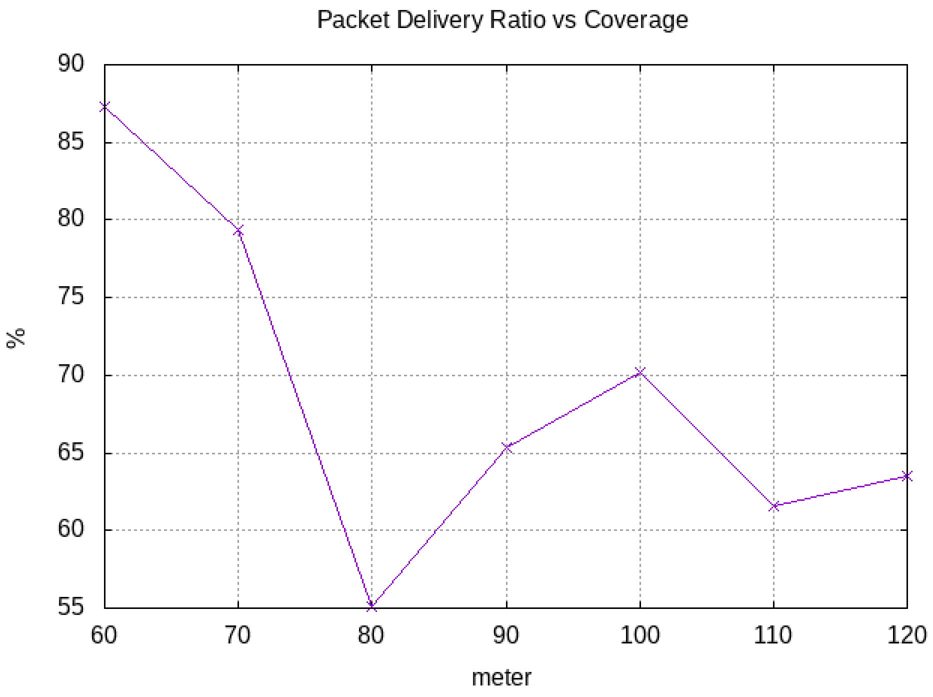

Packet Delivery Ratio vs Packets per second
53.5
53
52.5
52
51.5
51
50.5
50
49.5
%
10
15
20
25
30
35
40
45
50
packets per second

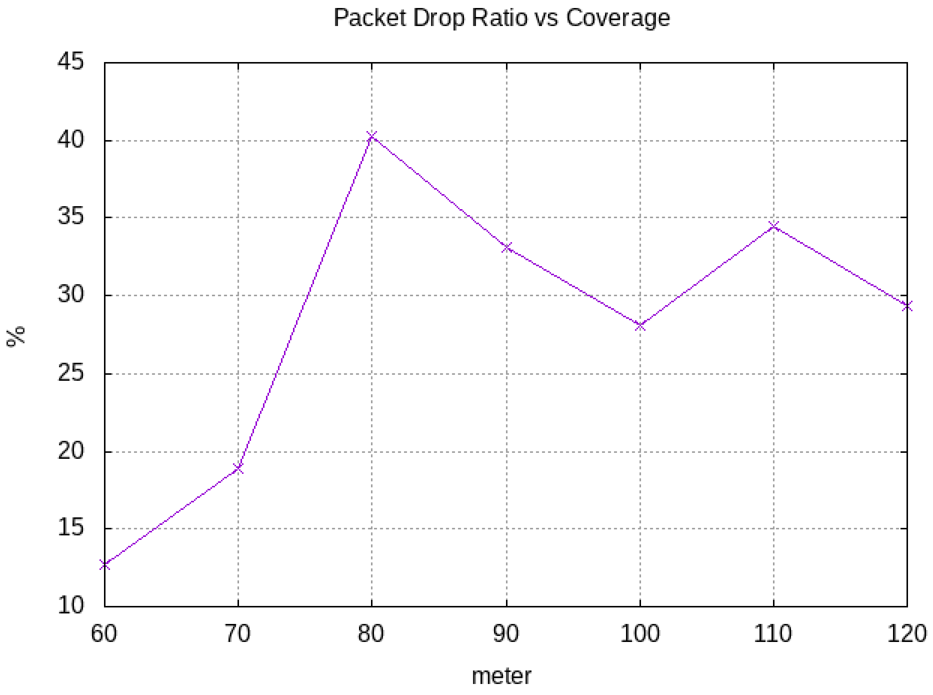

Packet Drop Ratio vs Packets per second
50.5
50
49.5
49
48.5
48
47.5
47
46.5
46
%
10
15
20
25
30
35
40
45
50
packets per second

Varying coverage:



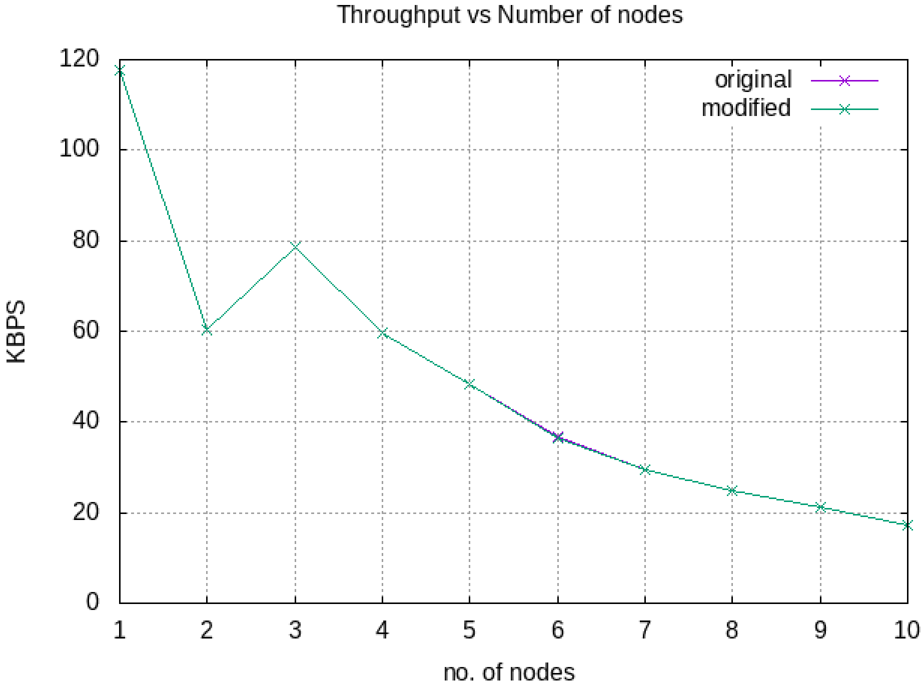


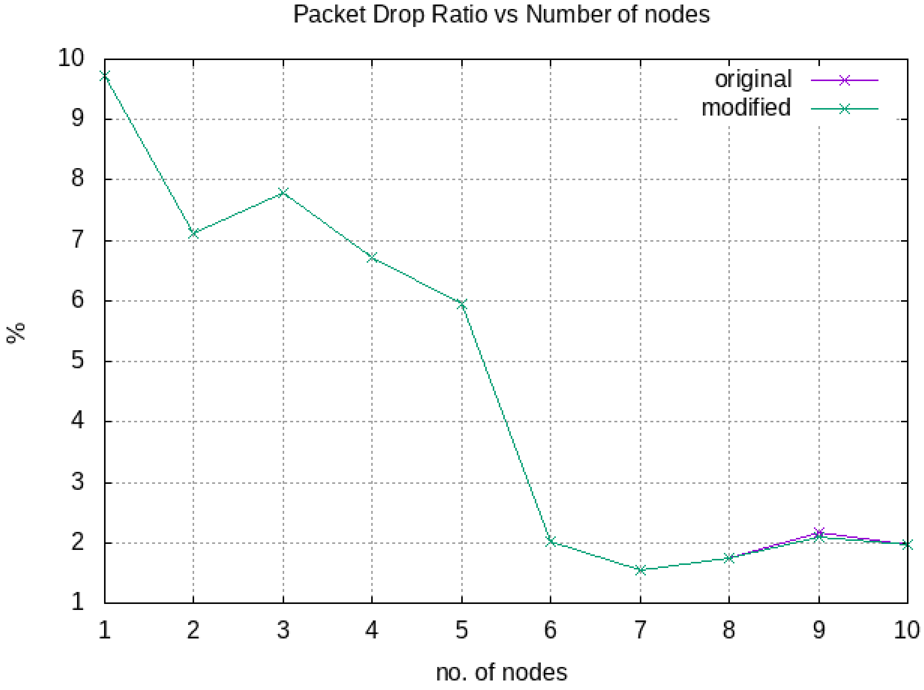

Packet Delivery Ratio vs Coverage
90
85
80
75
70
65
60
55
%
60
70
80
90
100
110
120
meter


Packet Drop Ratio vs Coverage
45
40
35
30
25
20
15
10
%
60
70
80
90
100
110
120
meter

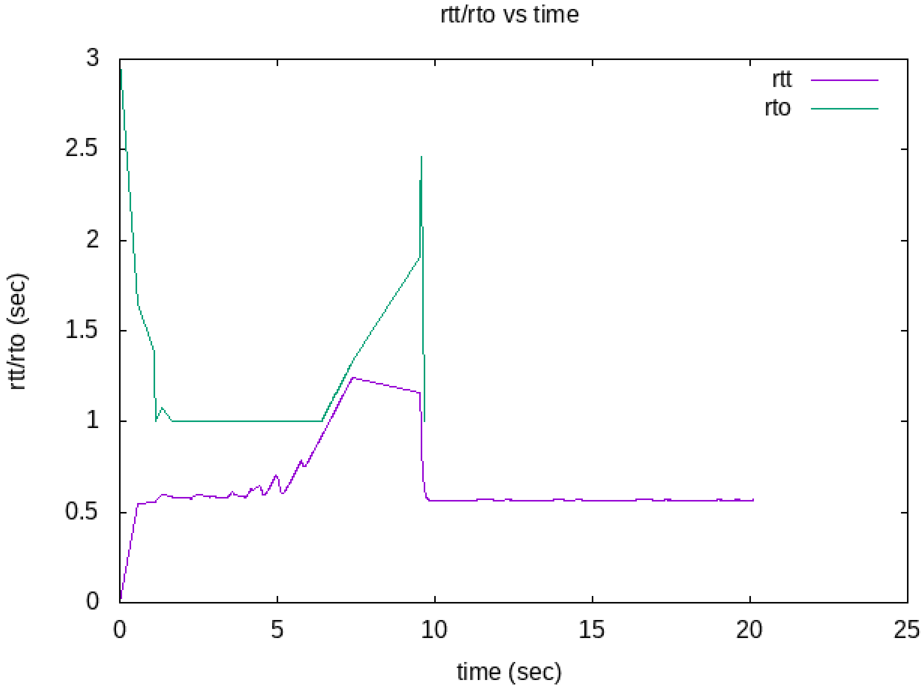
Graphs (Task B)

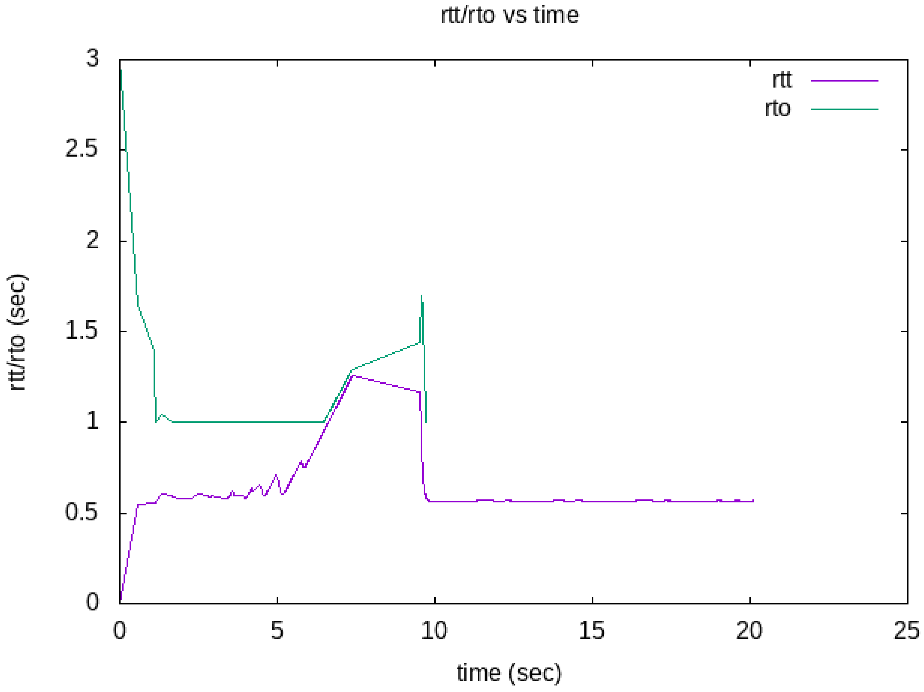
Varying number of nodes:





RTT/RTO (first one original, second one modified):





Summary of Findings

Task A:

- Throughput decreases when the number of nodes or flows increases because of increasing congestion. On the other hand, it increases when packets per second increases.
- End-to-end delay increases when the number of nodes or flows or packets per second increases.
- Packet delivery ratio decreases when number of nodes or flows increases.

Task B:

- Throughput and drop ratio stayed pretty similar. But sudden increase of RTO improved almost as expected as in the paper.