A Report on
<Network Simulator 2>

1505091 - Rituparna Datta
1505095 - Jamalia Sultana Jisha

Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology
(BUET)
Dhaka 1000
January 13, 2019

Contents

1 Introduction

NS2, short for Network Simulator 2, is a popular open source network simulator for carrying out network experimentation that runs on linux.

This report details the simulation for wireless 802.11 (mobile) and wireless 802.15.4 (static) network for both existing protocols and slightly modified protocols. For each simulation, some parameters were varied and various outputs were observed.As bonus work, Wimax 802.16 network was simulated, simulation by varying queue and per node throughput were generated for the other two networks. A cross transmission between wired and wireless 802.11 (static) is also simulated.

2 Given Topologies

1. Wireless 802.11 (mobile)
2. Wireless 802.15.04 (static)

3 Parameters varied for plotting grapgh

1. Number of nodes : 20 40 60 80 100
2. Number of flows : 10 20 30 40 50
3. Number of packets per second : 100 200 300 400 500
4. Speed of the nodes :(m/s) 5 10 15 20 25 - (for wireless mobile nodes only)
5. Coverage area :TX Range, 2 X TX Range, 3 X TX Range, 4X TX Range, 5X TX Range - (for wireless static nodes only)

4 Modification made in the simulator

- Congestion control in *tcp.cc*
  - A new case in the switch-case block was added for congestion control. For this value of $open_wnd$ variable is set in $ns-default.tcl$ in tcl/lib folder.
    The sender TCP updates its congestion window size in the congestion avoidance phase according to the following equation when it receives an ACK packet from receiver TCP as , $cwnd = cwnd + (f/cwnd)$
    The algorithm was used here, is sort of * Slow-Start Phase (Exponential Increment) */
    if (cwnd_ ¡ ssthresh)
    $cwnd\_ = cwnd\_ + 1;$
    $/ * CongestionAvoidancePhase * /$

$elsef = wnd\_const\_ * ssthresh\_/cwnd\_ * pow(cwnd\_, k\_parameter\_); cwnd\_ = cwnd\_ + (f/cwnd\_);$

  - Since we know in slow start, *cwnd* needs to be increased exponentially, we tried to see it's effect on throughput by increasing it on different manner. For this, added new variable $slow_start_cwnd$ and this was initialized in *tcl/lib*

- In **AODV** routing protocol,
  - ttl_start value was decreased. Also small value of *ttl* was set for local route repair process.
  - After taking **local repair process** , if the link still broken, *RERR* was send to the sender to inform.
  - $ttl_threshold$ was increased.

- RTT calculation in *tcp.cc*
  - **Jacobson/Karels** Algorithm was implemented in RTT calculation, because using RTT for timeout doesn't work and at high loads round trip variance is high. For this , if mean variance is denoted by D, timeout = RTT + $4D$

- Fast recovery was introduced in TCP.For this fast retransmit in tcp is improved.

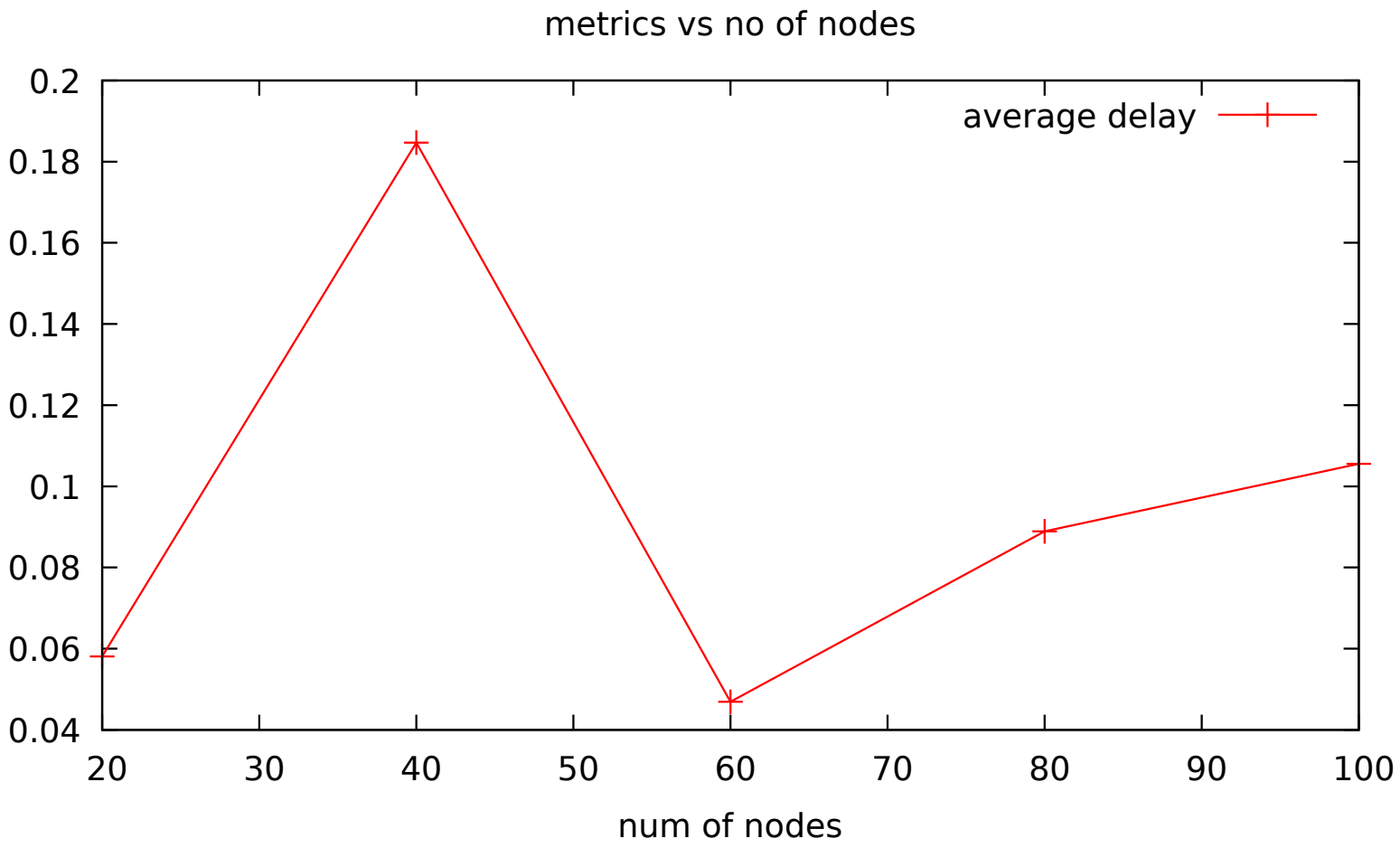
5 Output

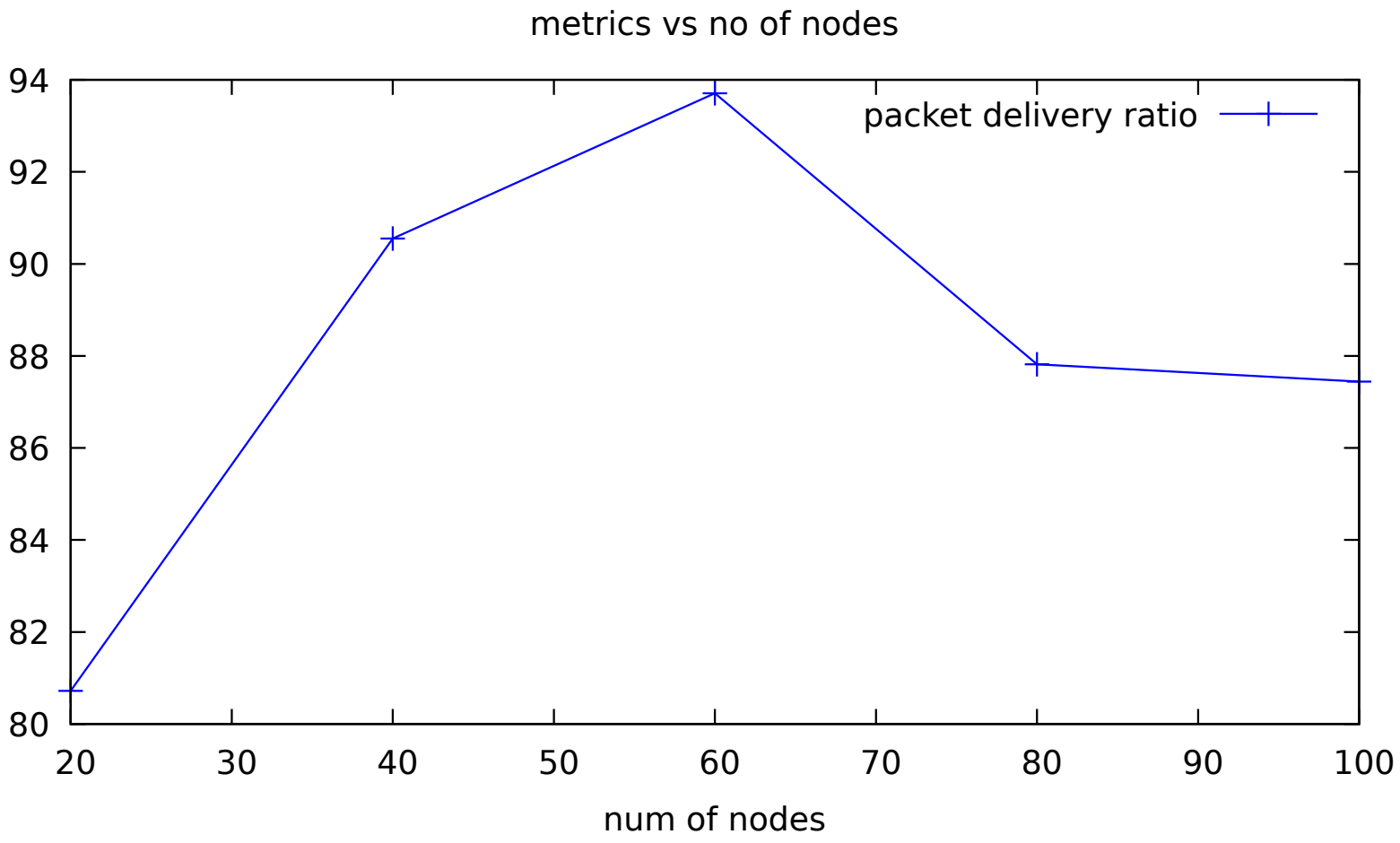
Outputs were observed for :

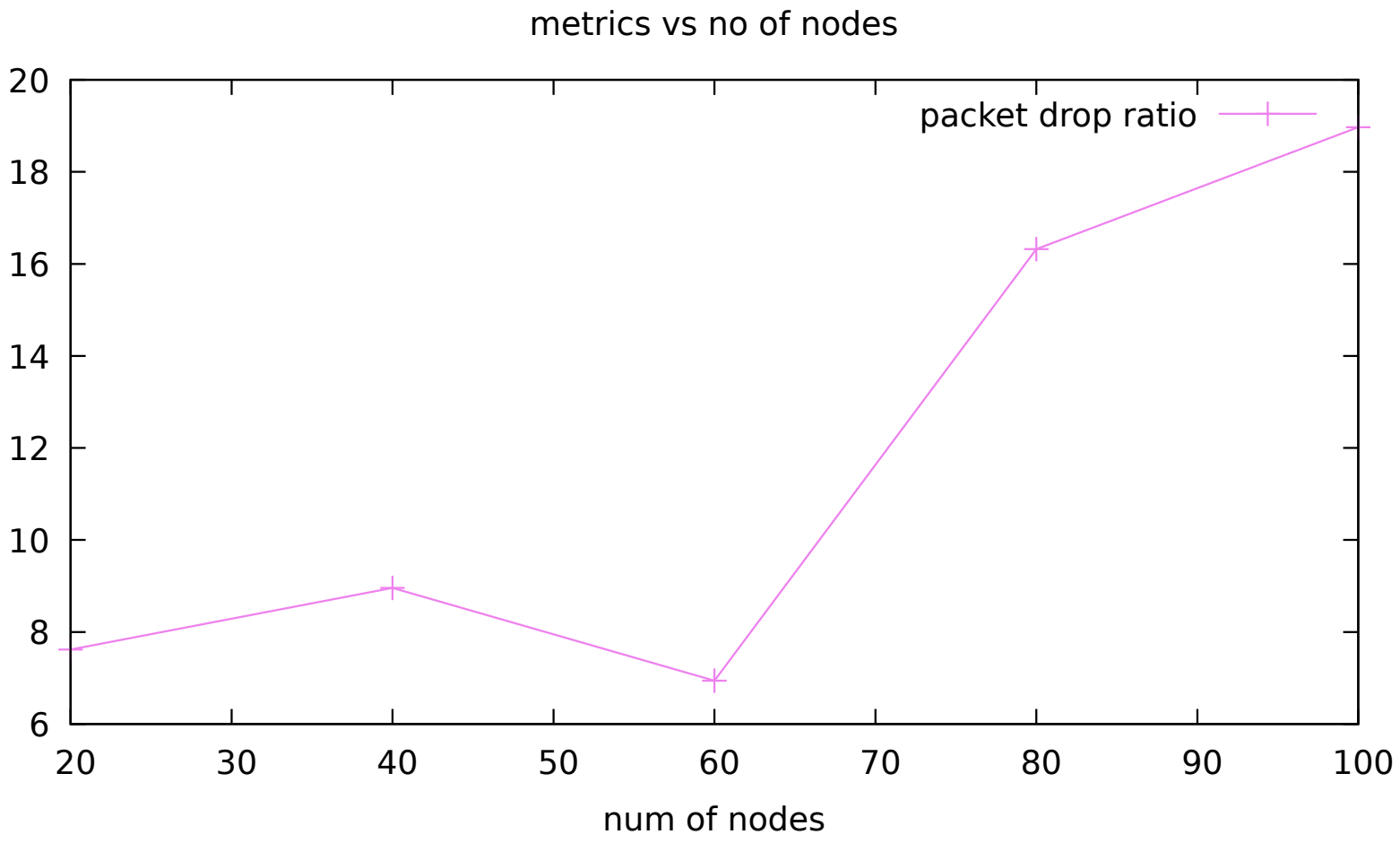
- Network throughput
- End to end delay
- Delivery ratio
- Drop ratio
- Energy Consumption
- Per node throughput

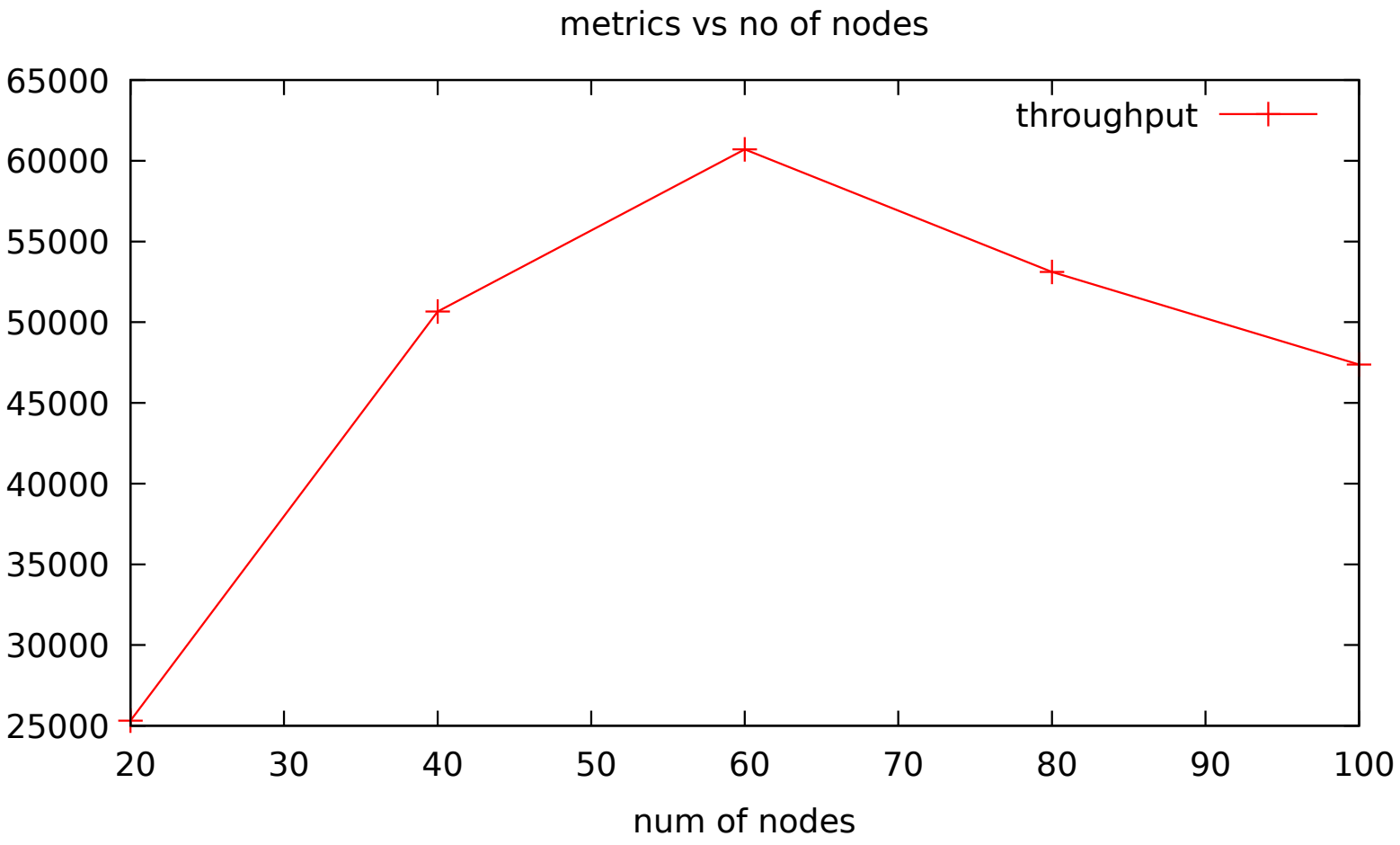
6 Graph

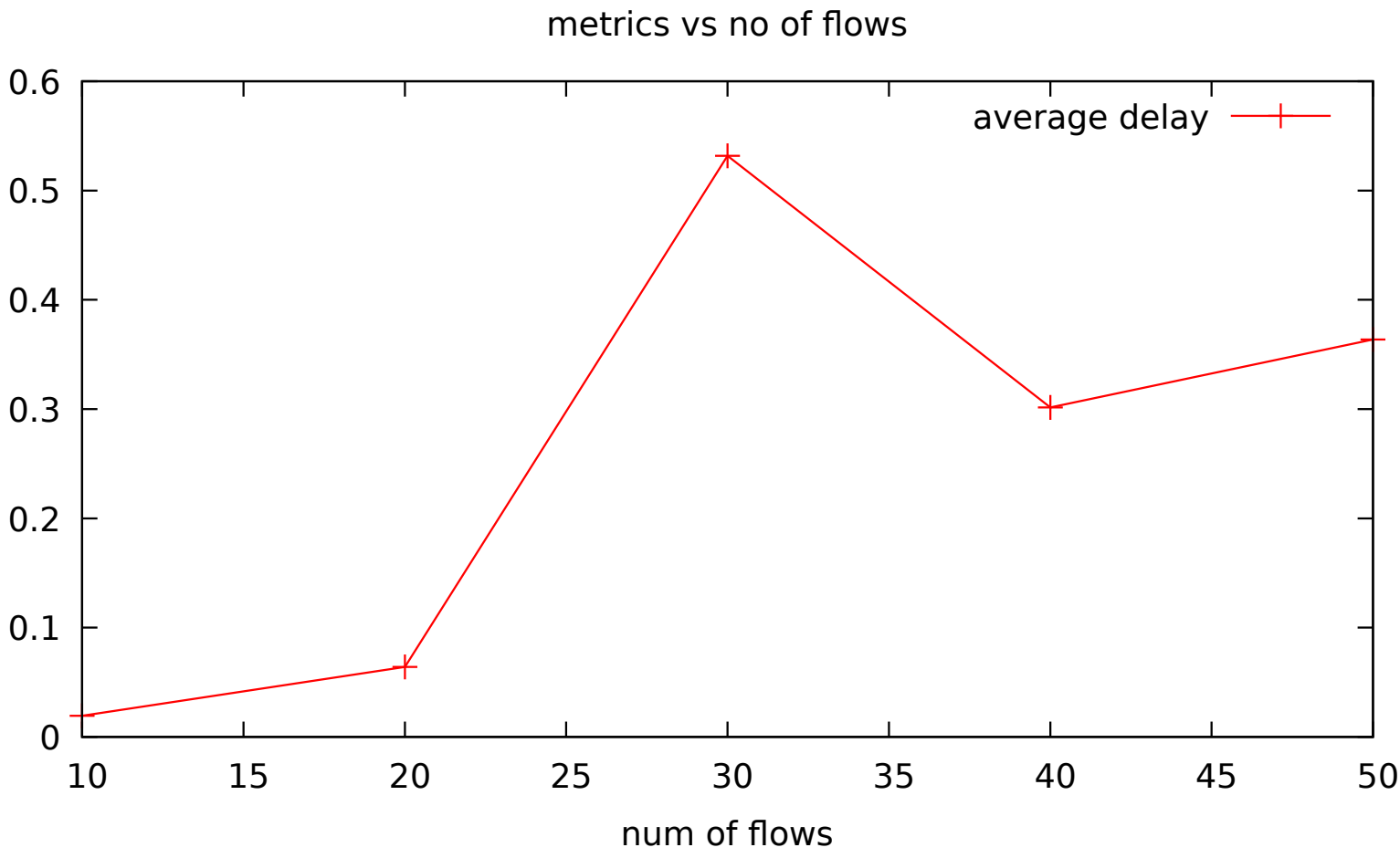
6.1 802.11 Mobile Original

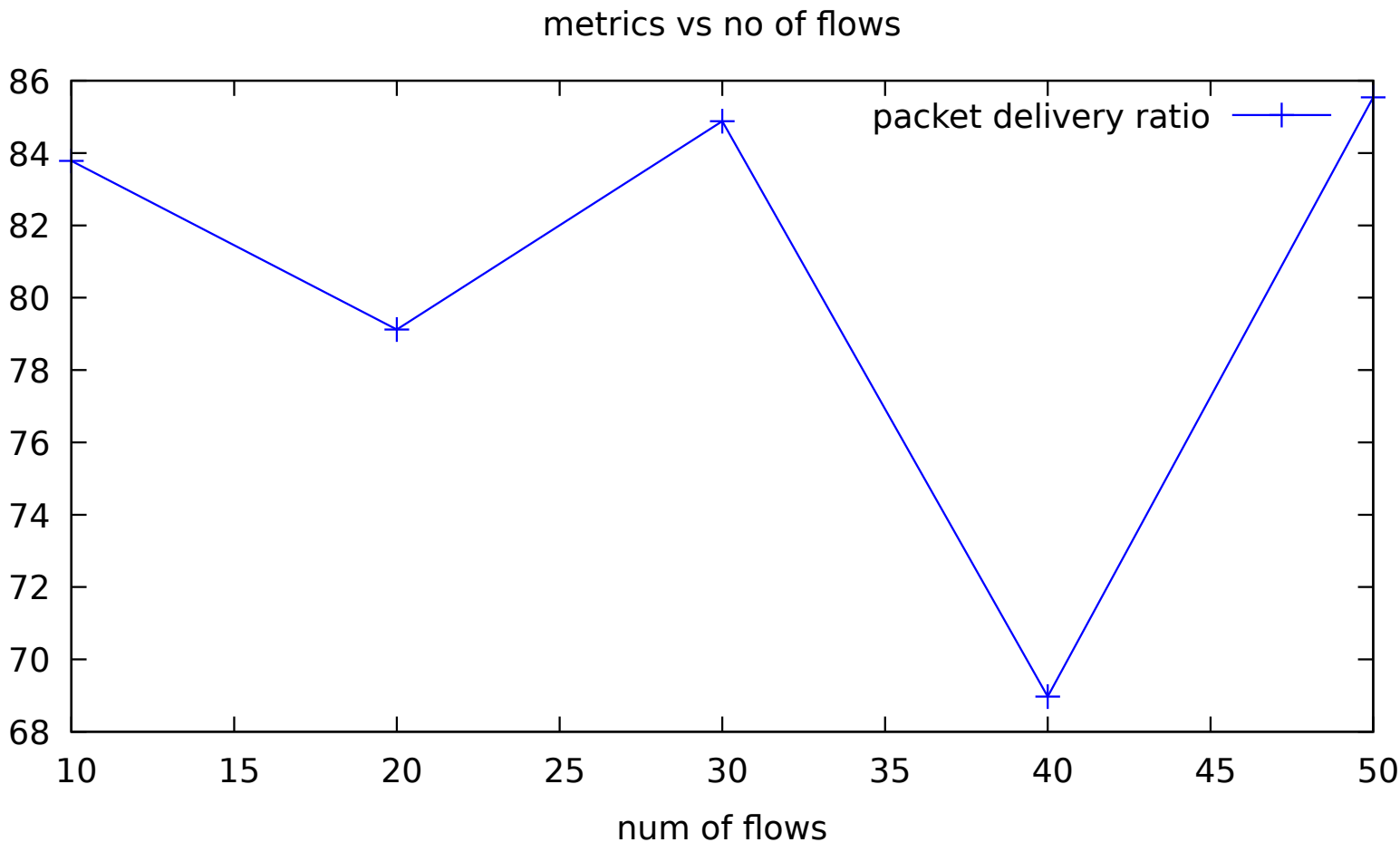

metrics vs no of nodes
average delay
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
20
30
40
50
60
70
80
90
100
num of nodes

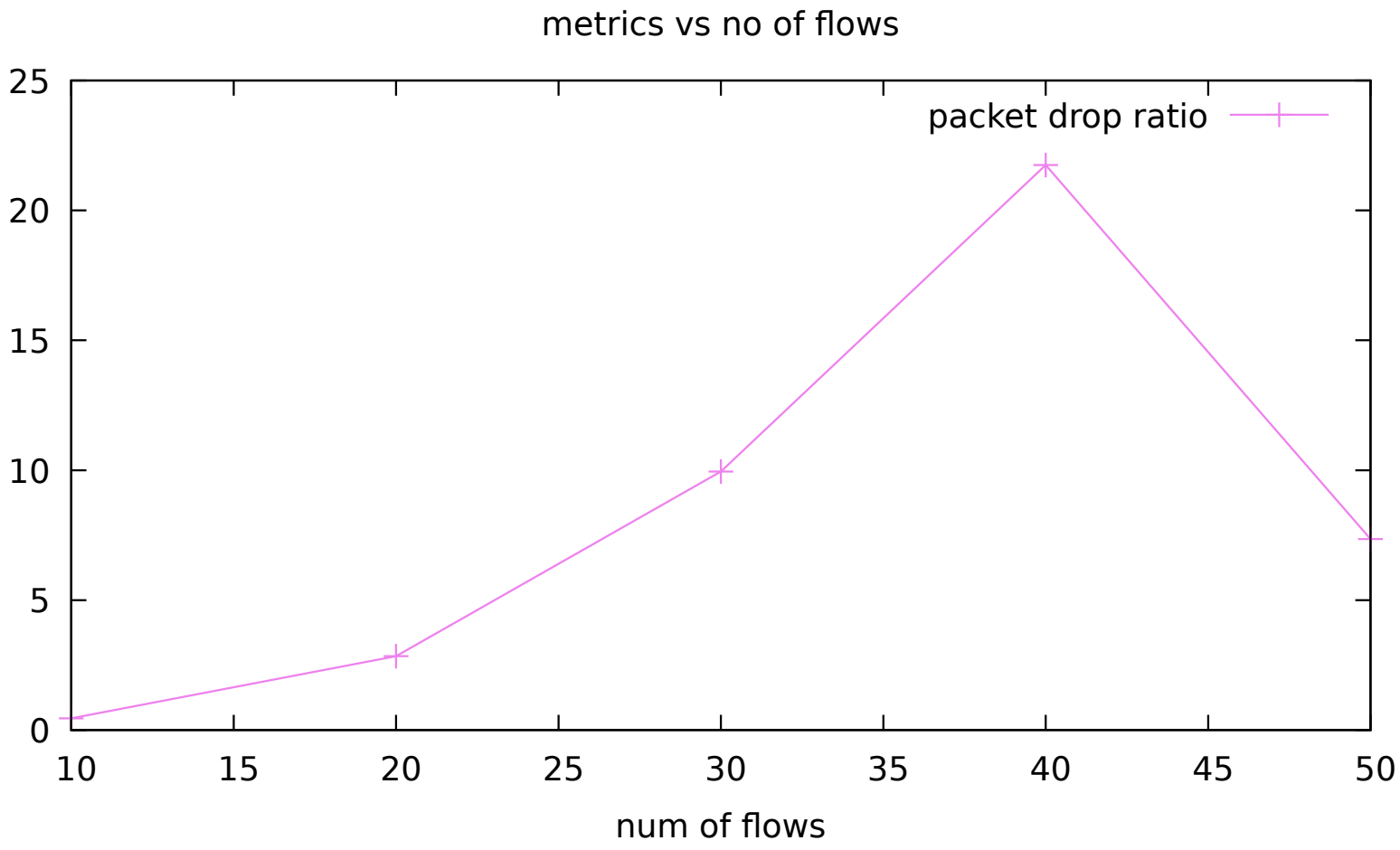

metrics vs no of nodes
packet delivery ratio
94
92
90
88
86
84
82
80
20
30
40
50
60
70
80
90
100
num of nodes

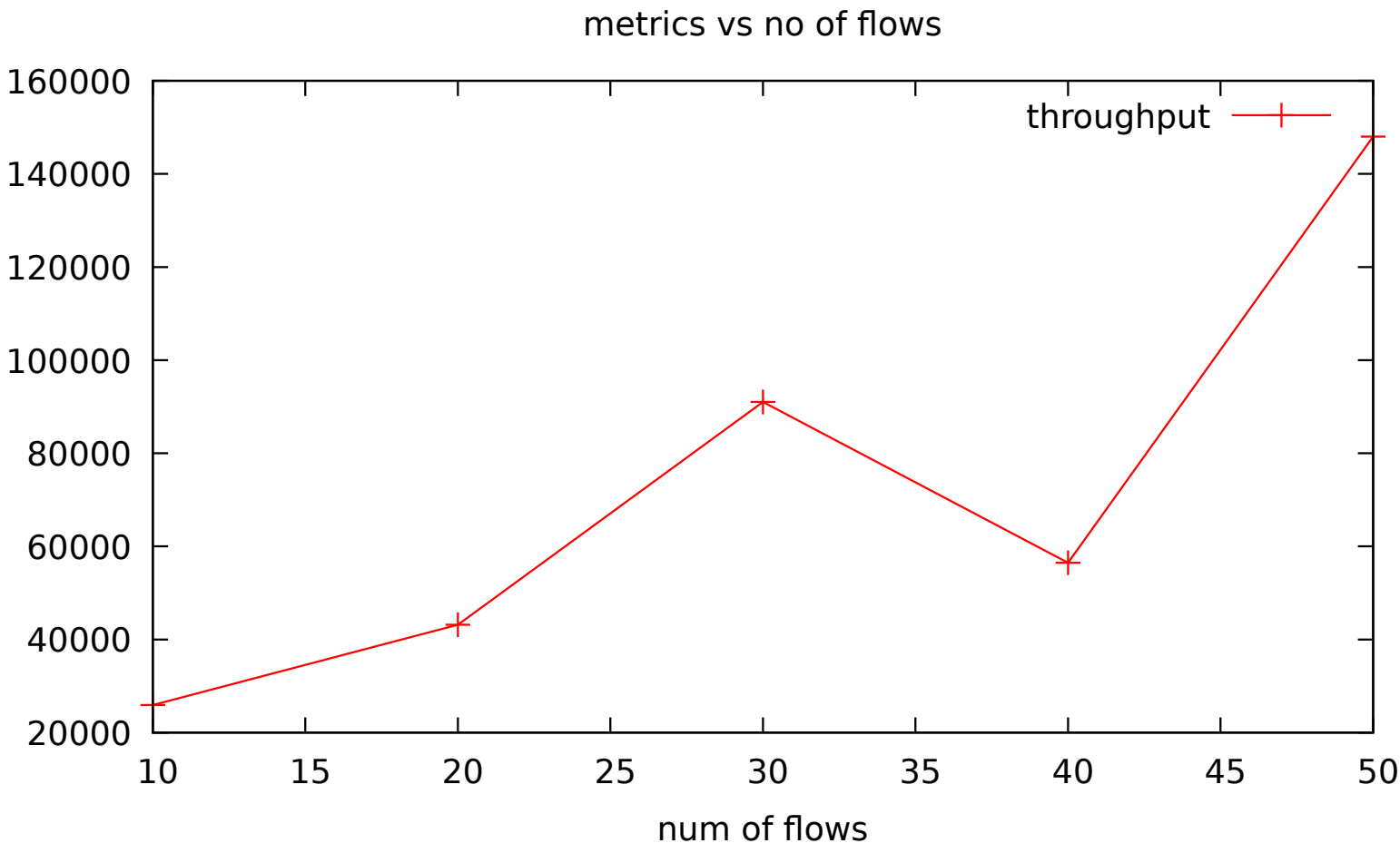

metrics vs no of nodes
packet drop ratio
20
18
16
14
12
10
8
6
20
30
40
50
60
70
80
90
100
num of nodes

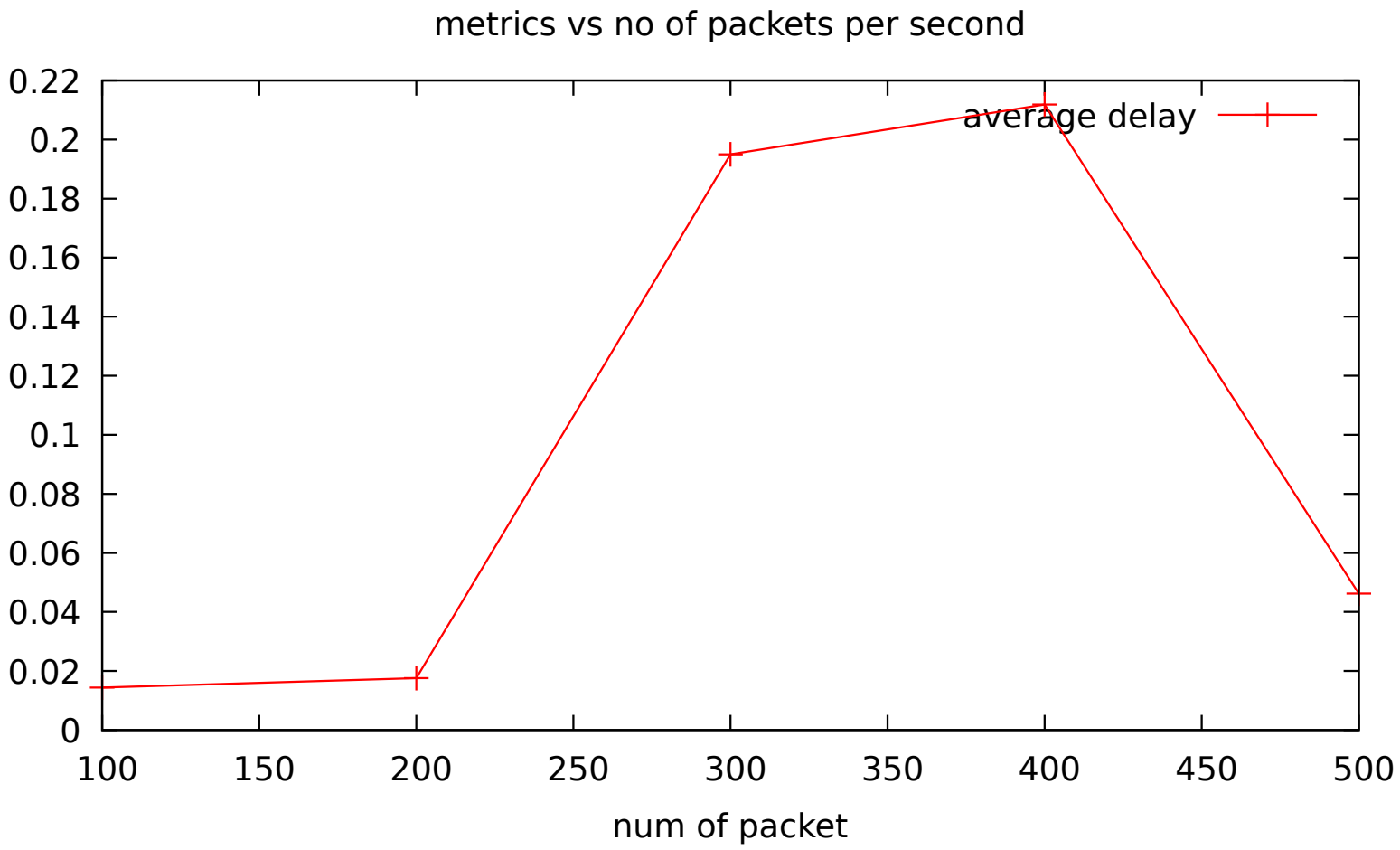

metrics vs no of nodes
throughput
65000
60000
55000
50000
45000
40000
35000
30000
25000
20
30
40
50
60
70
80
90
100
num of nodes

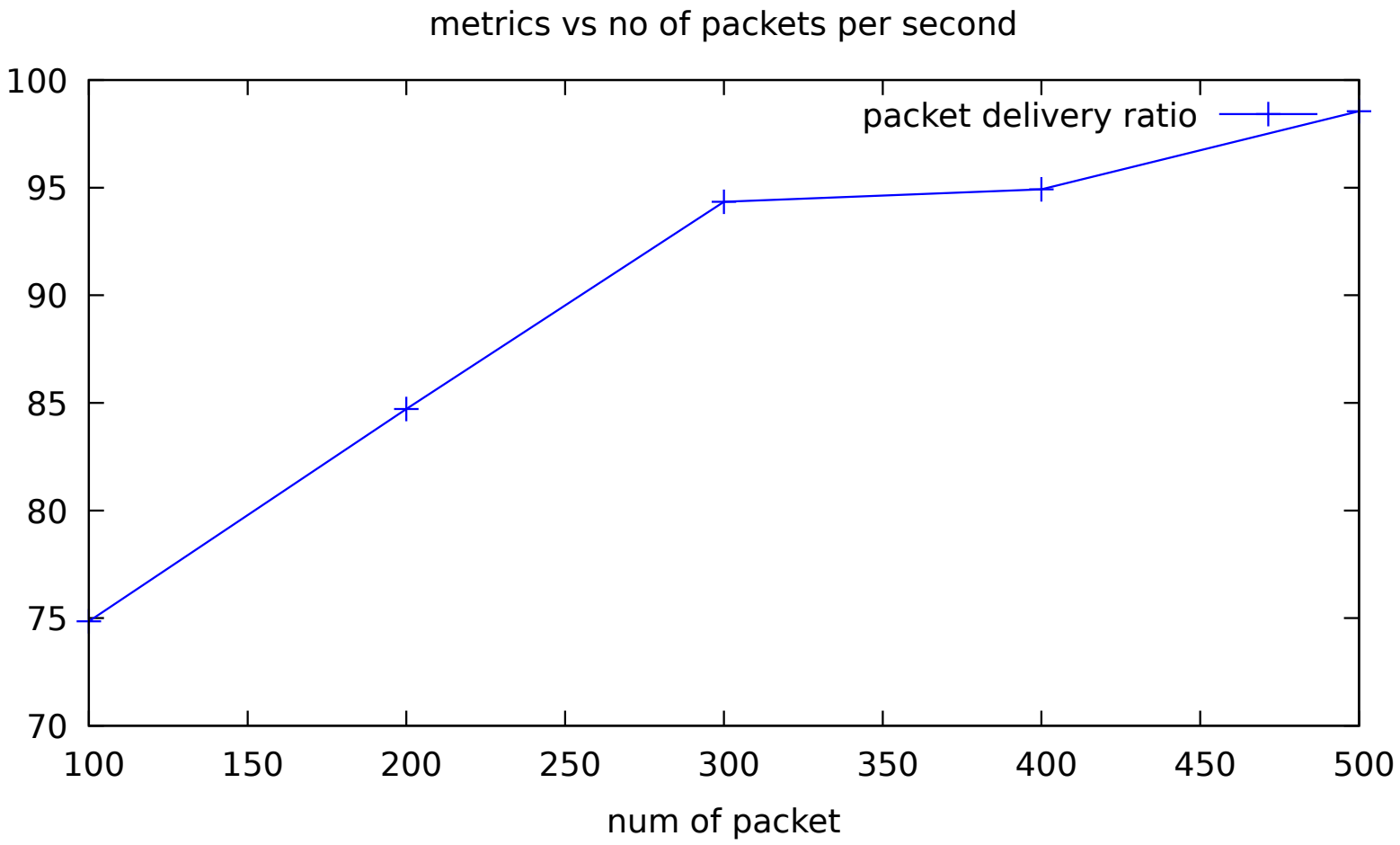

metrics vs no of flows
average delay
0.6
0.5
0.4
0.3
0.2
0.1
0
10
15
20
25
30
35
40
45
50
num of flows

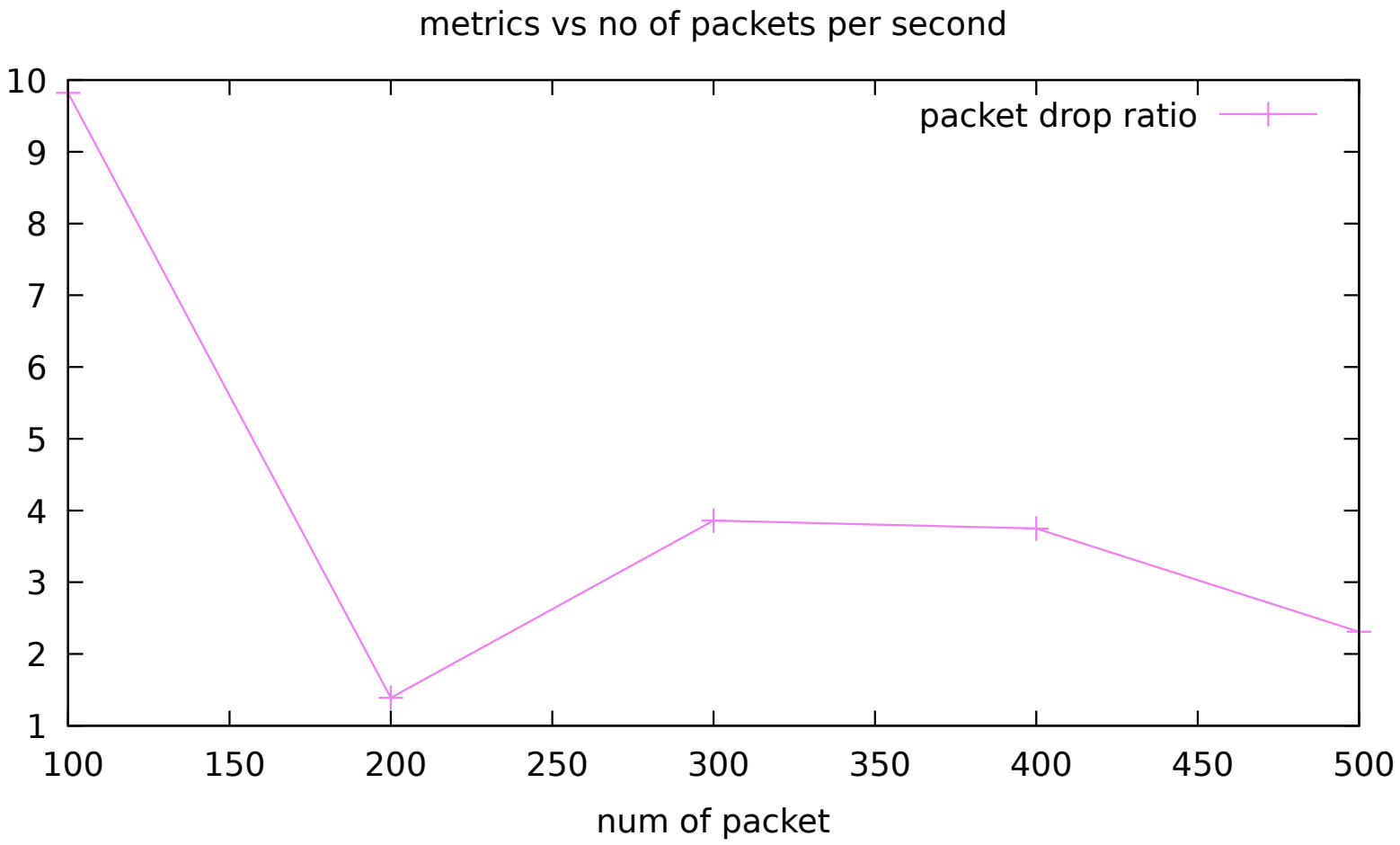

metrics vs no of flows
packet delivery ratio
86
84
82
80
78
76
74
72
70
68
10
15
20
25
30
35
40
45
50
num of flows

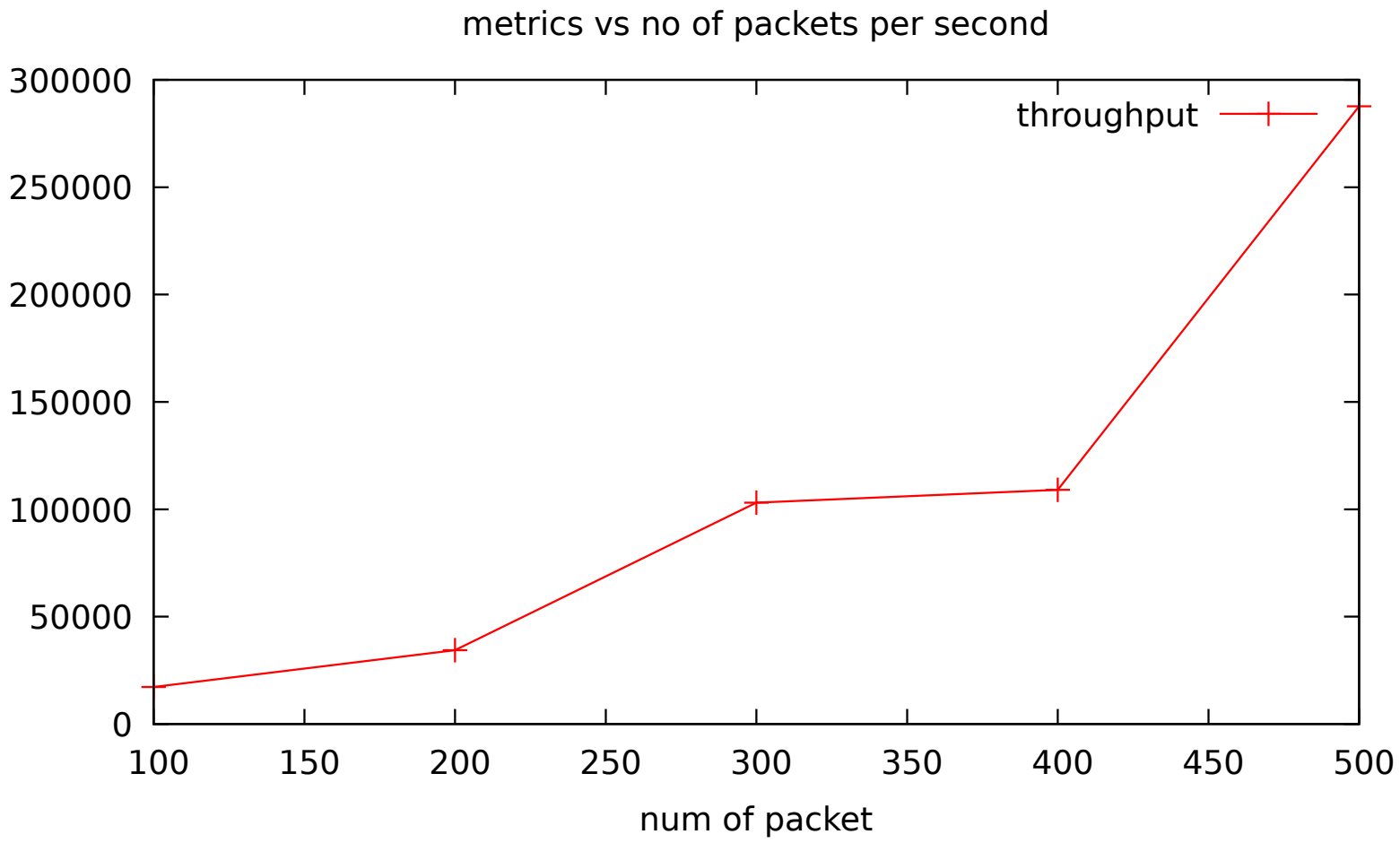

metrics vs no of flows
packet drop ratio
25
20
15
10
5
0
10
15
20
25
30
35
40
45
50
num of flows

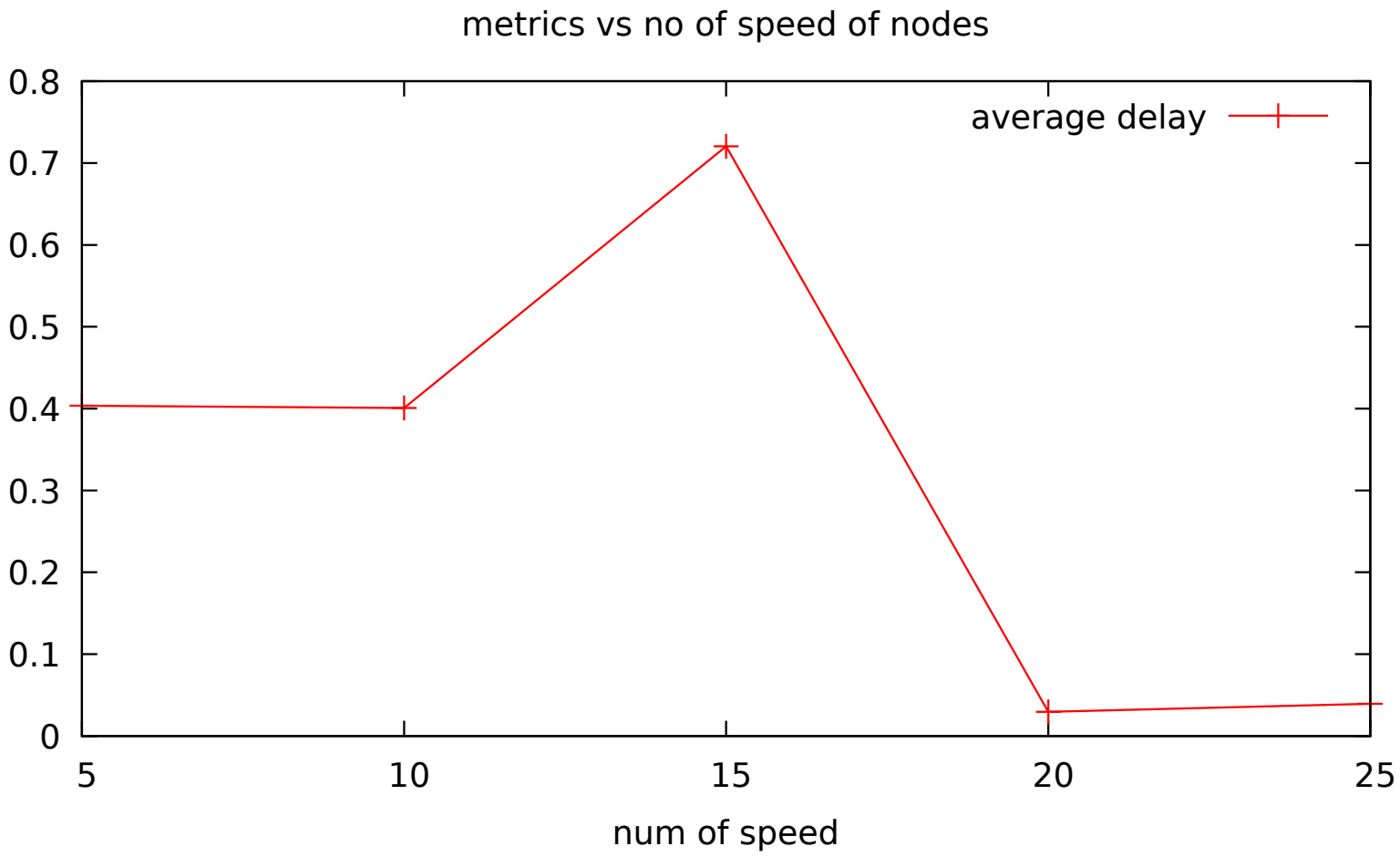

metrics vs no of flows
throughput
160000
140000
120000
100000
80000
60000
40000
20000
10
15
20
25
30
35
40
45
50
num of flows

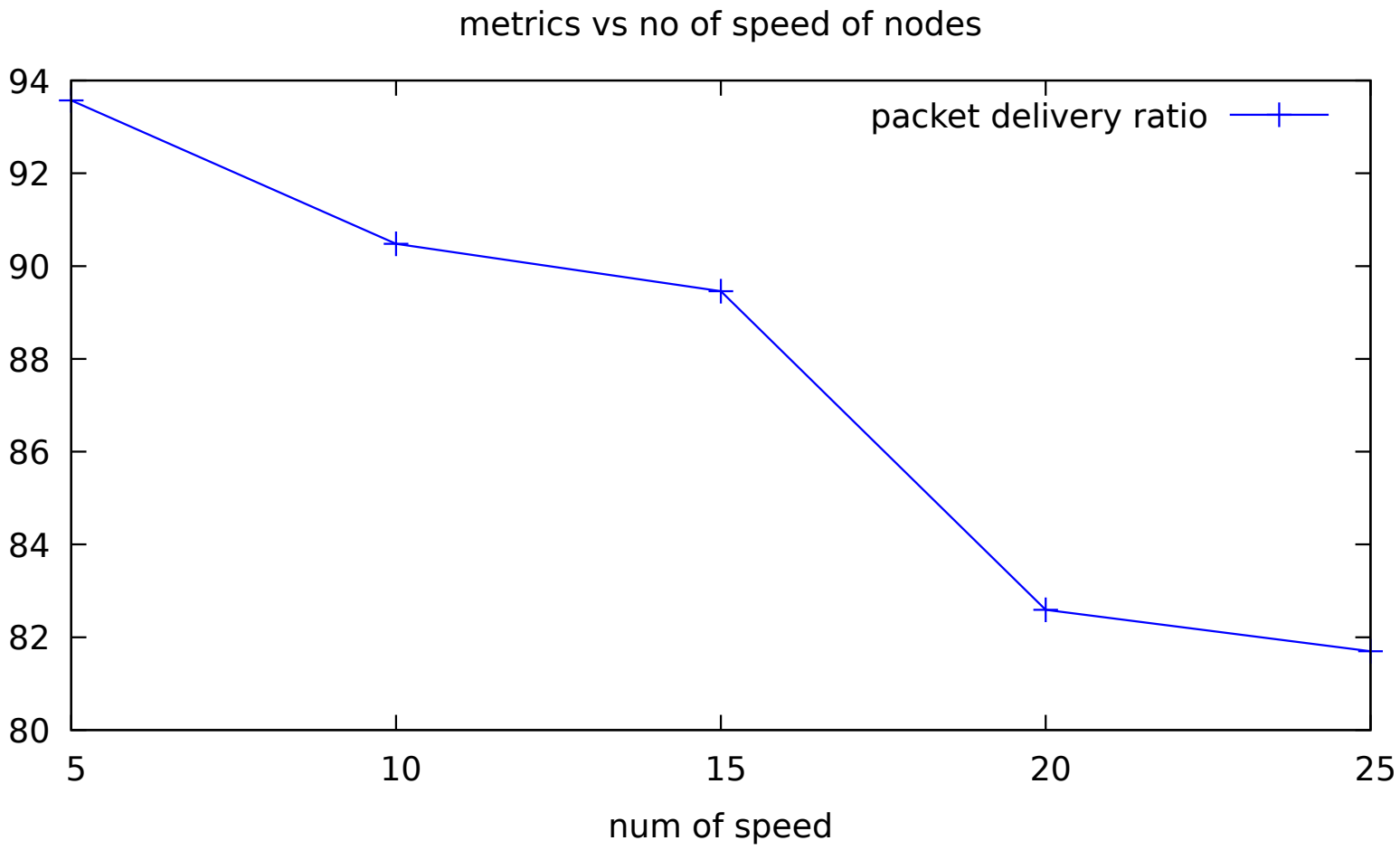

metrics vs no of packets per second
average delay
0.22
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
100
150
200
250
300
350
400
450
500
num of packet

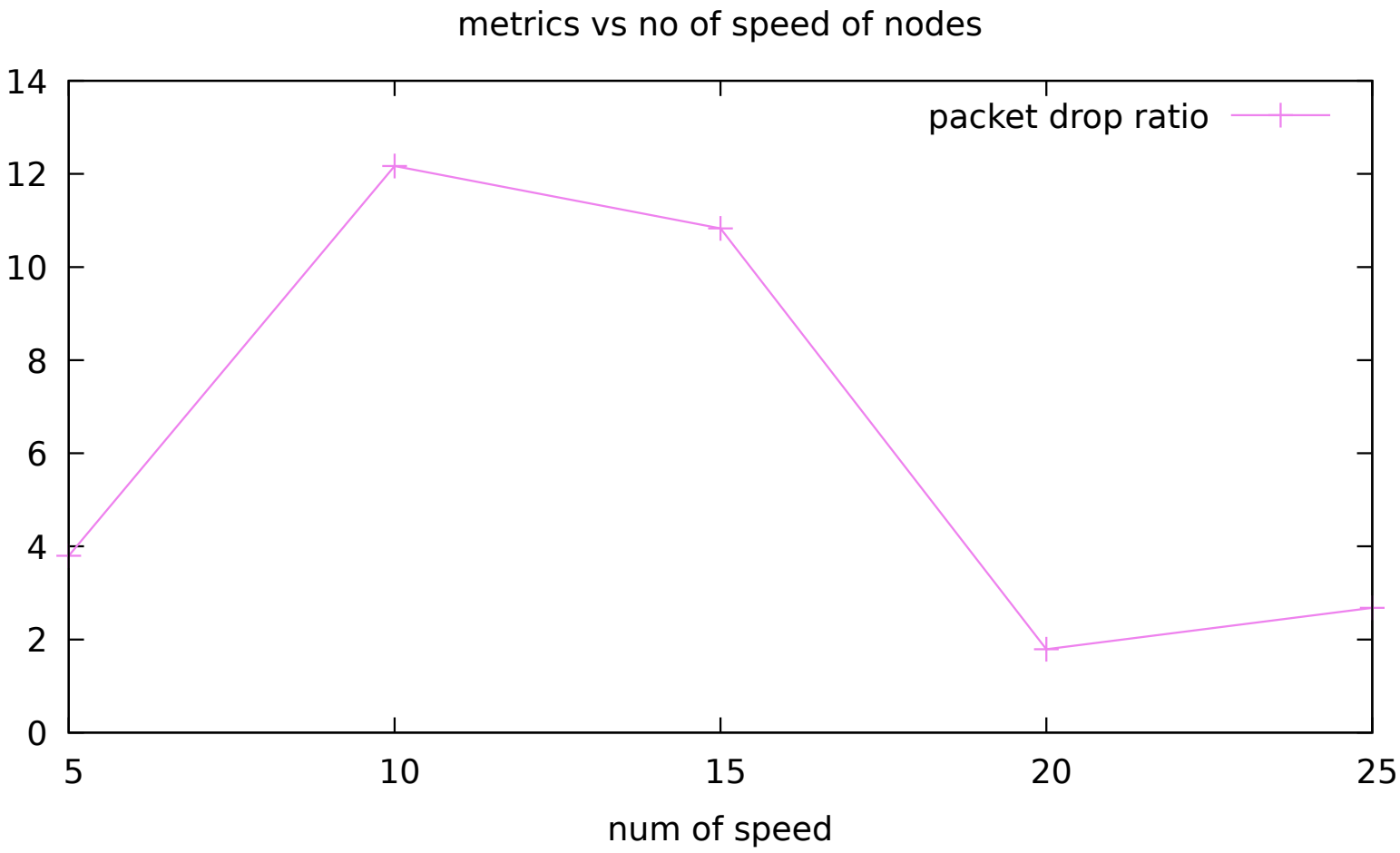

metrics vs no of packets per second
packet delivery ratio
100
95
90
85
80
75
70
100
150
200
250
300
350
400
450
500
num of packet

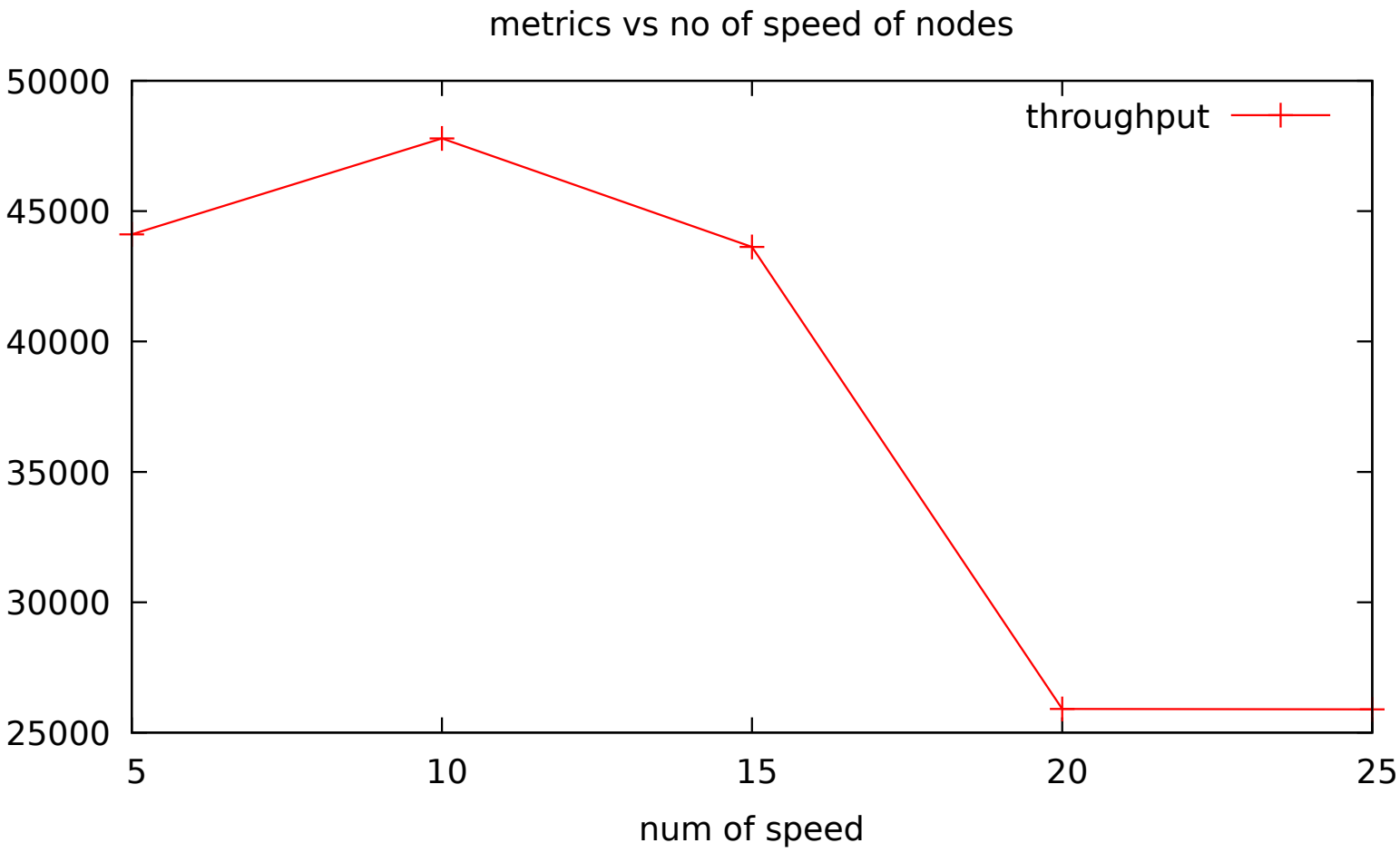

metrics vs no of packets per second
packet drop ratio
10
9
8
7
6
5
4
3
2
1
100
150
200
250
300
350
400
450
500
num of packet


metrics vs no of packets per second
throughput
300000
250000
200000
150000
100000
50000
0
100
150
200
250
300
350
400
450
500
num of packet

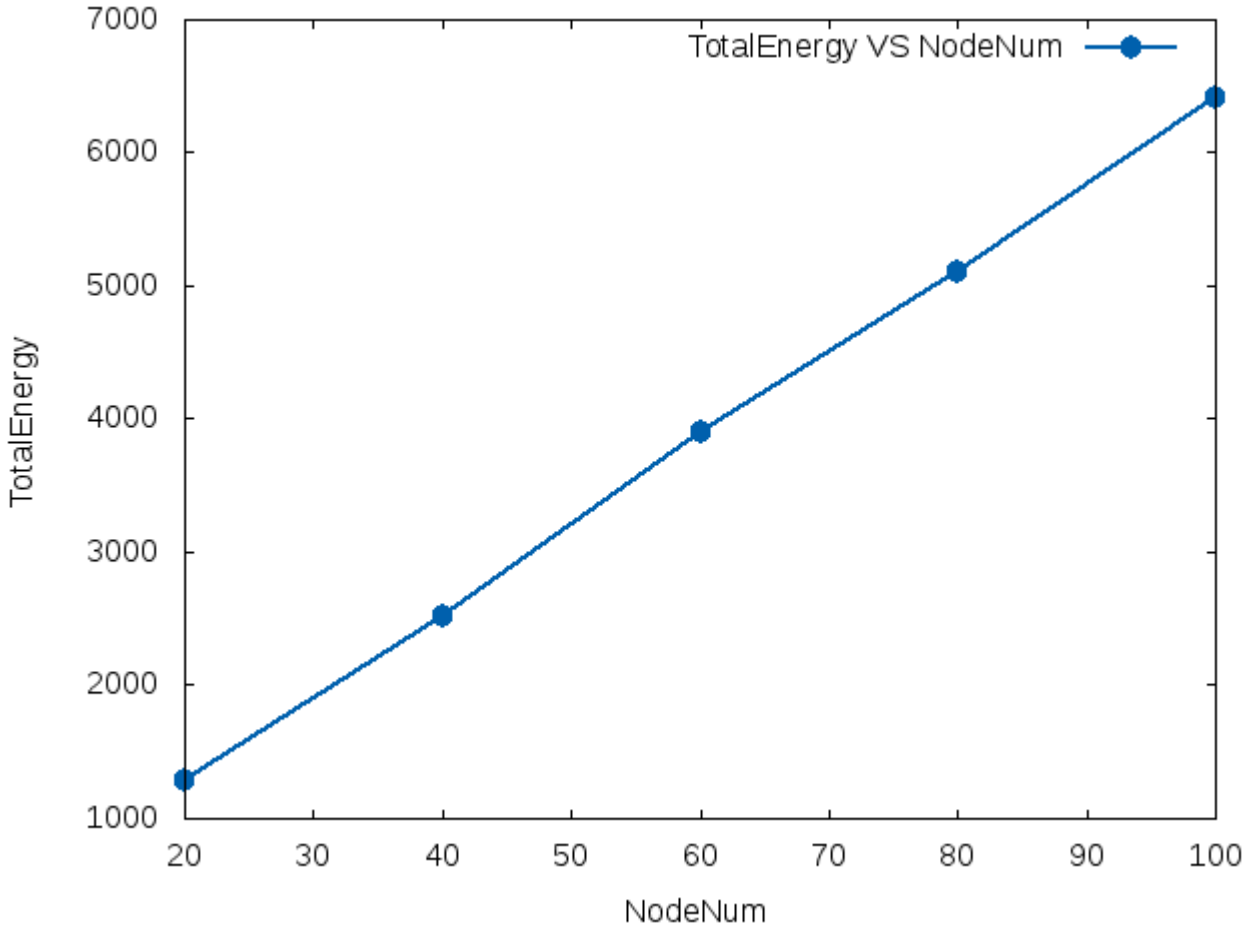

metrics vs no of speed of nodes
average delay
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
5
10
15
20
25
num of speed


metrics vs no of speed of nodes
packet delivery ratio
94
92
90
88
86
84
82
80
5
10
15
20
25
num of speed

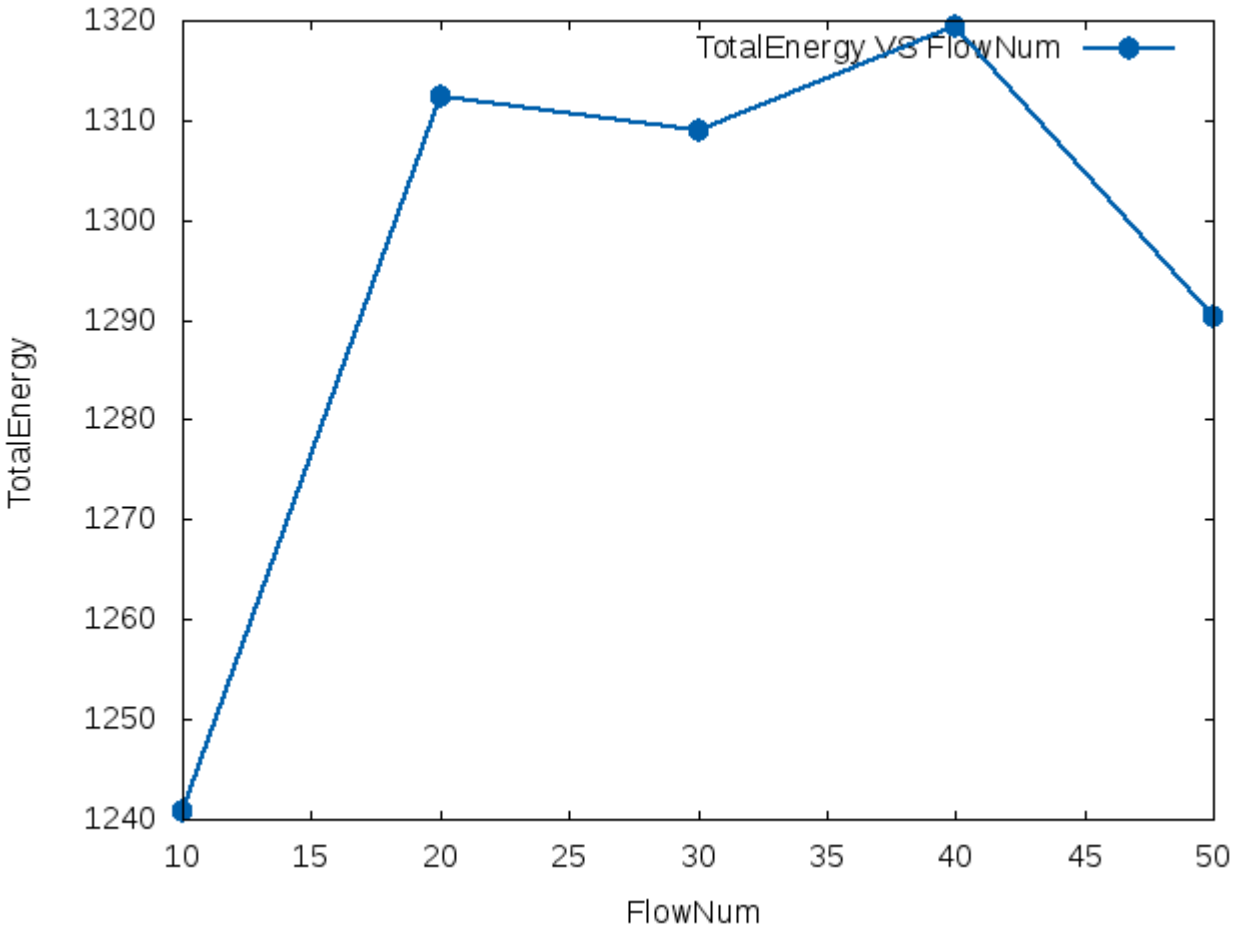

metrics vs no of speed of nodes
packet drop ratio
14
12
10
8
6
4
2
0
5
10
15
20
25
num of speed


metrics vs no of speed of nodes
50000
45000
40000
35000
30000
25000
5
10
15
20
25
num of speed
throughput

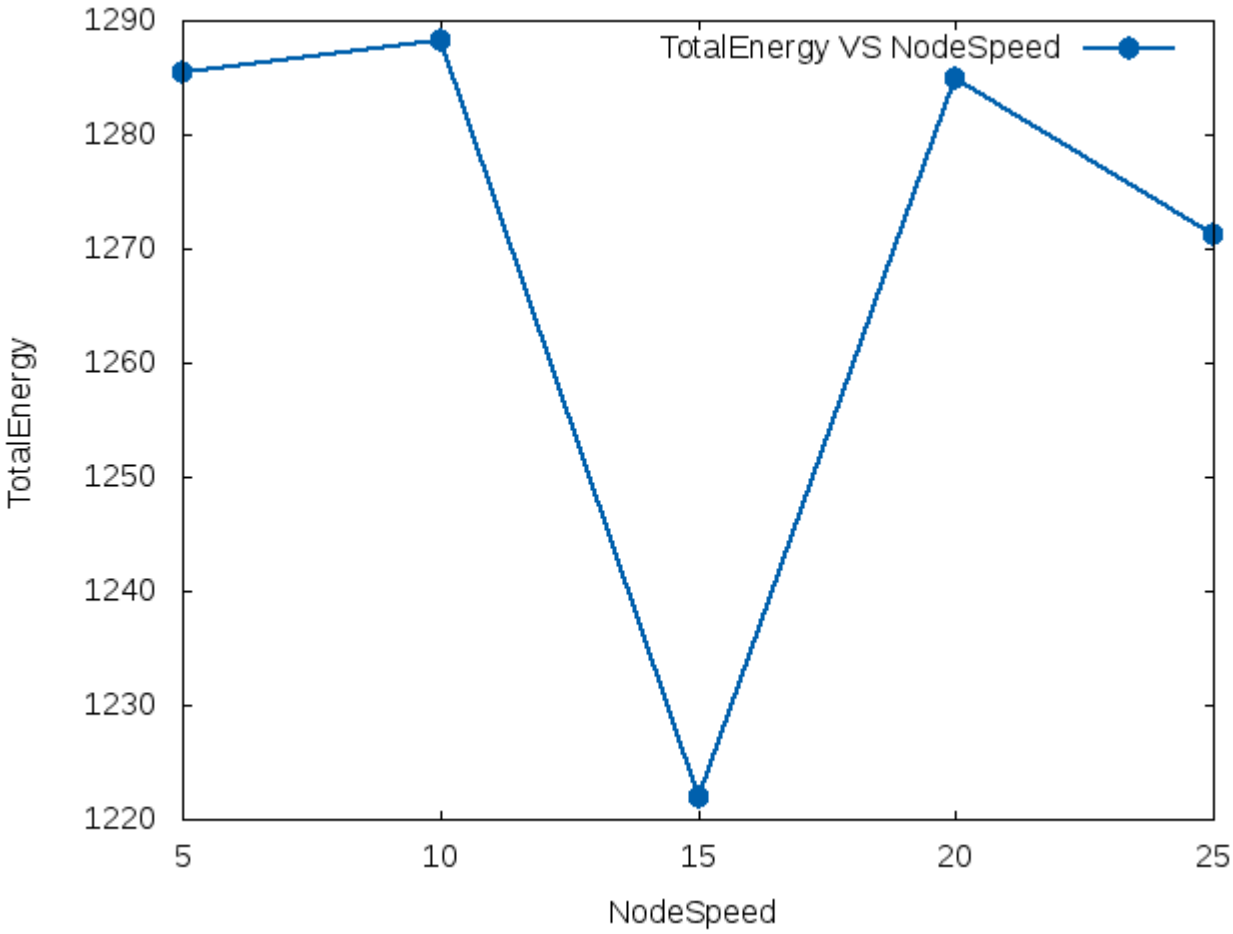
VS NodeNum.png



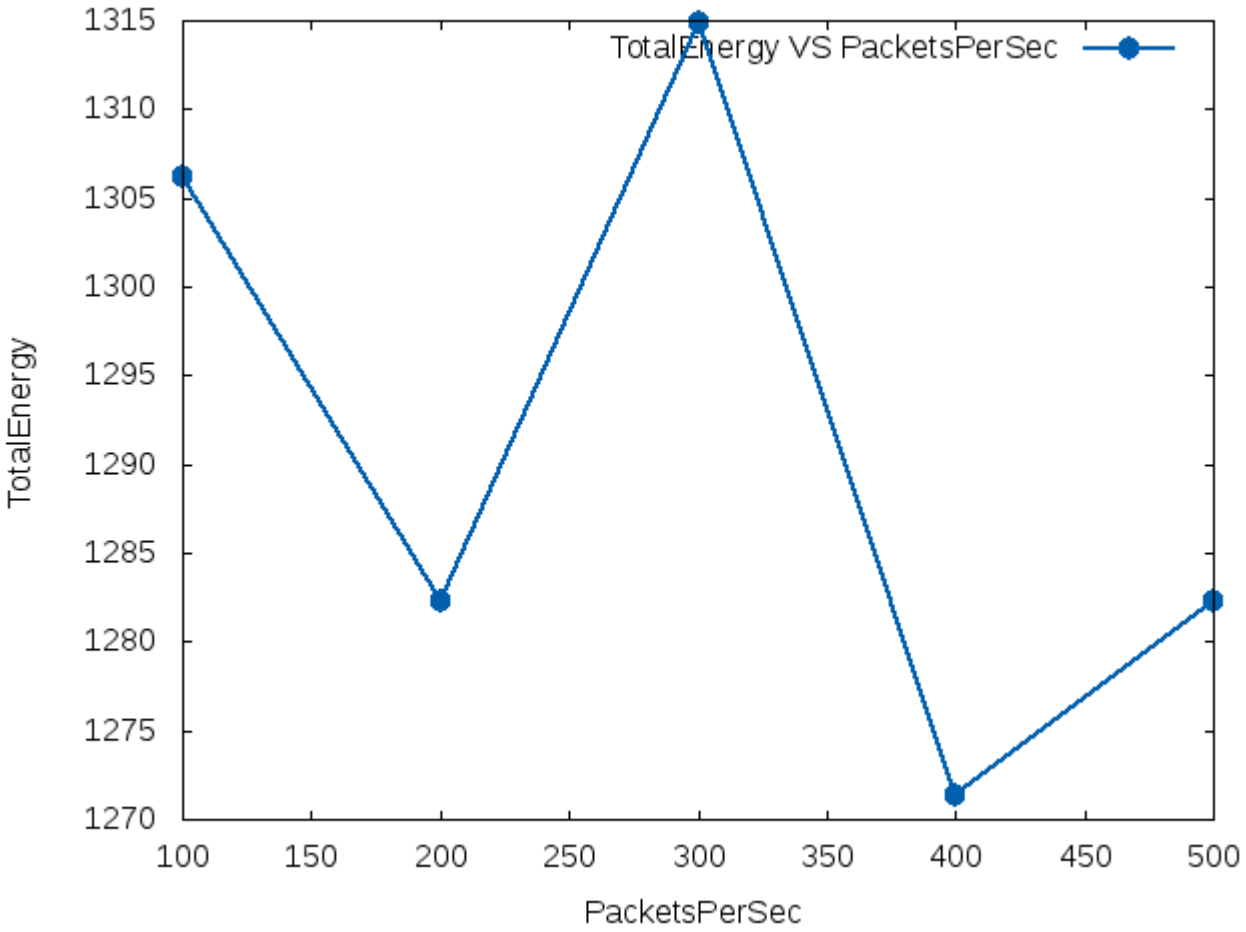
VS FlowNum.png



VS NodeSpeed.png

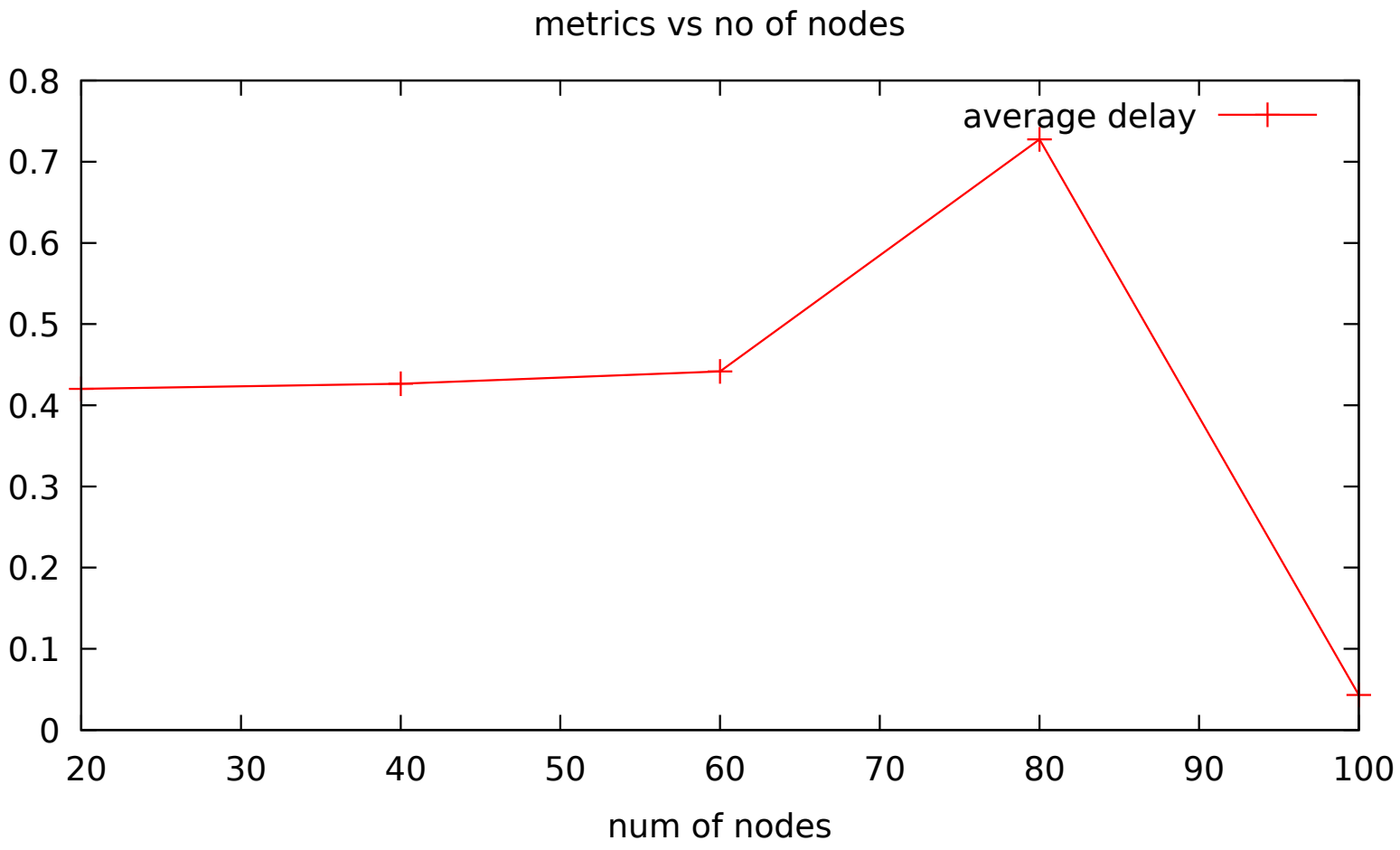


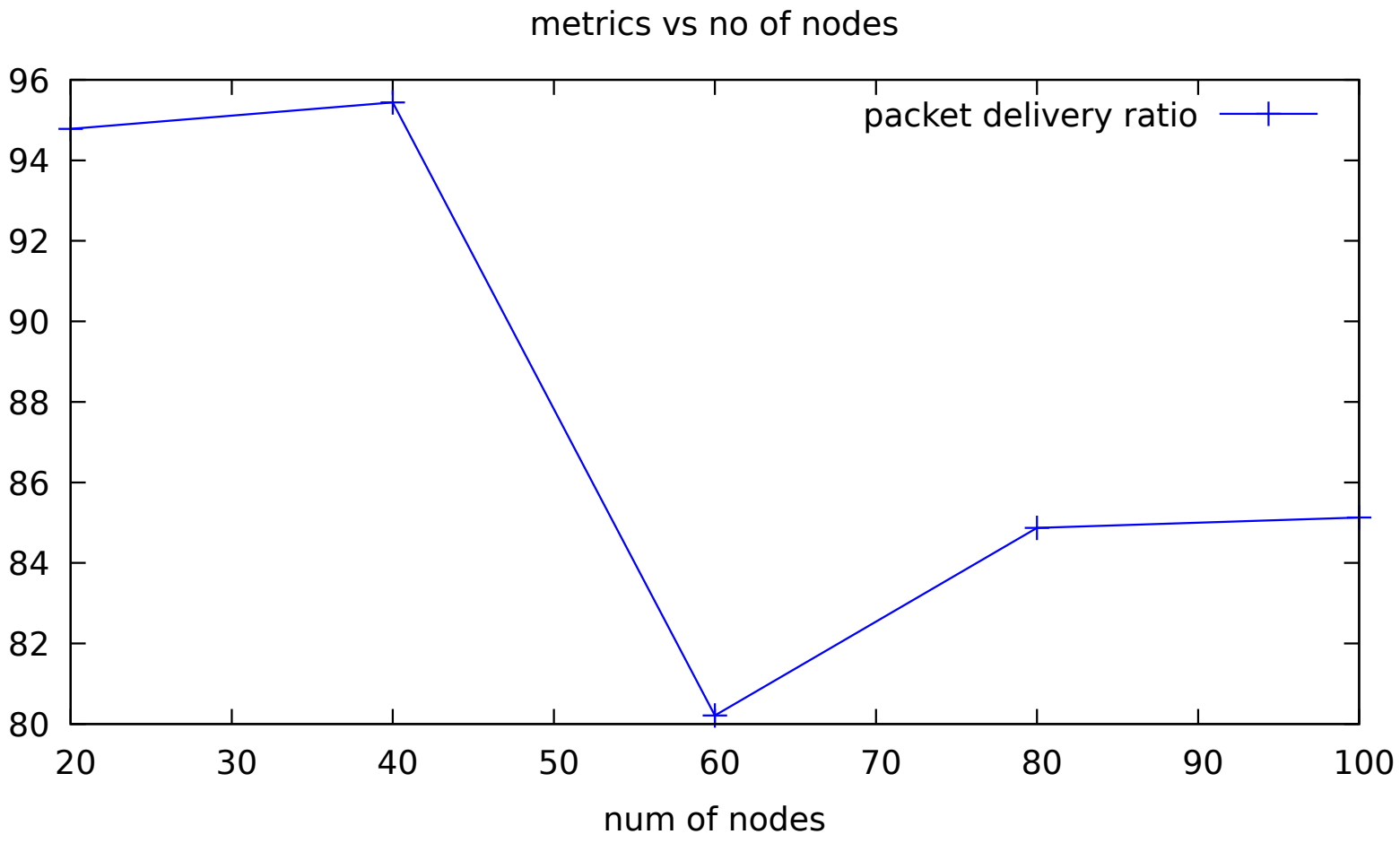
VS PacketsPerSec.png

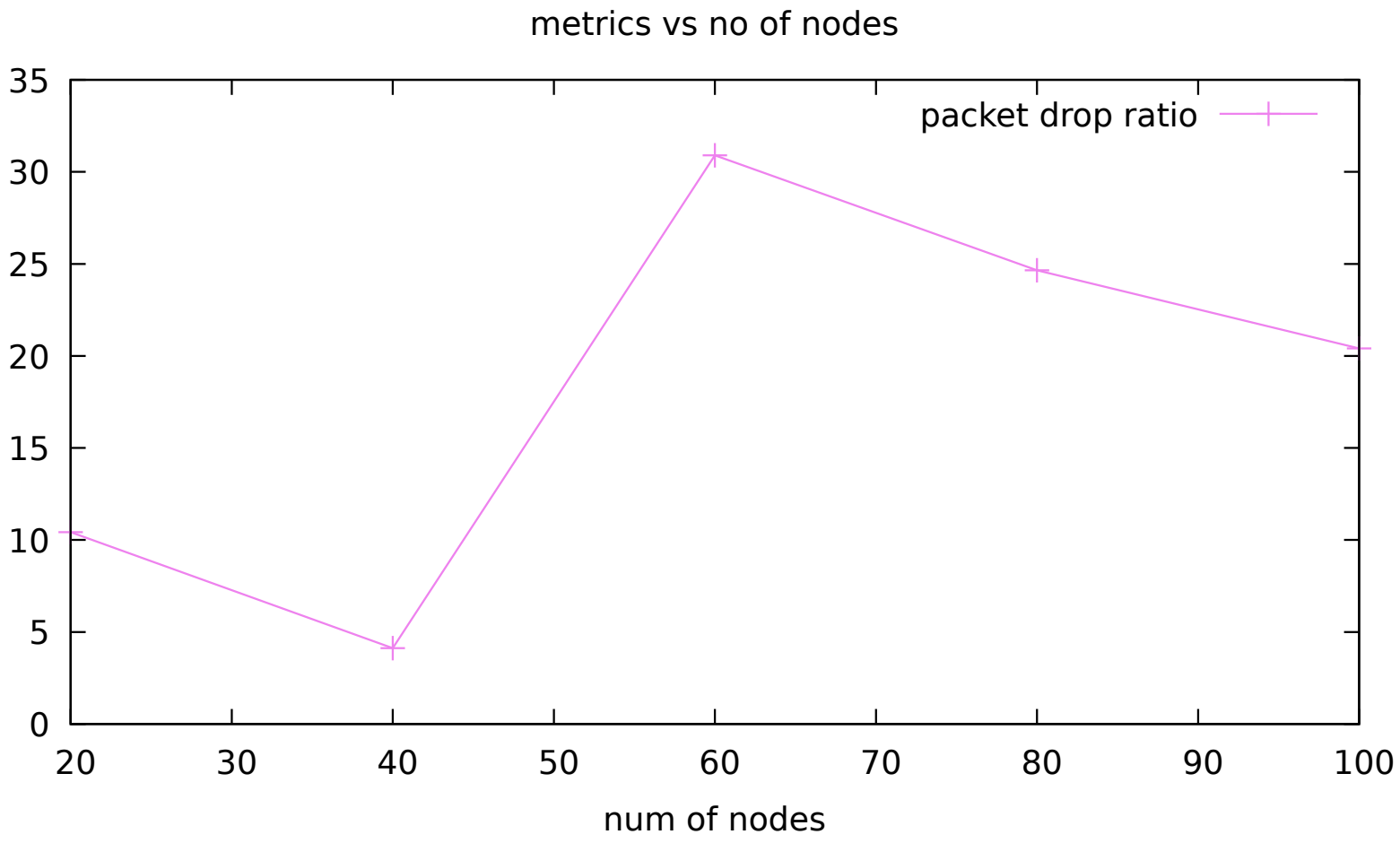

TotalEnergy VS PacketsPerSec
TotalEnergy
PacketsPerSec
1315
1310
1305
1300
1295
1290
1285
1280
1275
1270
100
150
200
250
300
350
400
450
500

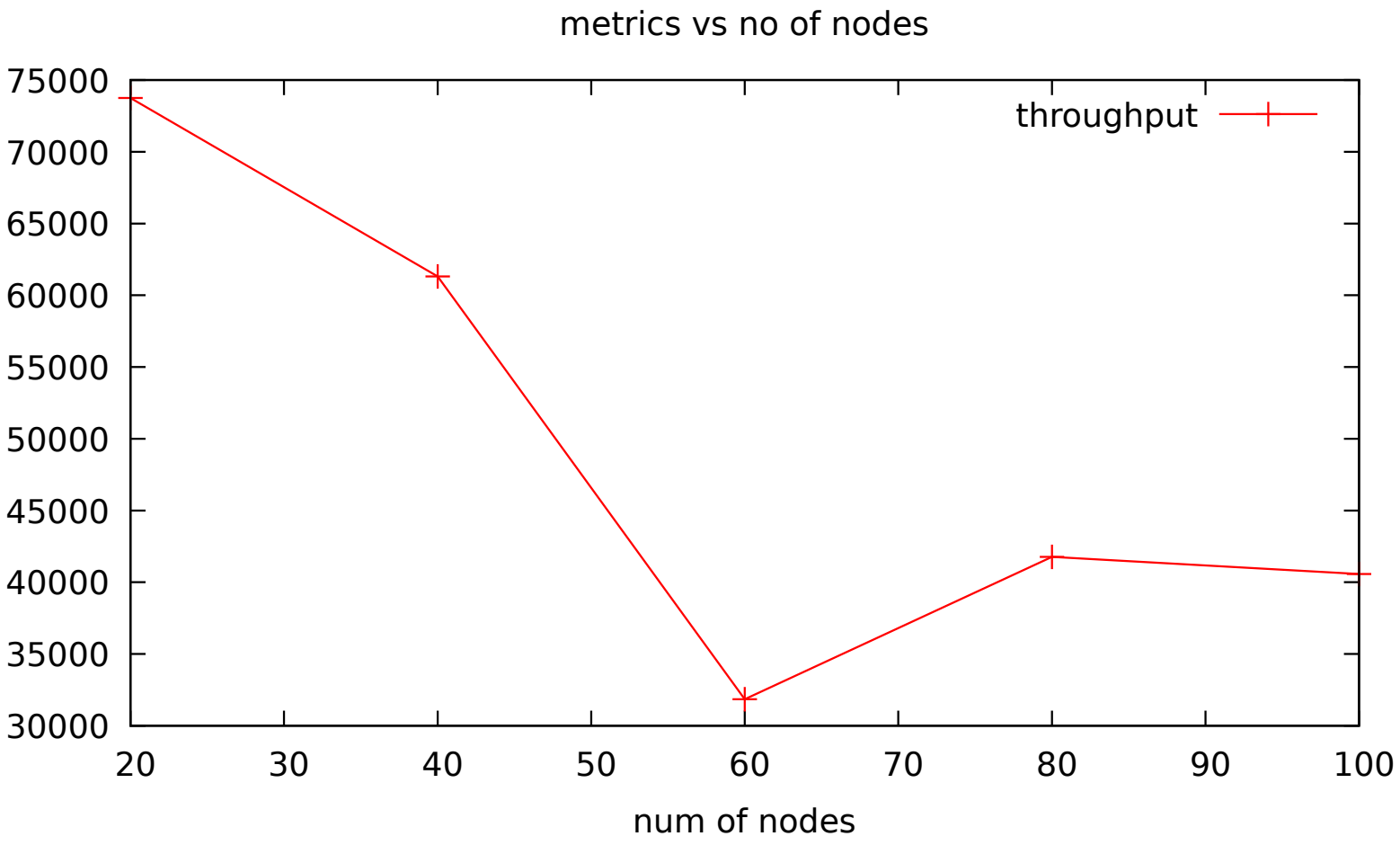
6.2 802.11 Mobile Modified

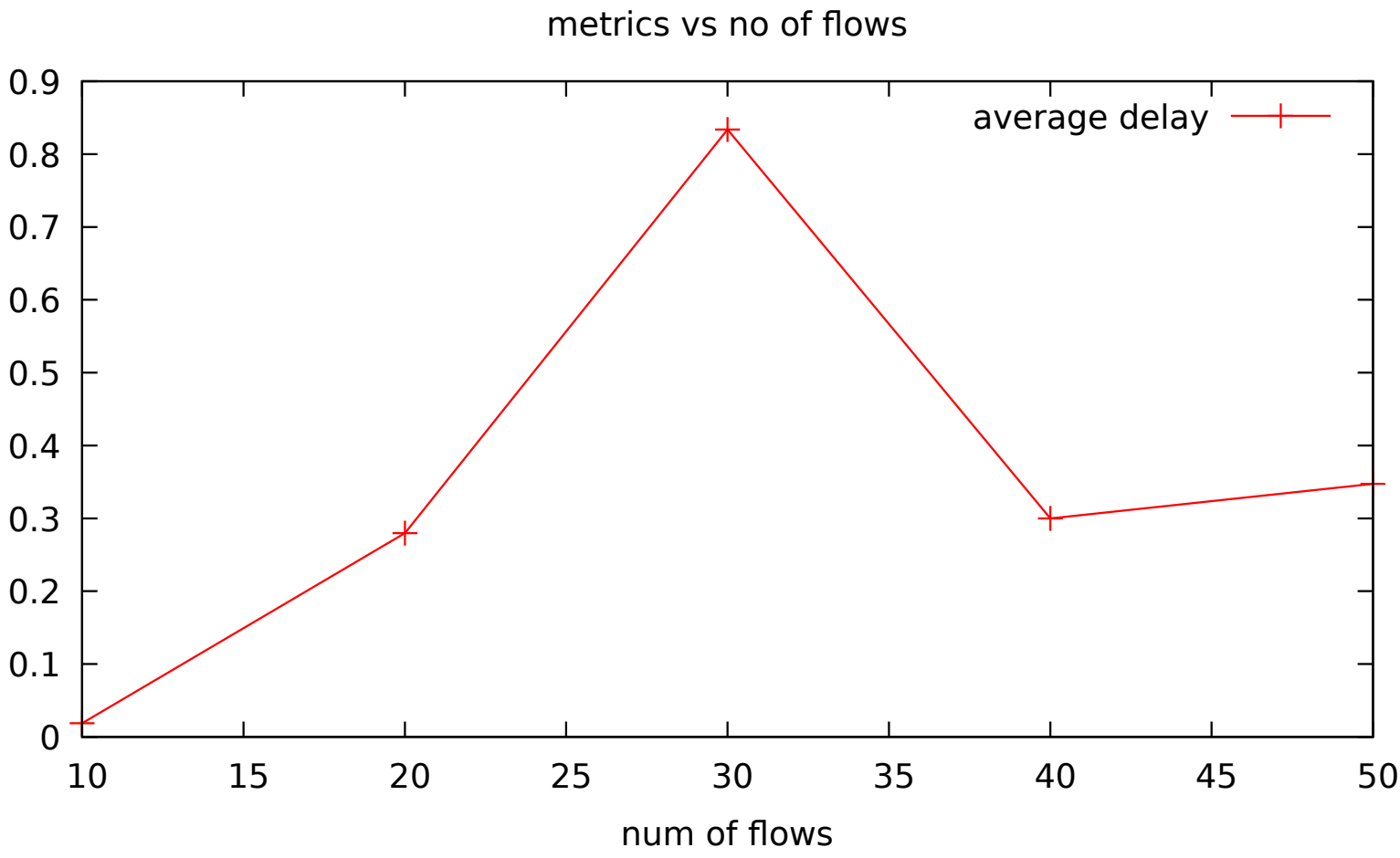
After modifying tcp, the following graphs are generated. Here DSDV routing protocol is used.

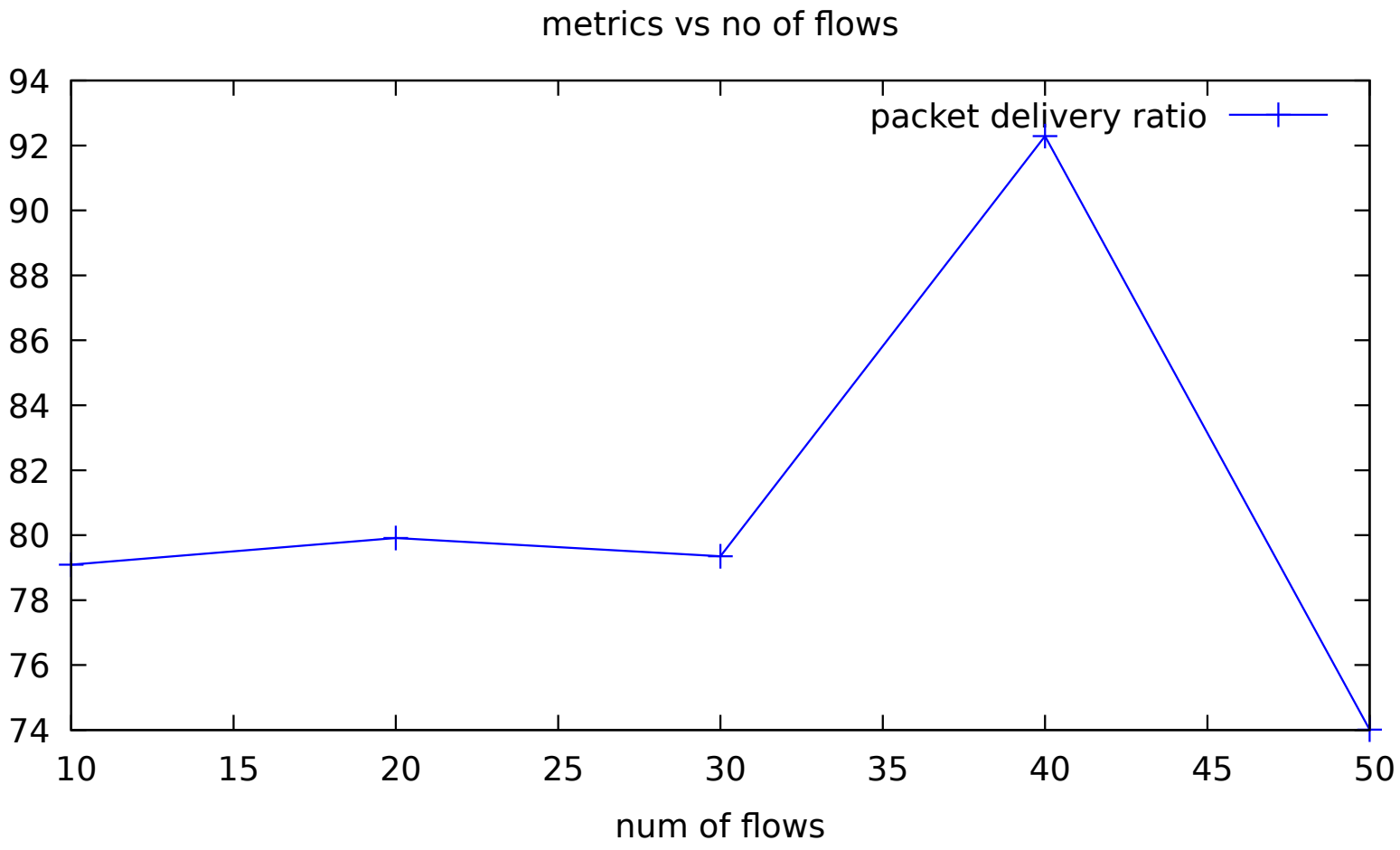

metrics vs no of nodes
average delay
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
20
30
40
50
60
70
80
90
100
num of nodes

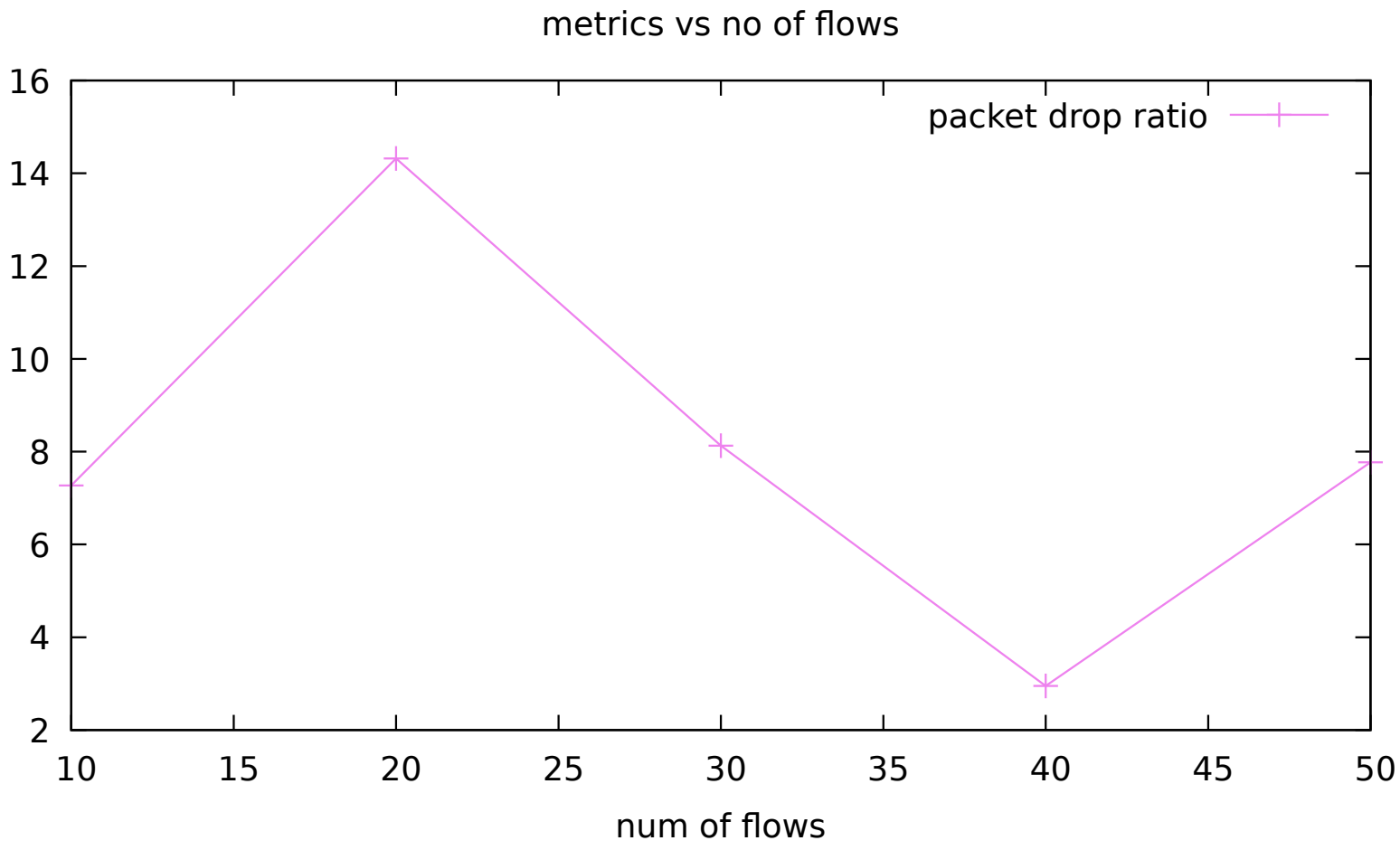

metrics vs no of nodes
packet delivery ratio
96
94
92
90
88
86
84
82
80
20
30
40
50
60
70
80
90
100
num of nodes

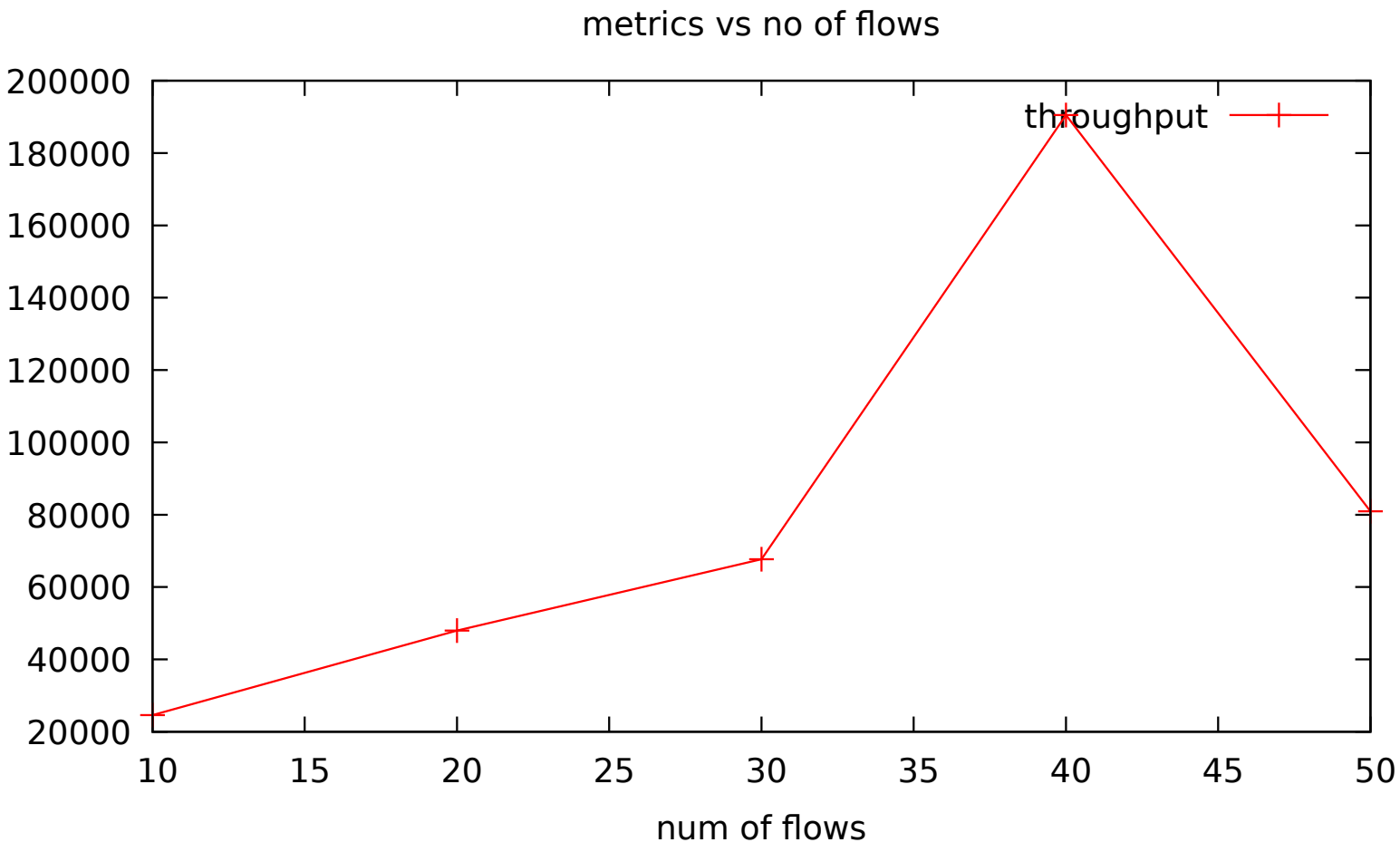

metrics vs no of nodes
packet drop ratio
35
30
25
20
15
10
5
0
20
30
40
50
60
70
80
90
100
num of nodes

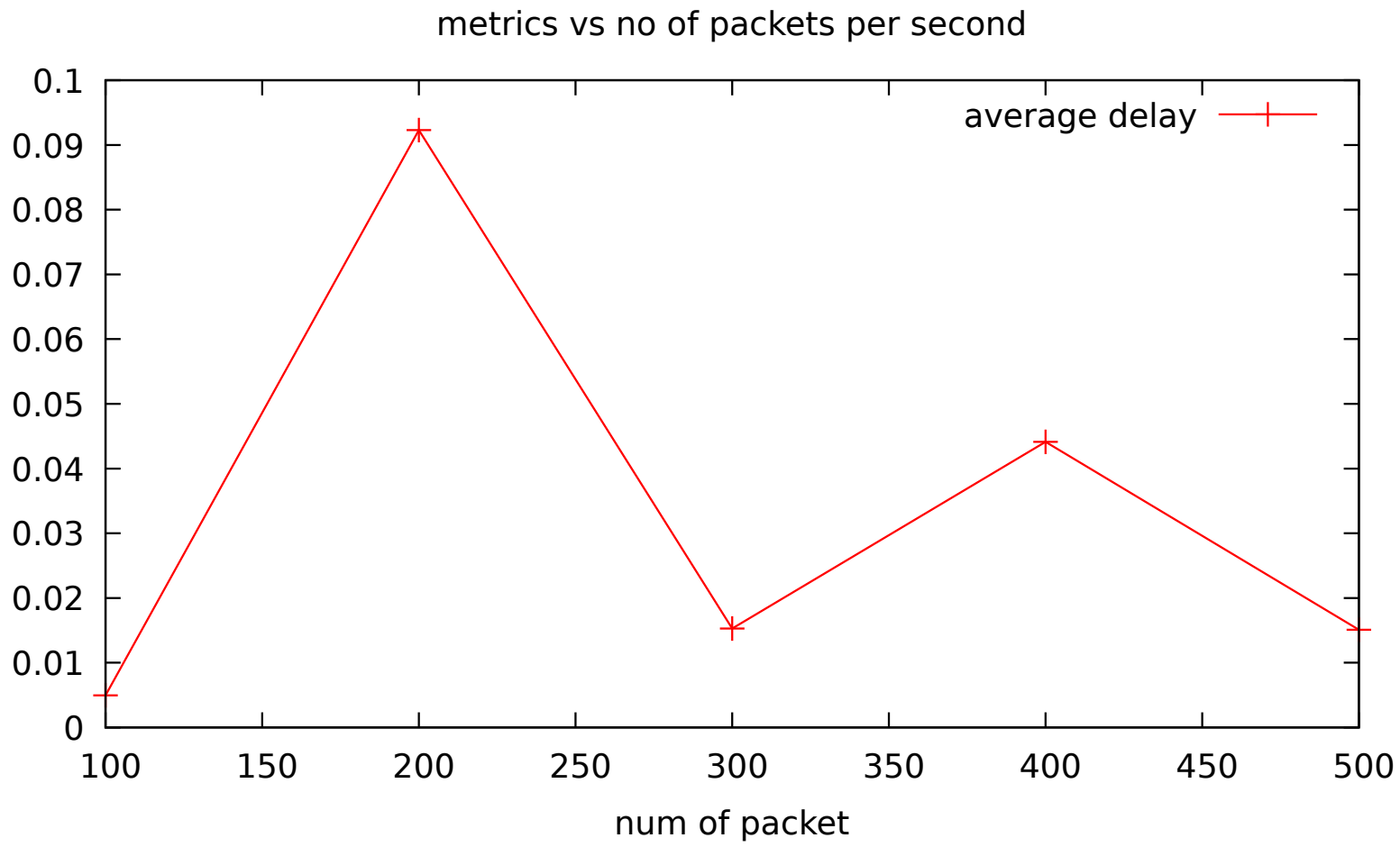

metrics vs no of nodes
throughput
75000
70000
65000
60000
55000
50000
45000
40000
35000
30000
20
30
40
50
60
70
80
90
100
num of nodes

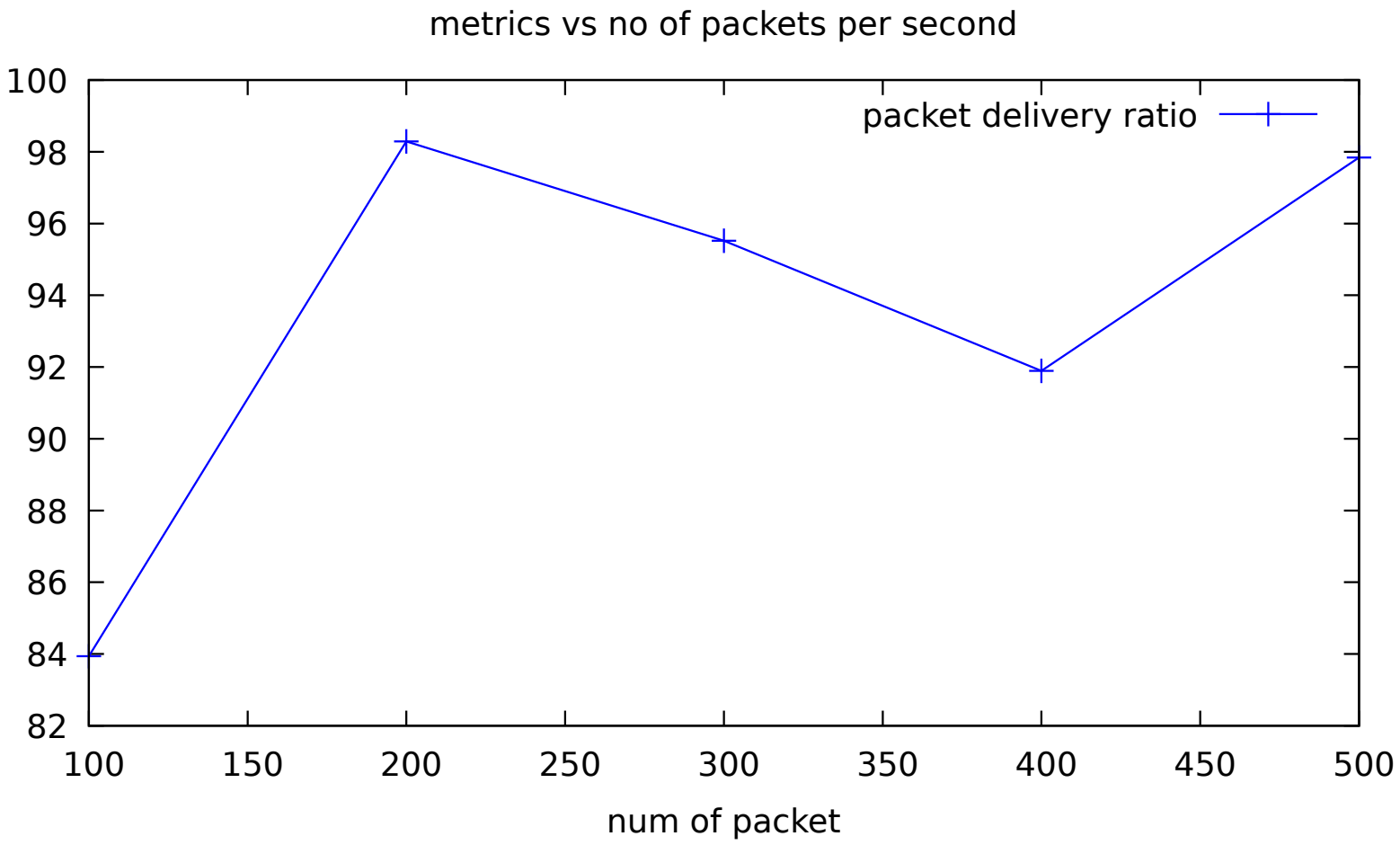

metrics vs no of flows
average delay
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
10
15
20
25
30
35
40
45
50
num of flows

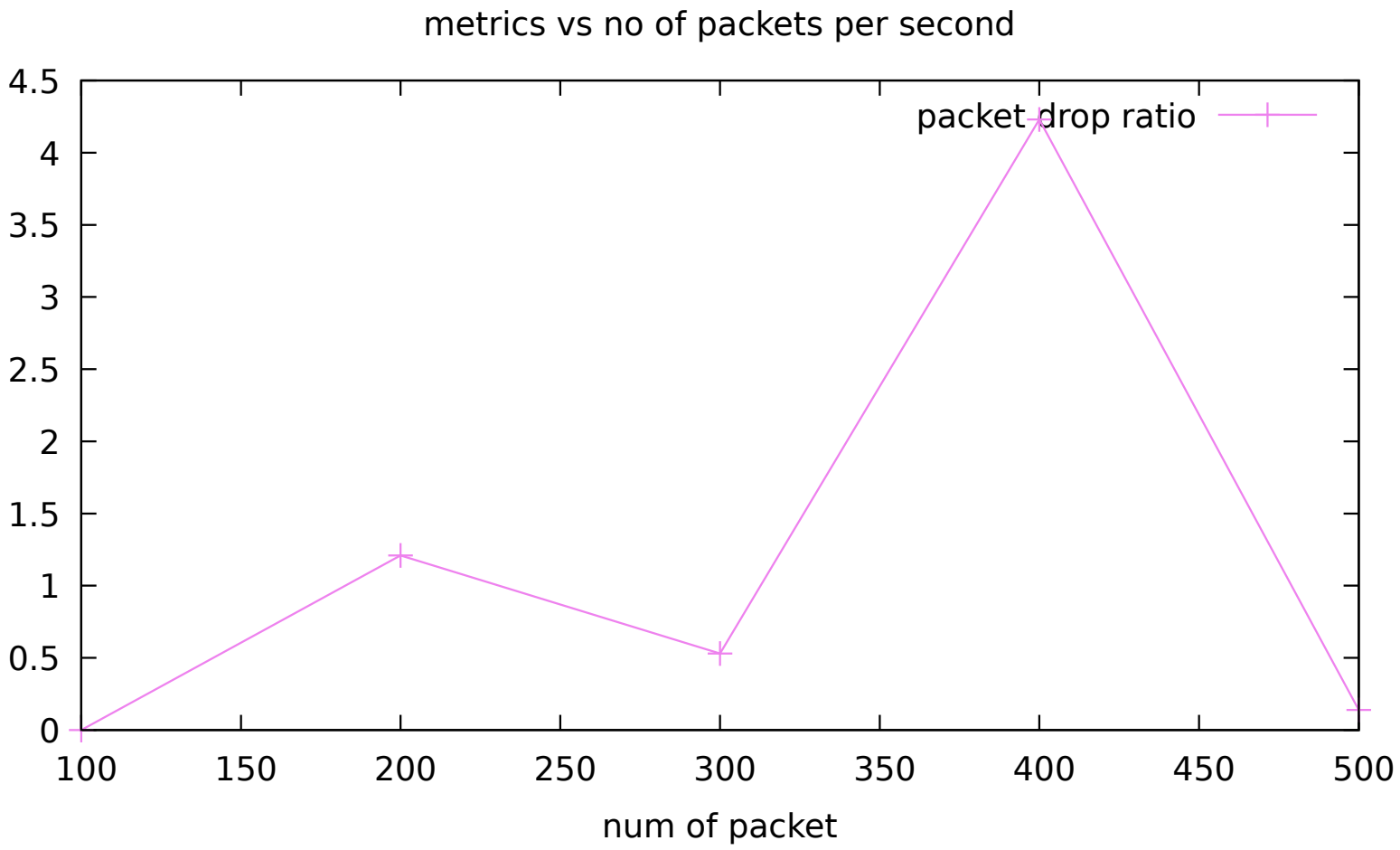

metrics vs no of flows
packet delivery ratio
94
92
90
88
86
84
82
80
78
76
74
10
15
20
25
30
35
40
45
50
num of flows

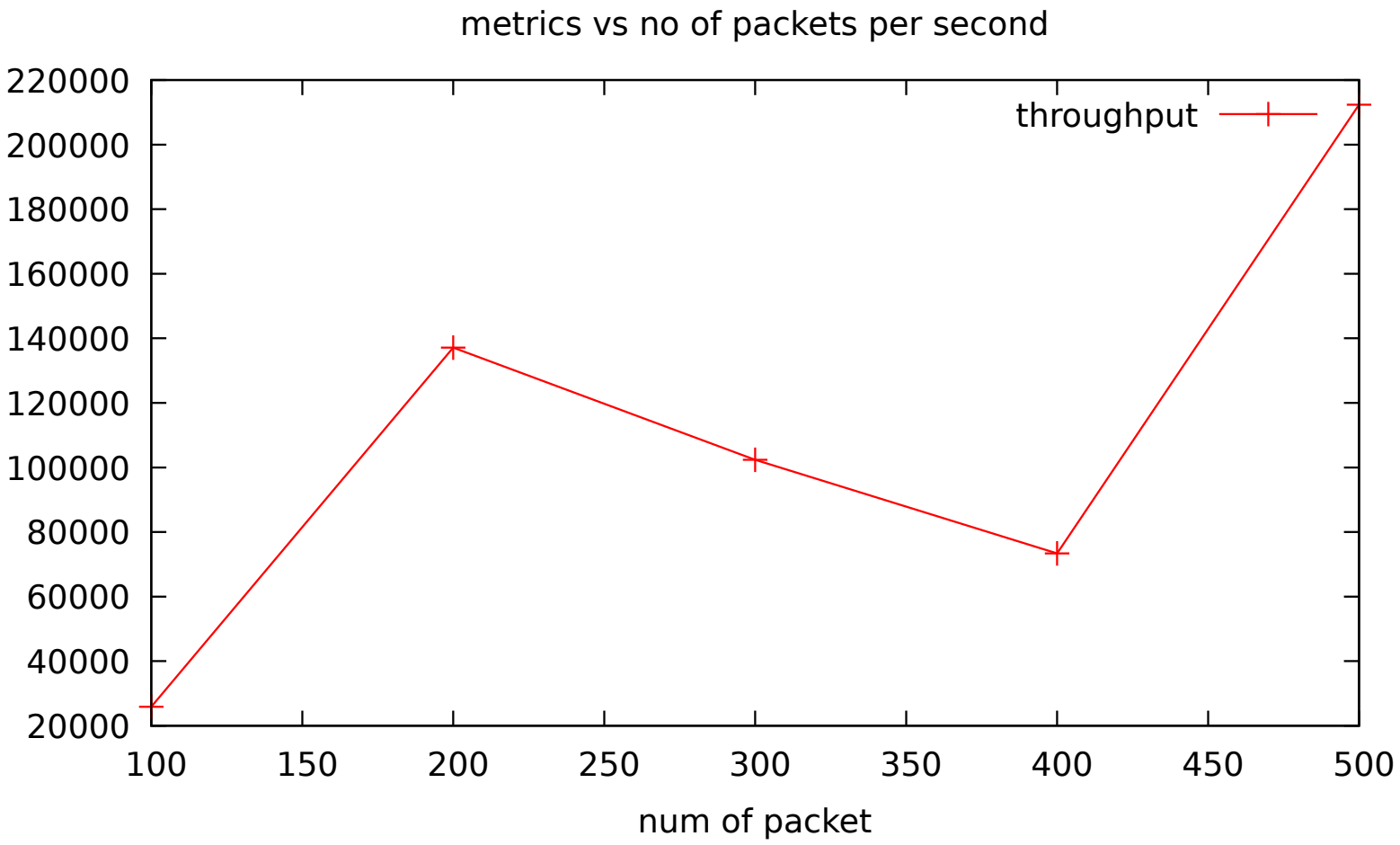

metrics vs no of flows
packet drop ratio
16
14
12
10
8
6
4
2
10
15
20
25
30
35
40
45
50
num of flows

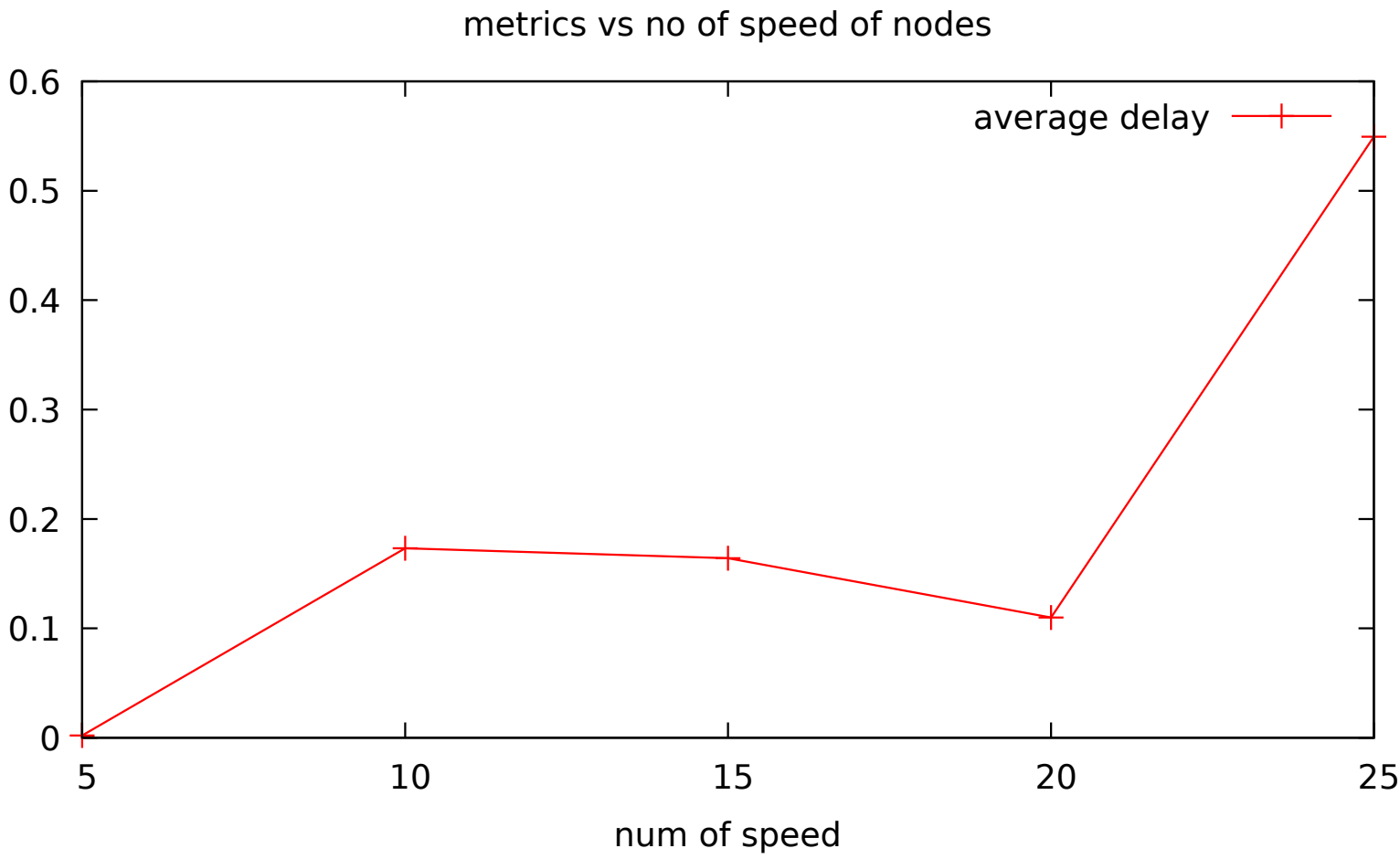

metrics vs no of flows
throughput
200000
180000
160000
140000
120000
100000
80000
60000
40000
20000
10
15
20
25
30
35
40
45
50
num of flows

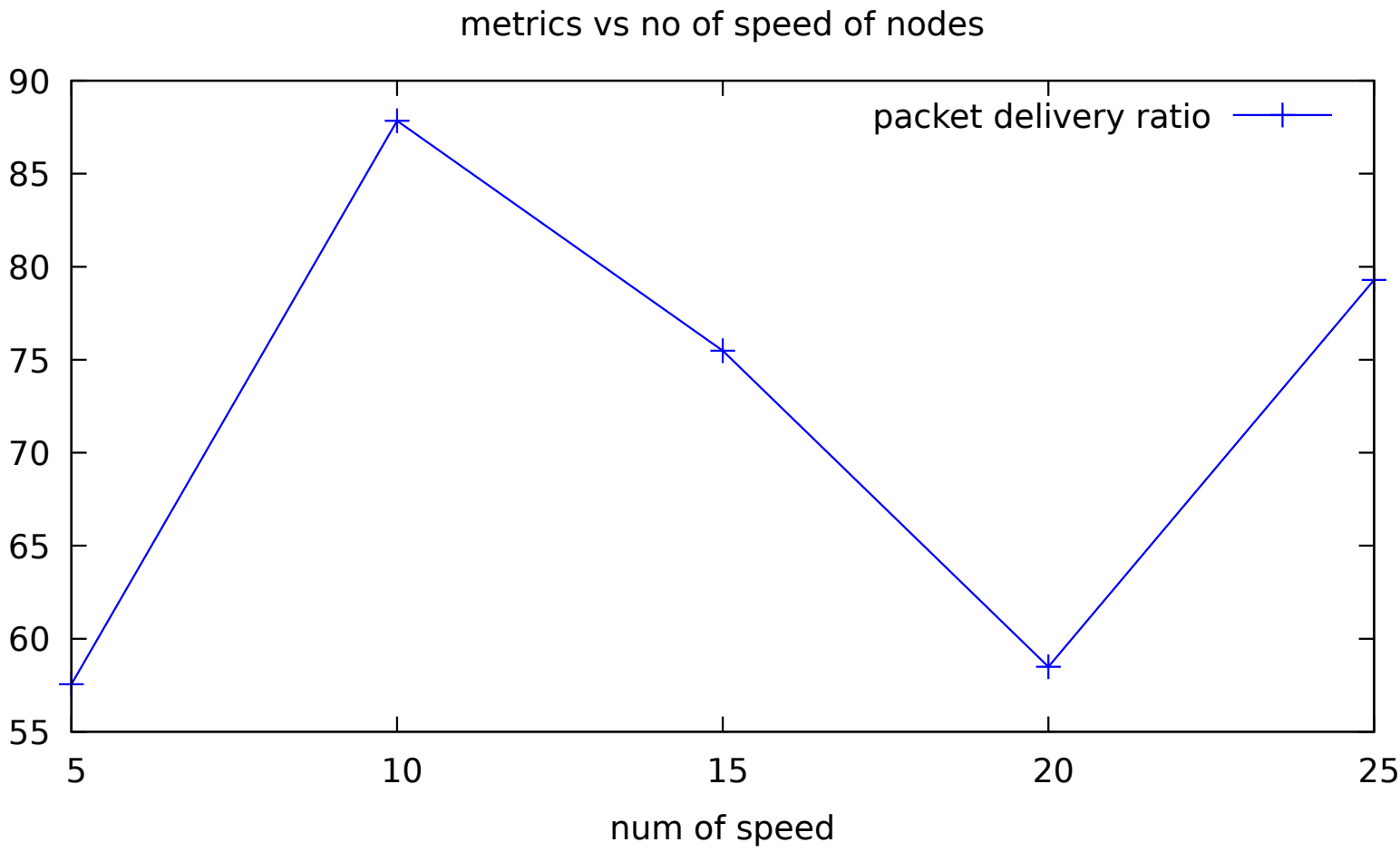

metrics vs no of packets per second
average delay
0.1
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
100
150
200
250
300
350
400
450
500
num of packet

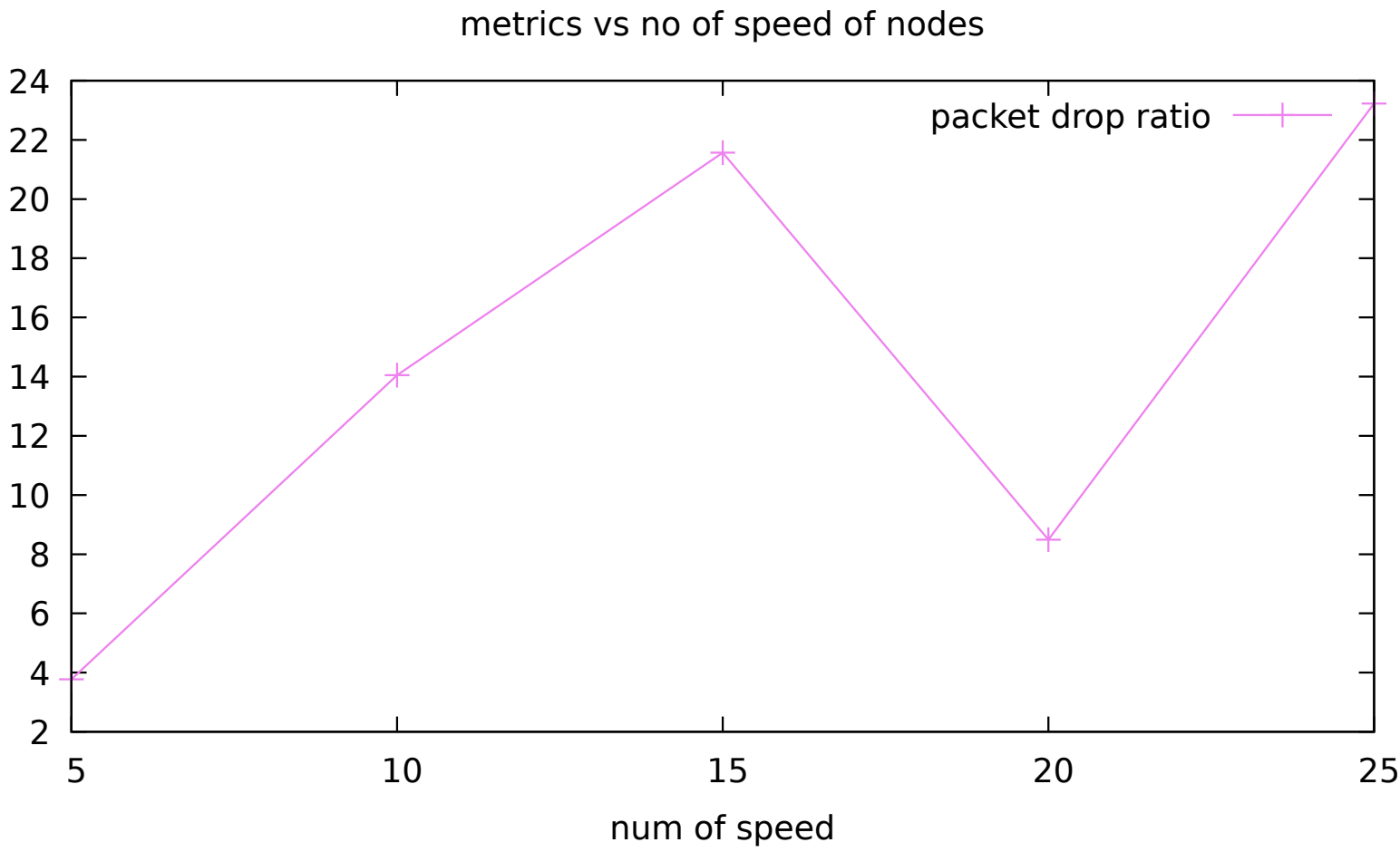

metrics vs no of packets per second
packet delivery ratio
100
98
96
94
92
90
88
86
84
82
100
150
200
250
300
350
400
450
500
num of packet

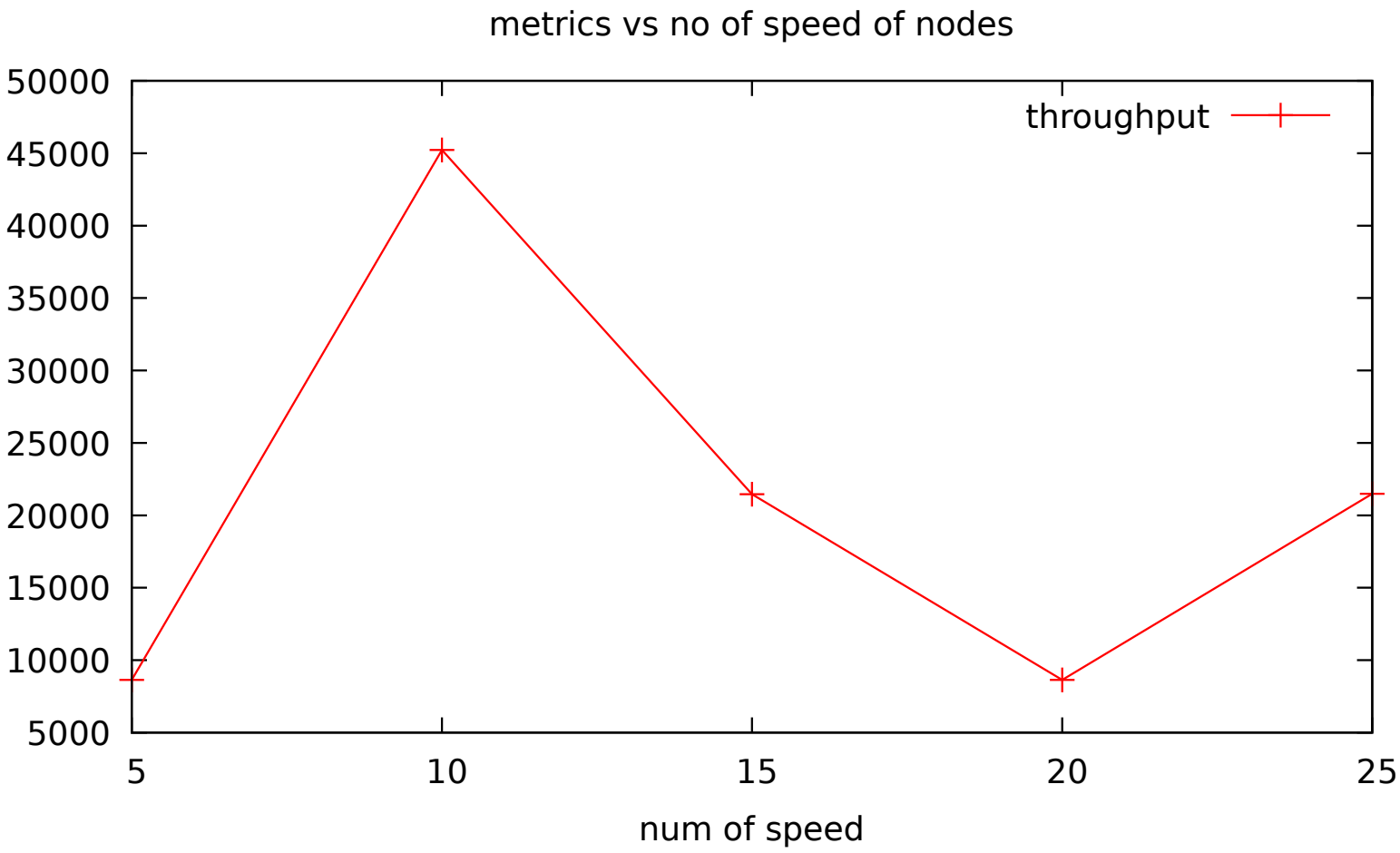

metrics vs no of packets per second
packet drop ratio
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
100
150
200
250
300
350
400
450
500
num of packet


metrics vs no of packets per second
throughput
220000
200000
180000
160000
140000
120000
100000
80000
60000
40000
20000
100
150
200
250
300
350
400
450
500
num of packet

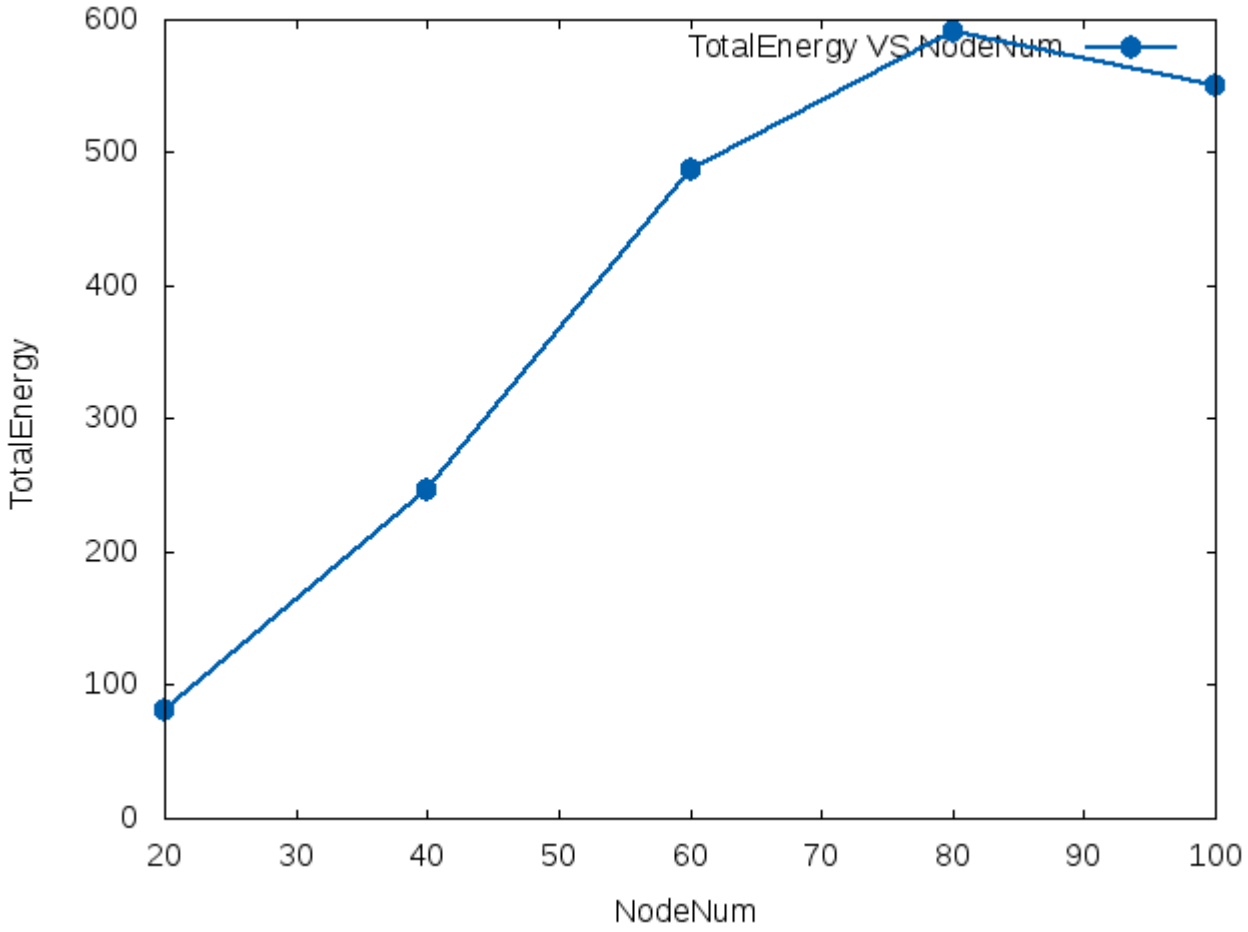

metrics vs no of speed of nodes
average delay
0.6
0.5
0.4
0.3
0.2
0.1
0
5
10
15
20
25
num of speed


metrics vs no of speed of nodes
packet delivery ratio
90
85
80
75
70
65
60
55
5
10
15
20
25
num of speed

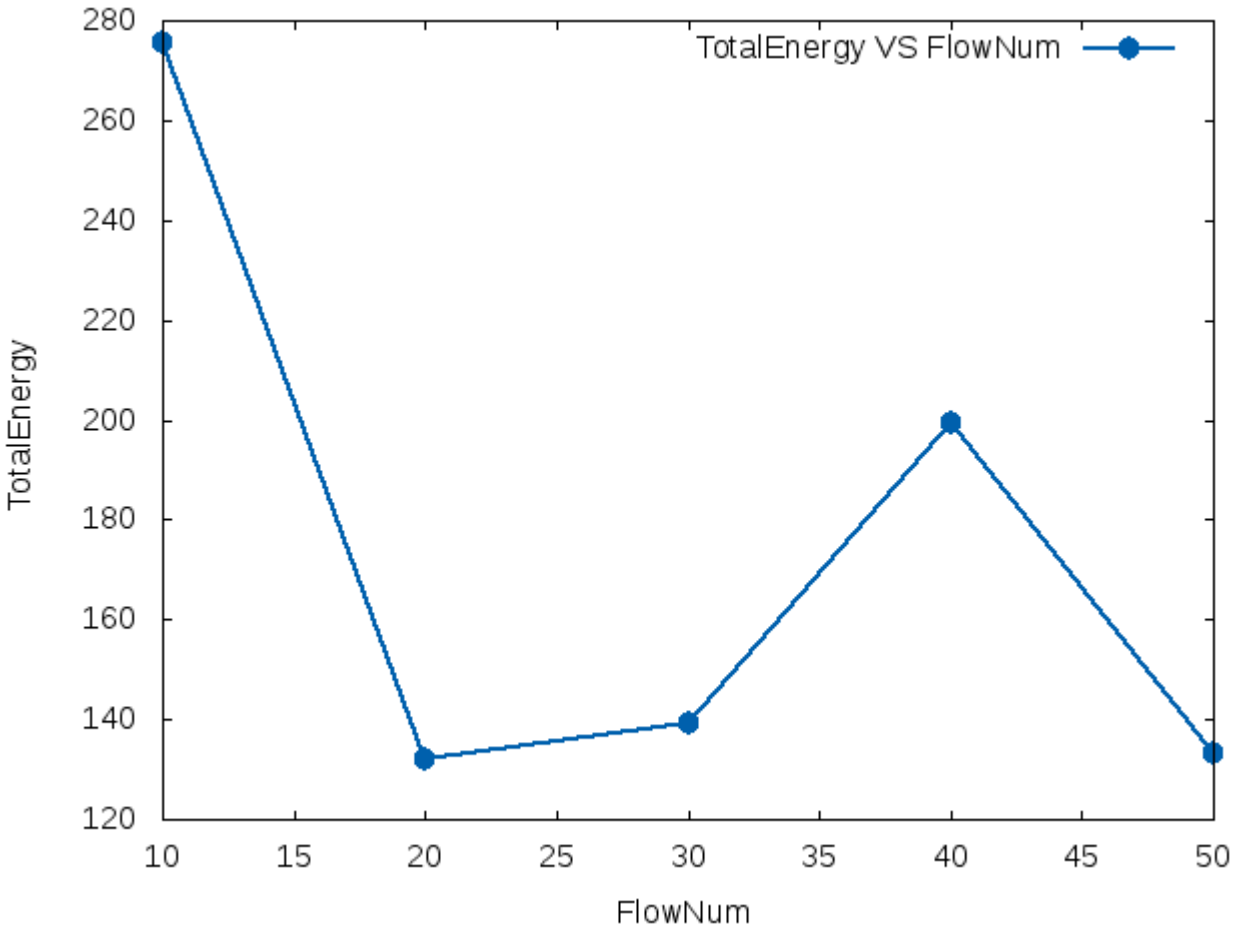

metrics vs no of speed of nodes
packet drop ratio
24
22
20
18
16
14
12
10
8
6
4
2
5
10
15
20
25
num of speed


metrics vs no of speed of nodes
throughput
50000
45000
40000
35000
30000
25000
20000
15000
10000
5000
5
10
15
20
25
num of speed

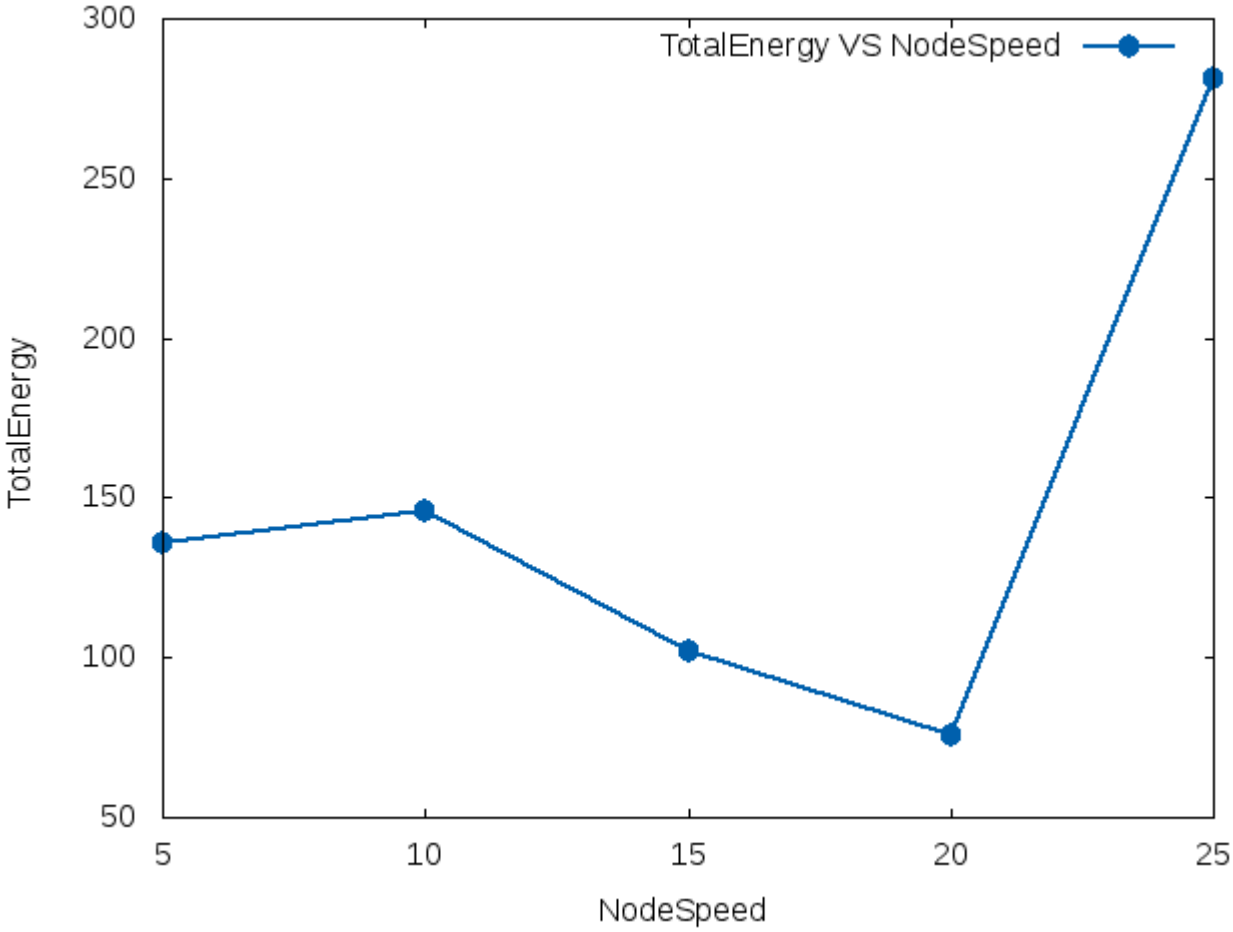
VS NodeNumMOD.png



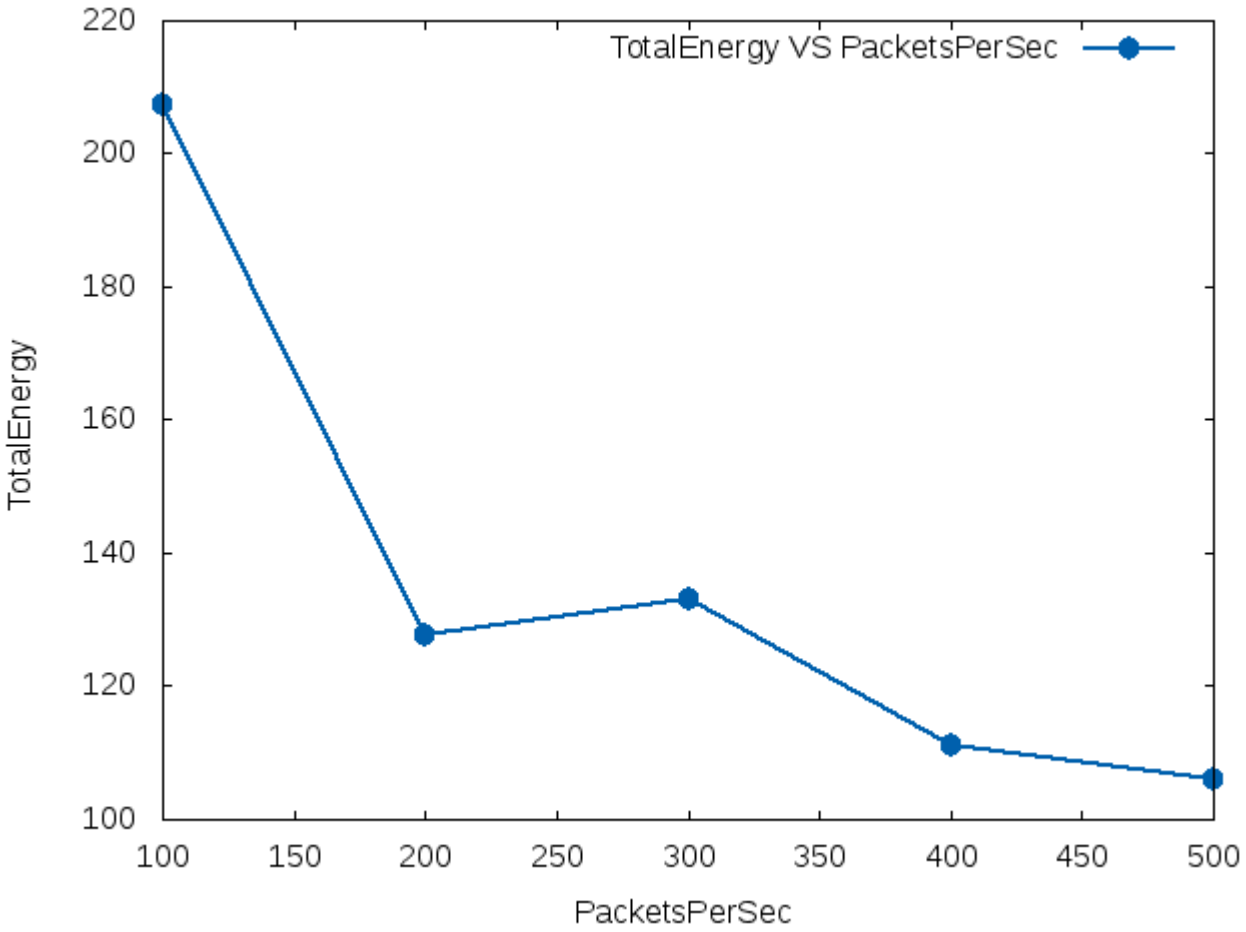
VS FlowNumMOD.png



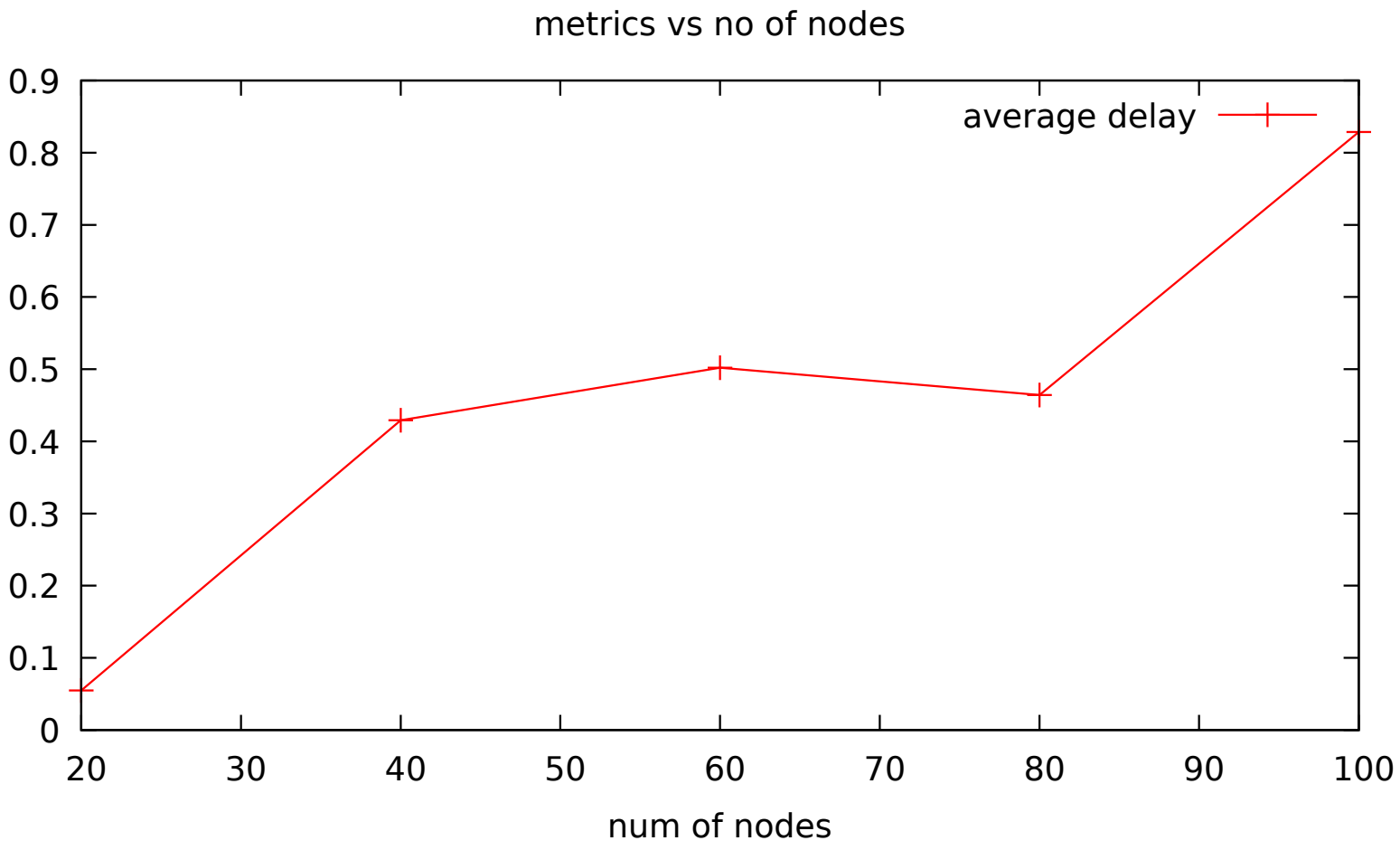
VS NodeSpeedMOD.png

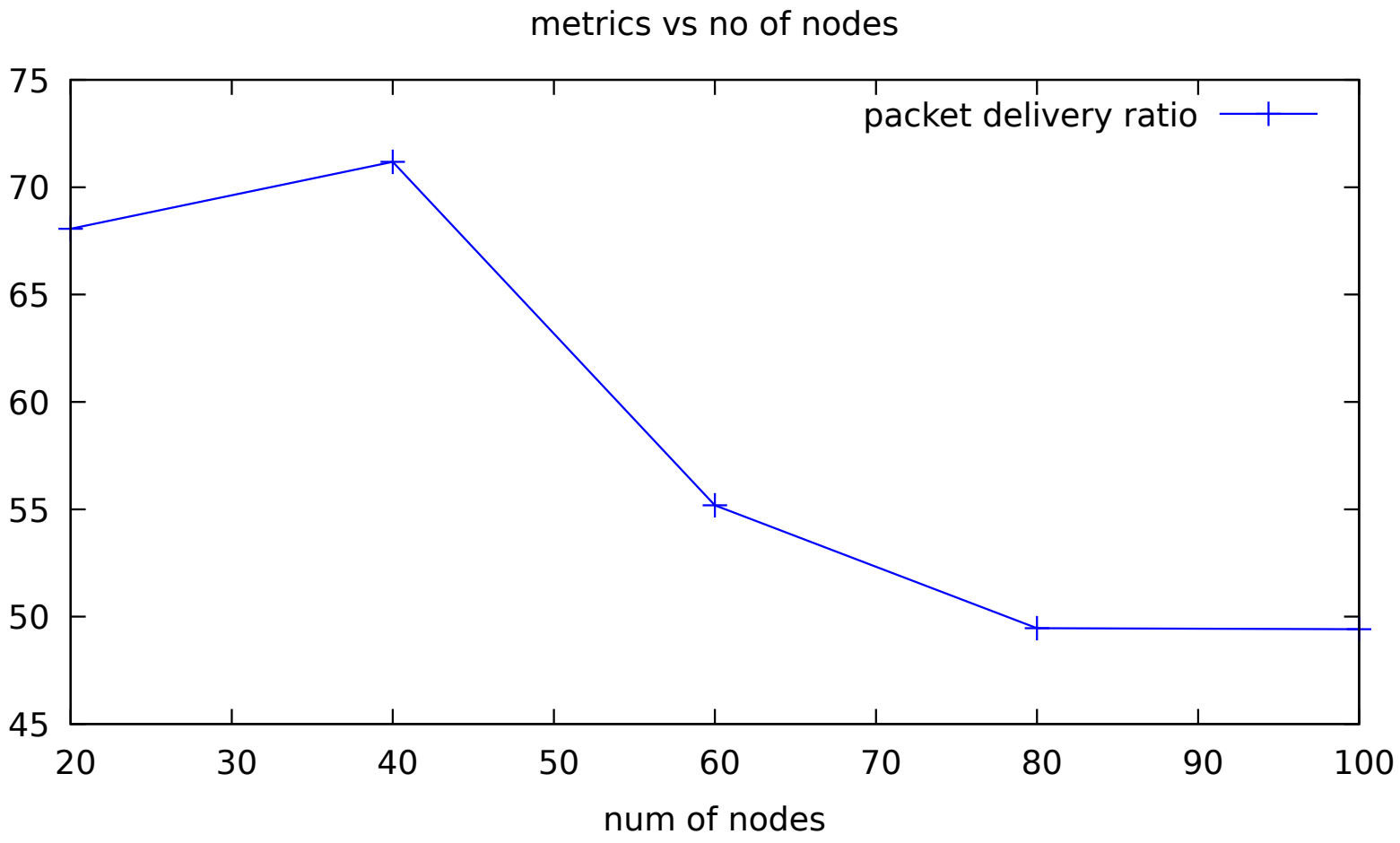


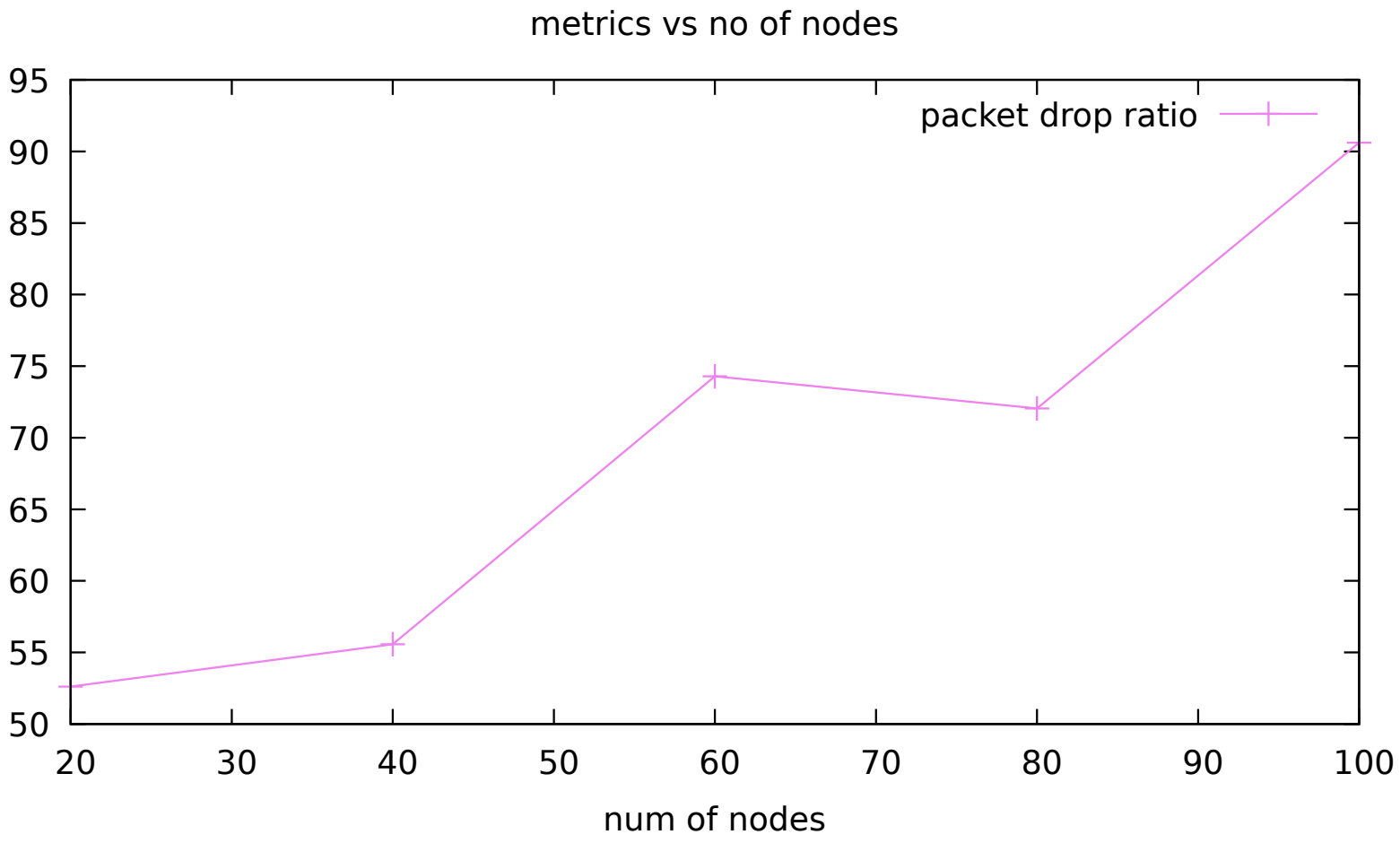
VS PacketsPerSecMOD.png

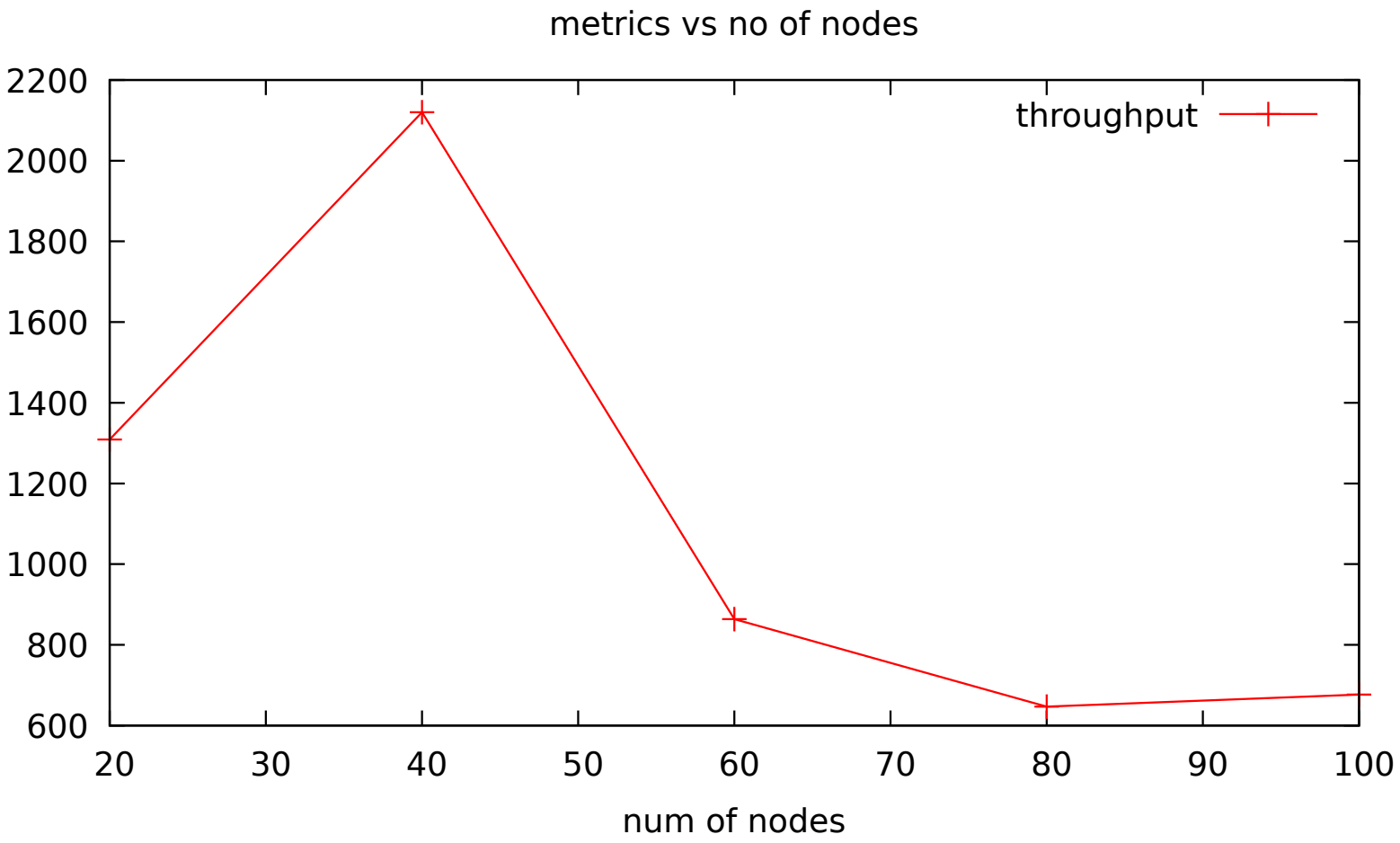

TotalEnergy VS PacketsPerSec
TotalEnergy
PacketsPerSec
220
200
180
160
140
120
100
100
150
200
250
300
350
400
450
500

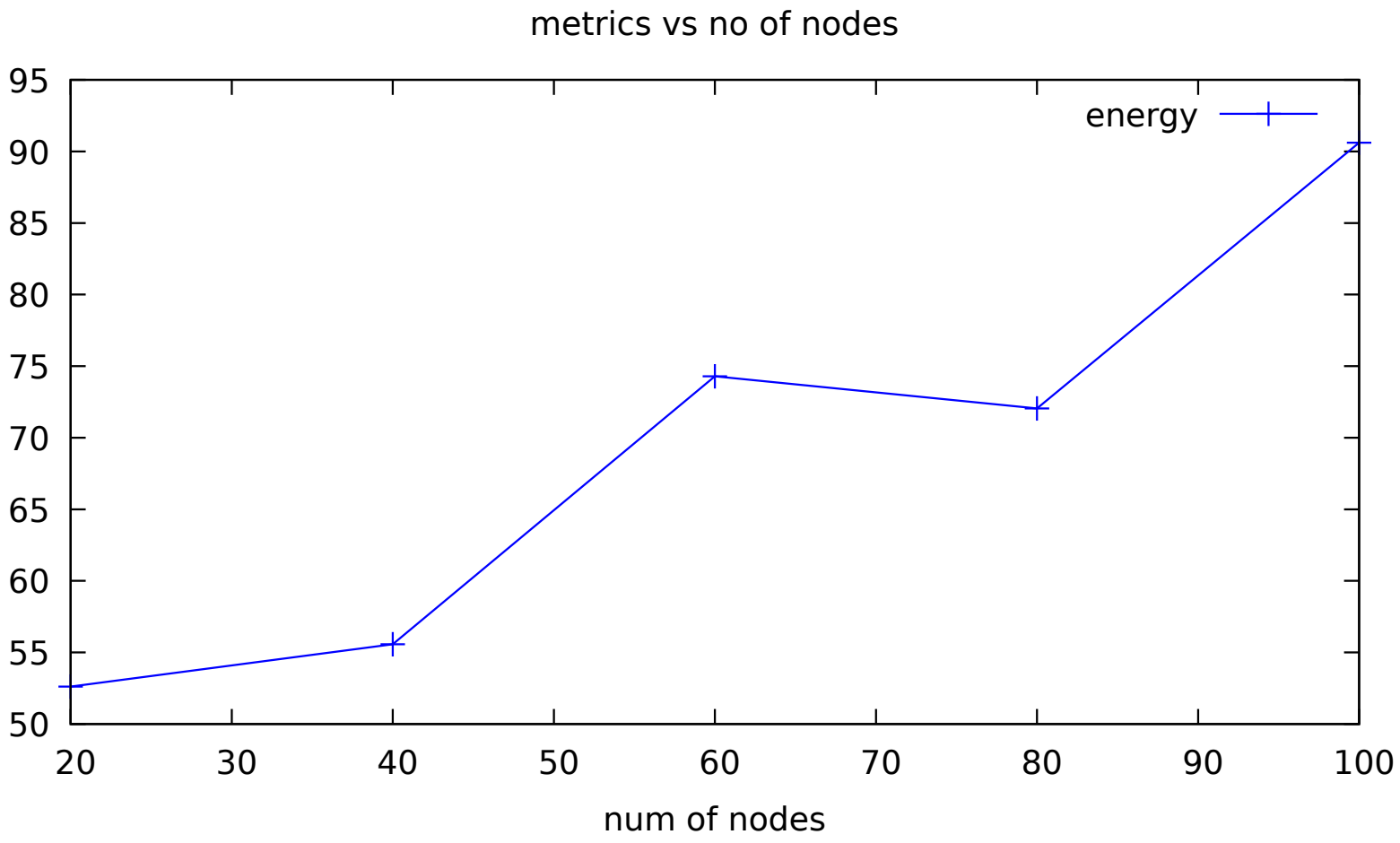
6.3 802.15.04 Static Original

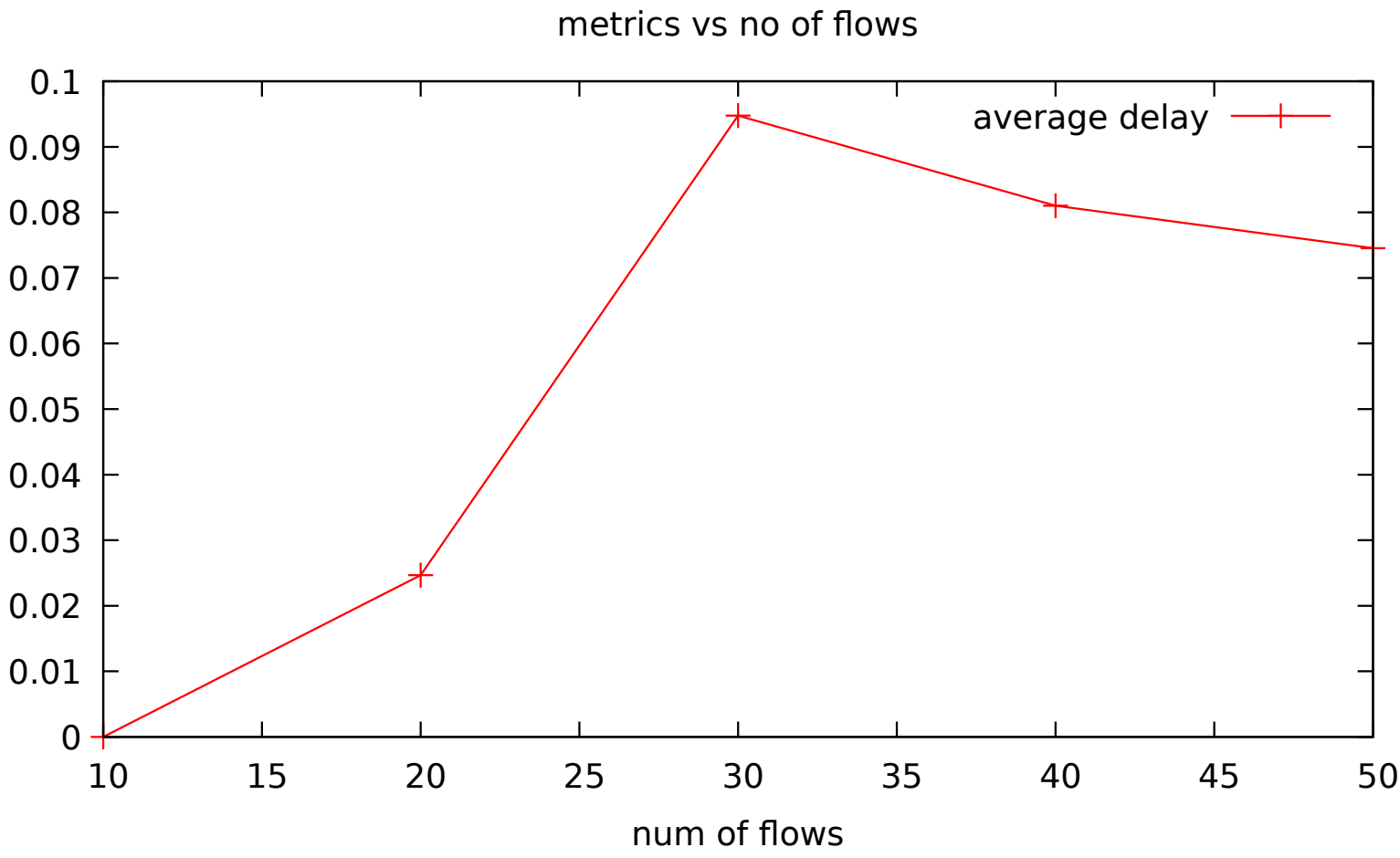

metrics vs no of nodes
average delay
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
20
30
40
50
60
70
80
90
100
num of nodes

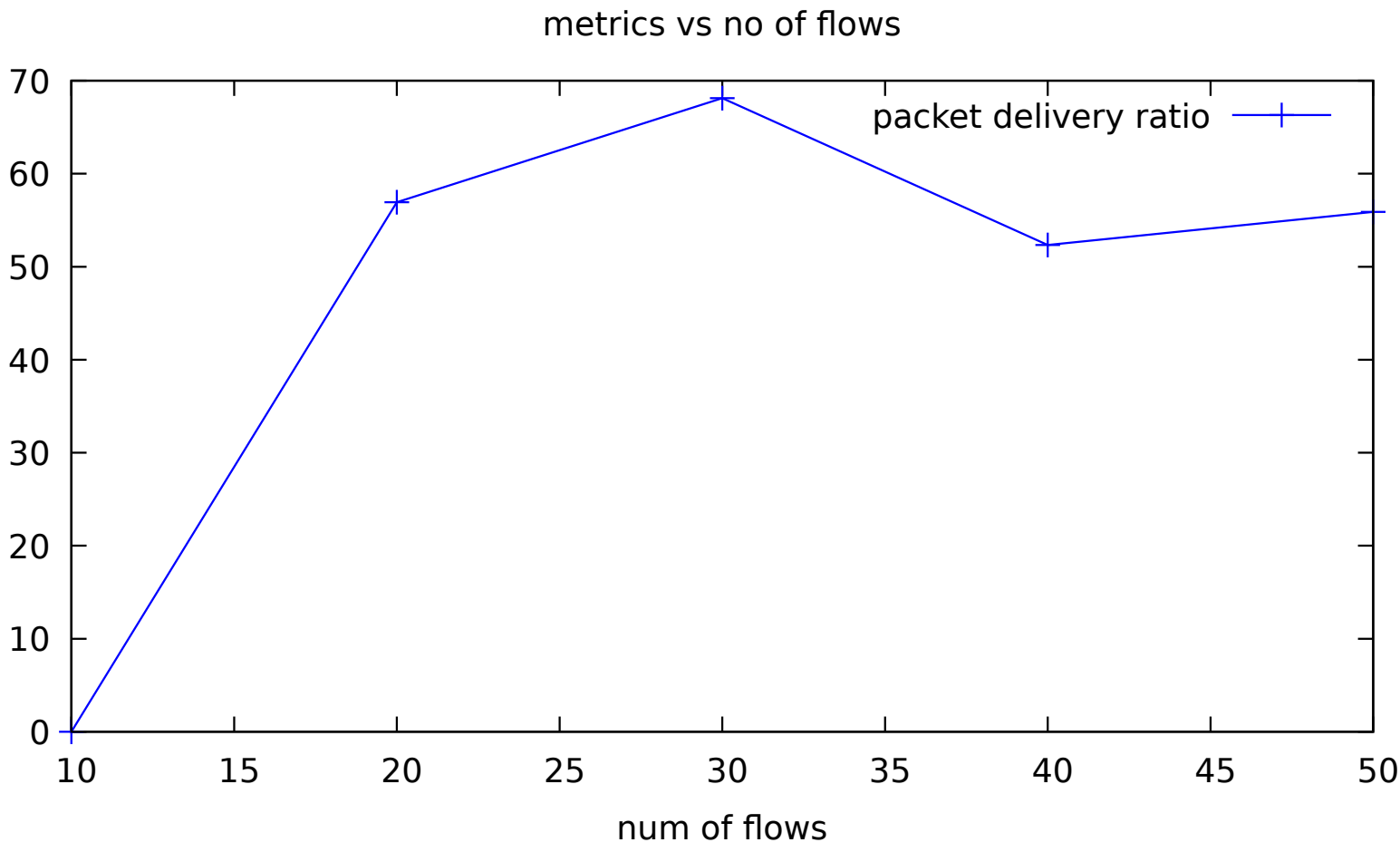

metrics vs no of nodes
packet delivery ratio
75
70
65
60
55
50
45
20
30
40
50
60
70
80
90
100
num of nodes

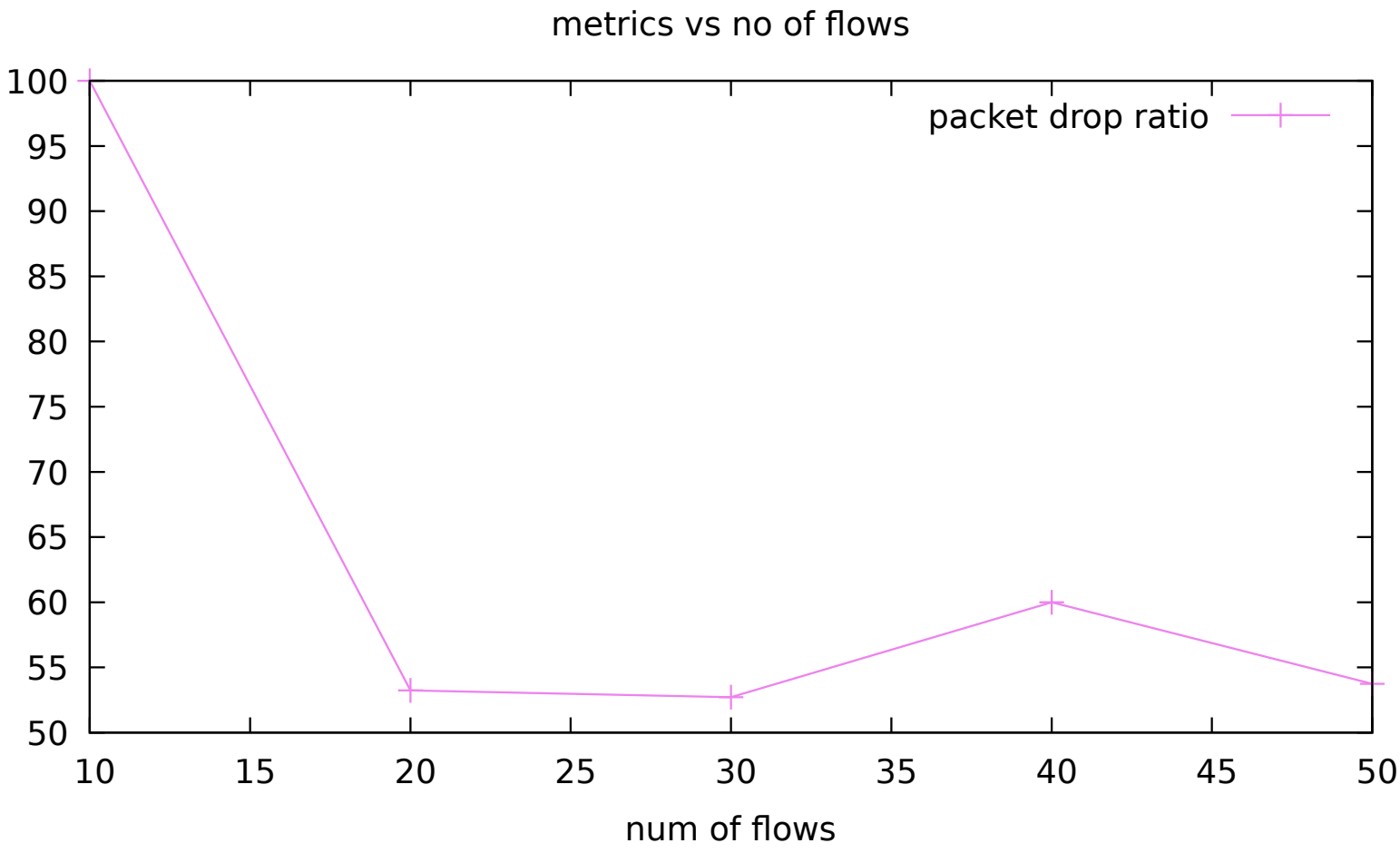

metrics vs no of nodes
packet drop ratio
95
90
85
80
75
70
65
60
55
50
20
30
40
50
60
70
80
90
100
num of nodes

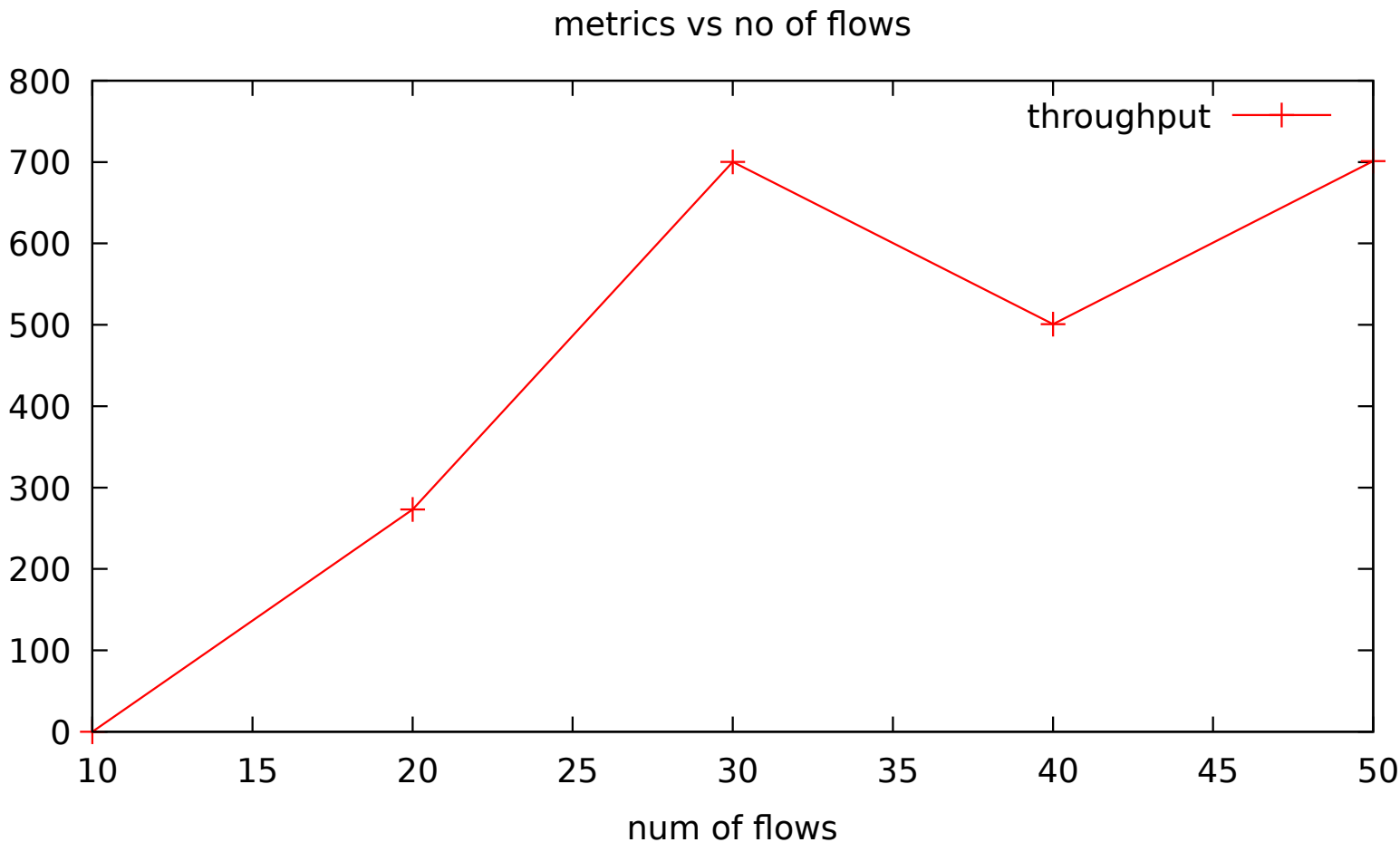

metrics vs no of nodes
throughput
2200
2000
1800
1600
1400
1200
1000
800
600
20
30
40
50
60
70
80
90
100
num of nodes

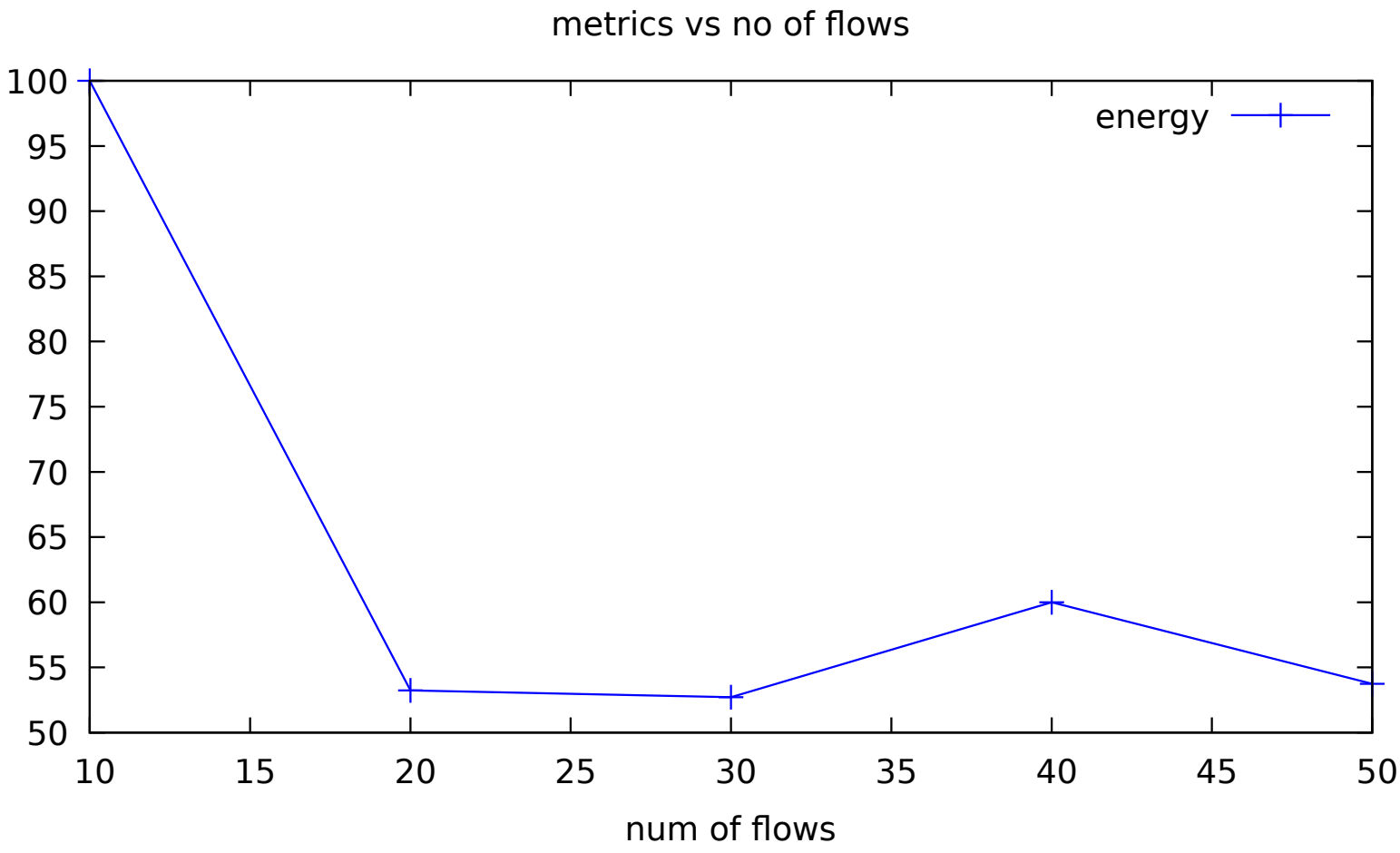

metrics vs no of nodes
energy
95
90
85
80
75
70
65
60
55
50
20
30
40
50
60
70
80
90
100
num of nodes

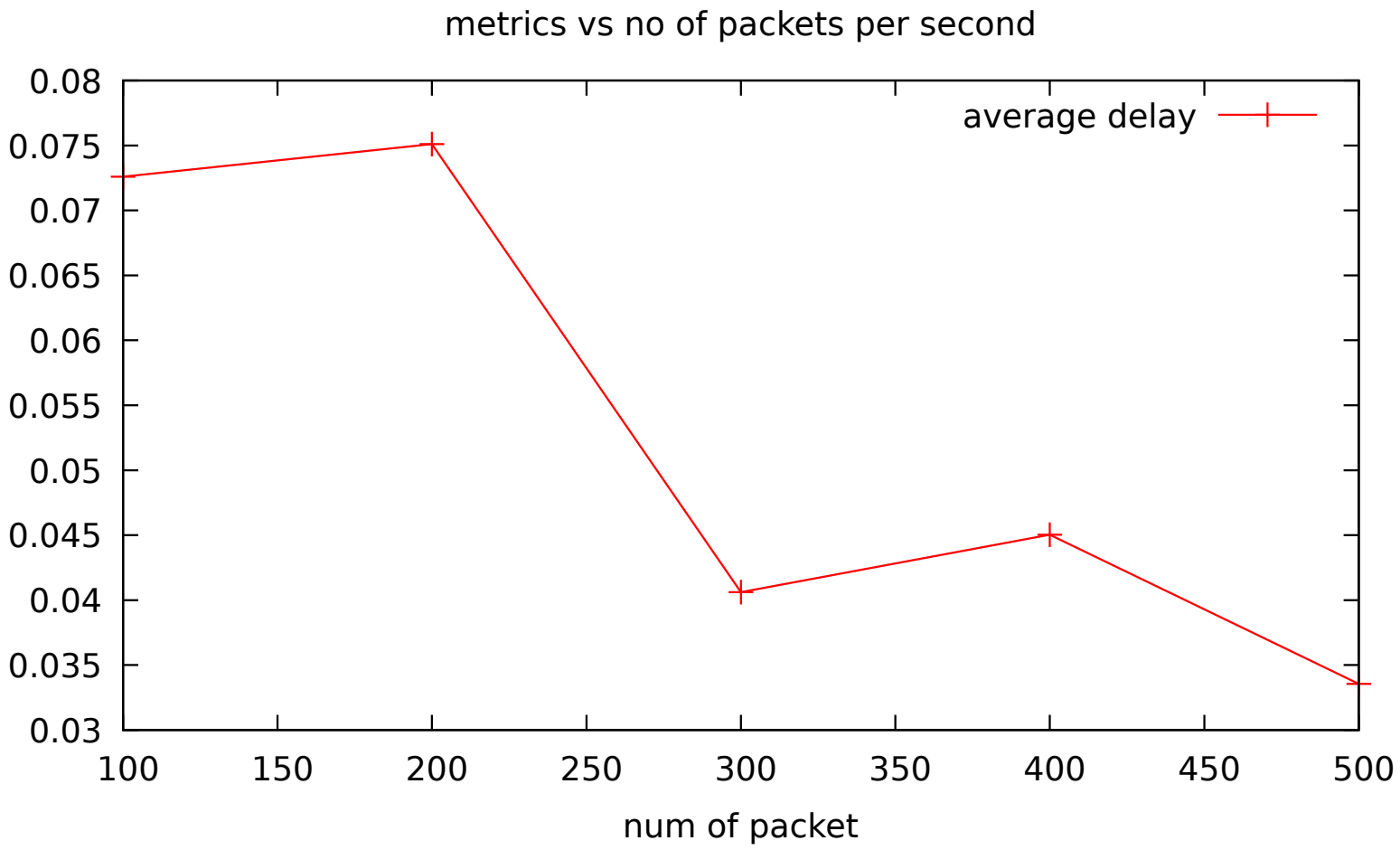

metrics vs no of flows
average delay
0.1
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
10
15
20
25
30
35
40
45
50
num of flows

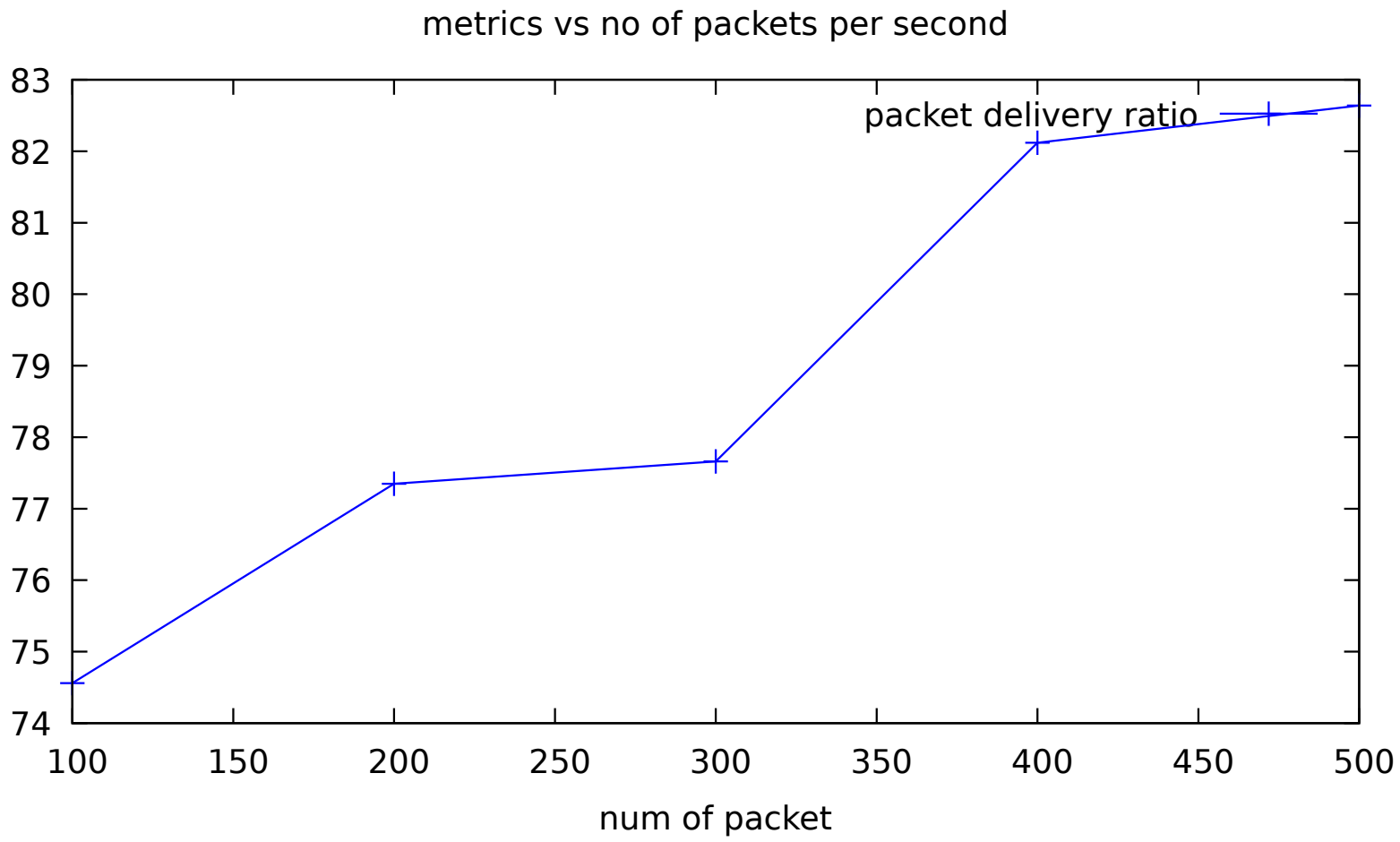

metrics vs no of flows
packet delivery ratio
70
60
50
40
30
20
10
0
10
15
20
25
30
35
40
45
50
num of flows

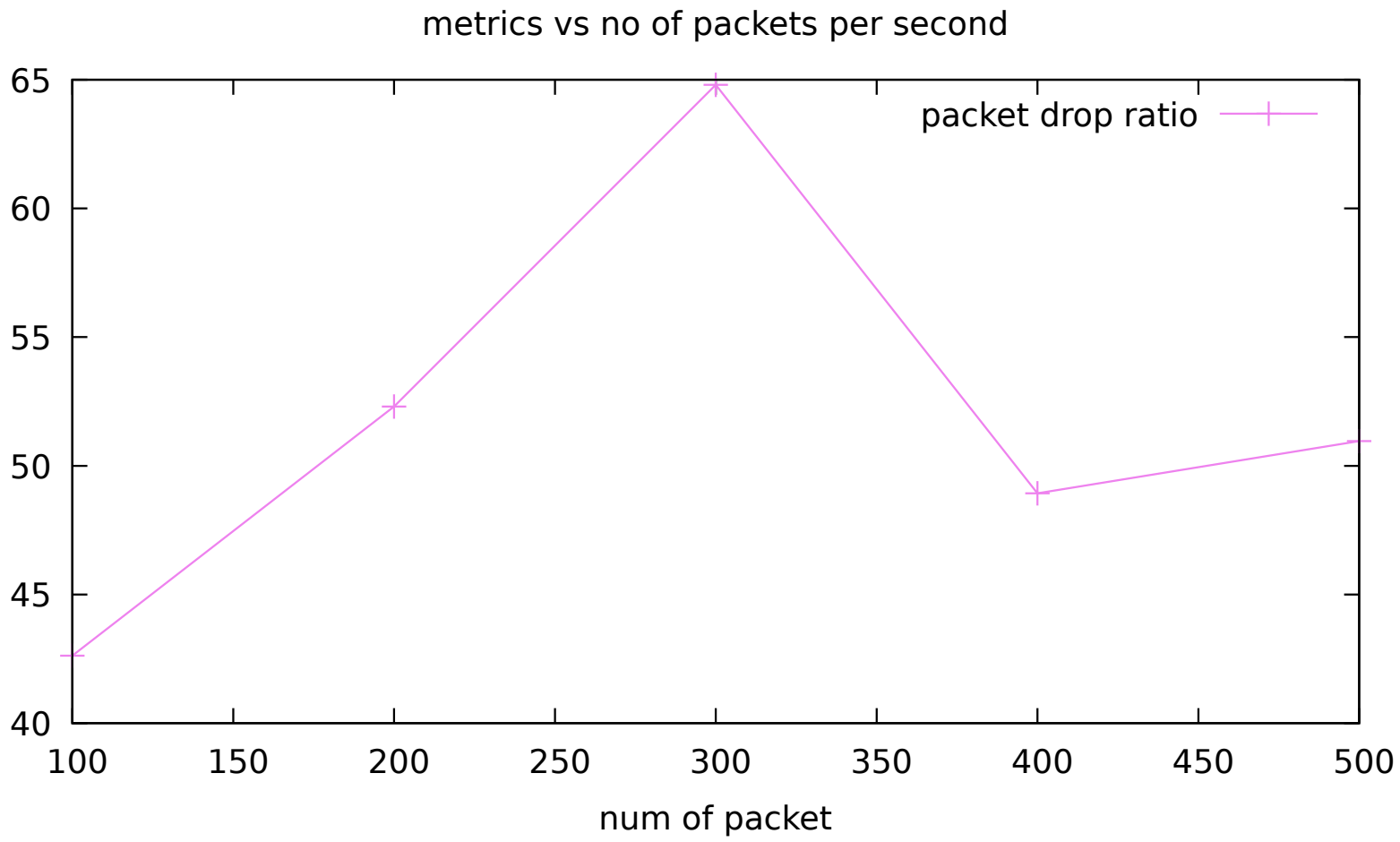

metrics vs no of flows
packet drop ratio
100
95
90
85
80
75
70
65
60
55
50
10
15
20
25
30
35
40
45
50
num of flows

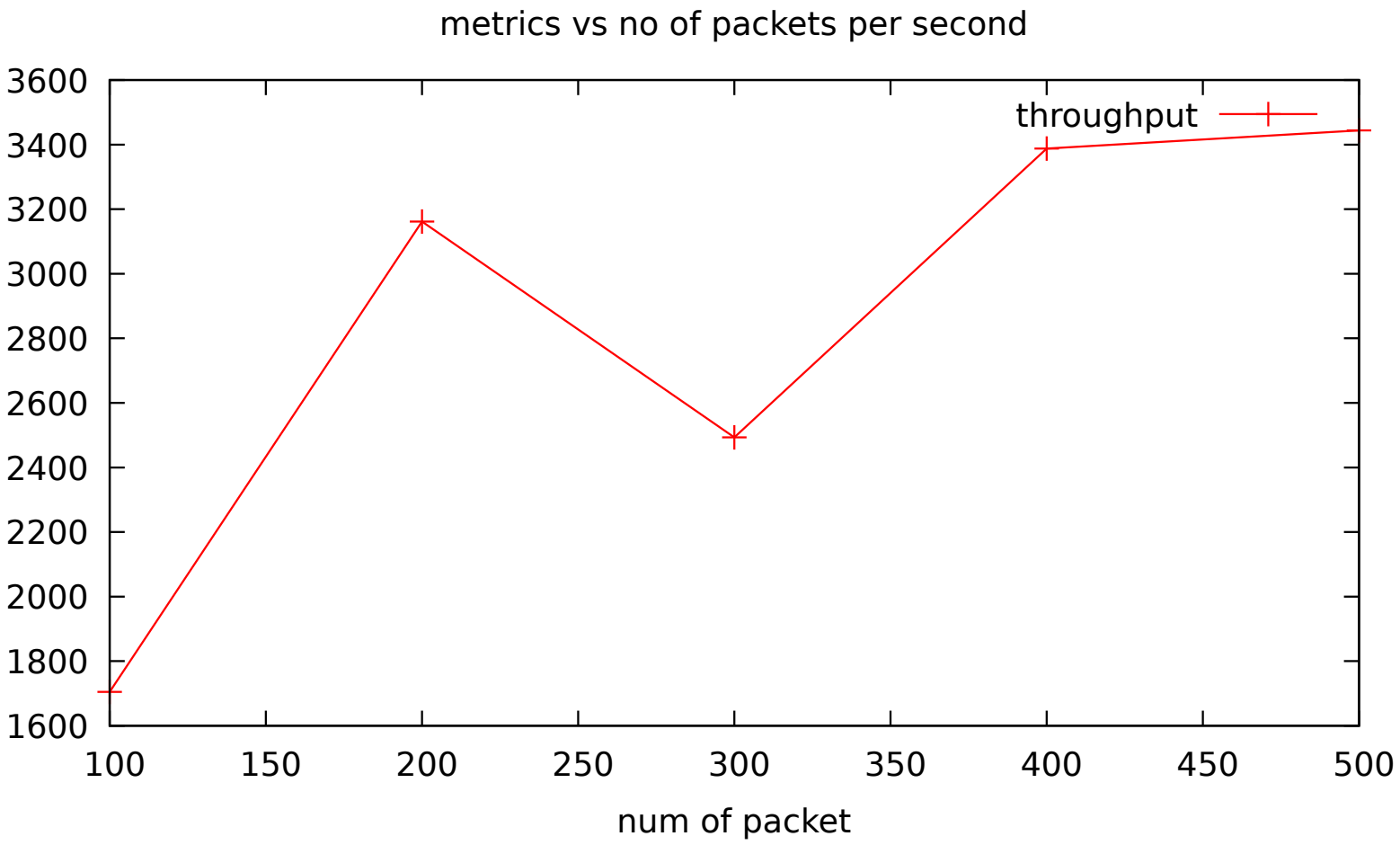

metrics vs no of flows
800
700
600
500
400
300
200
100
0
10
15
20
25
30
35
40
45
50
num of flows
throughput

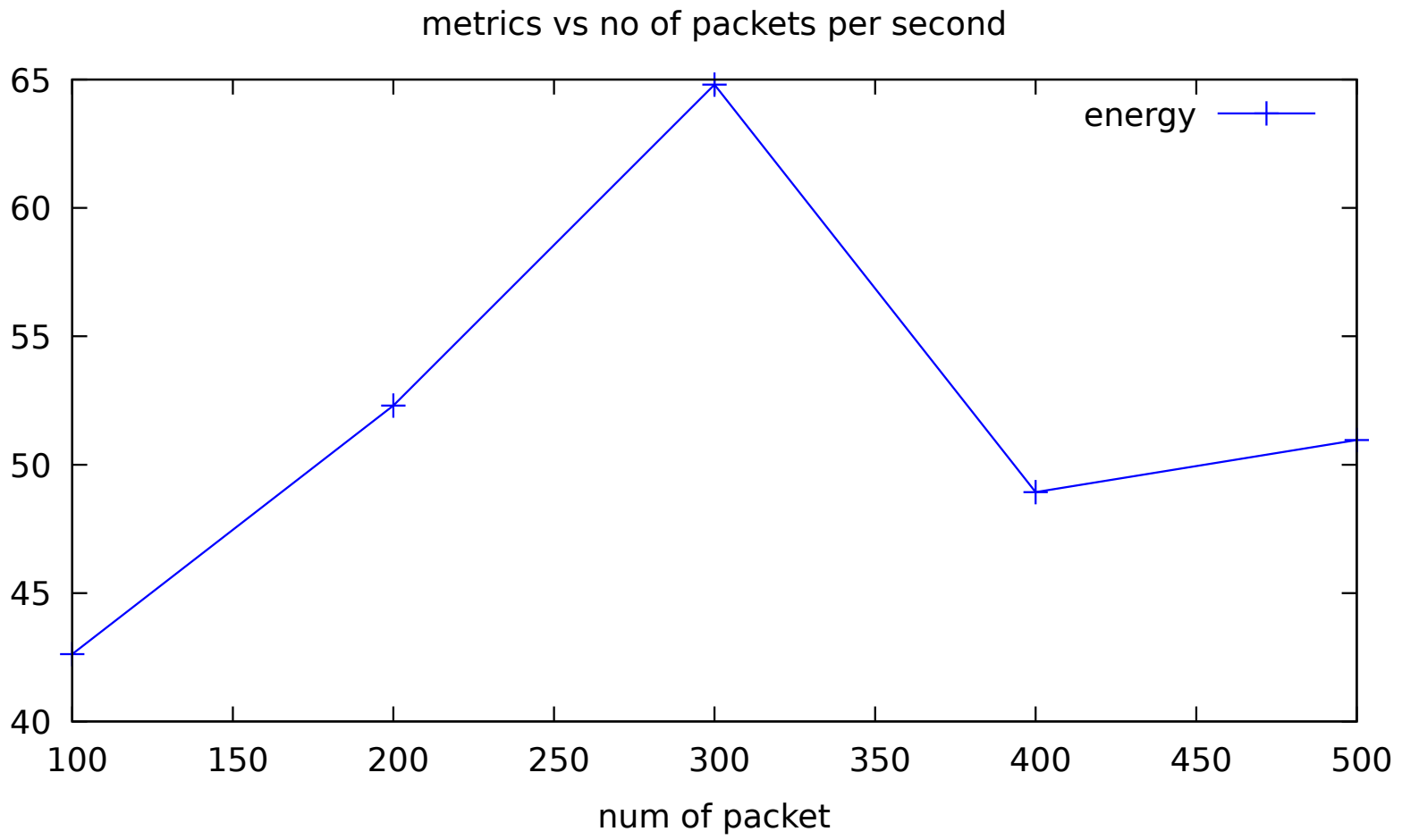

metrics vs no of flows
energy
100
95
90
85
80
75
70
65
60
55
50
10
15
20
25
30
35
40
45
50
num of flows

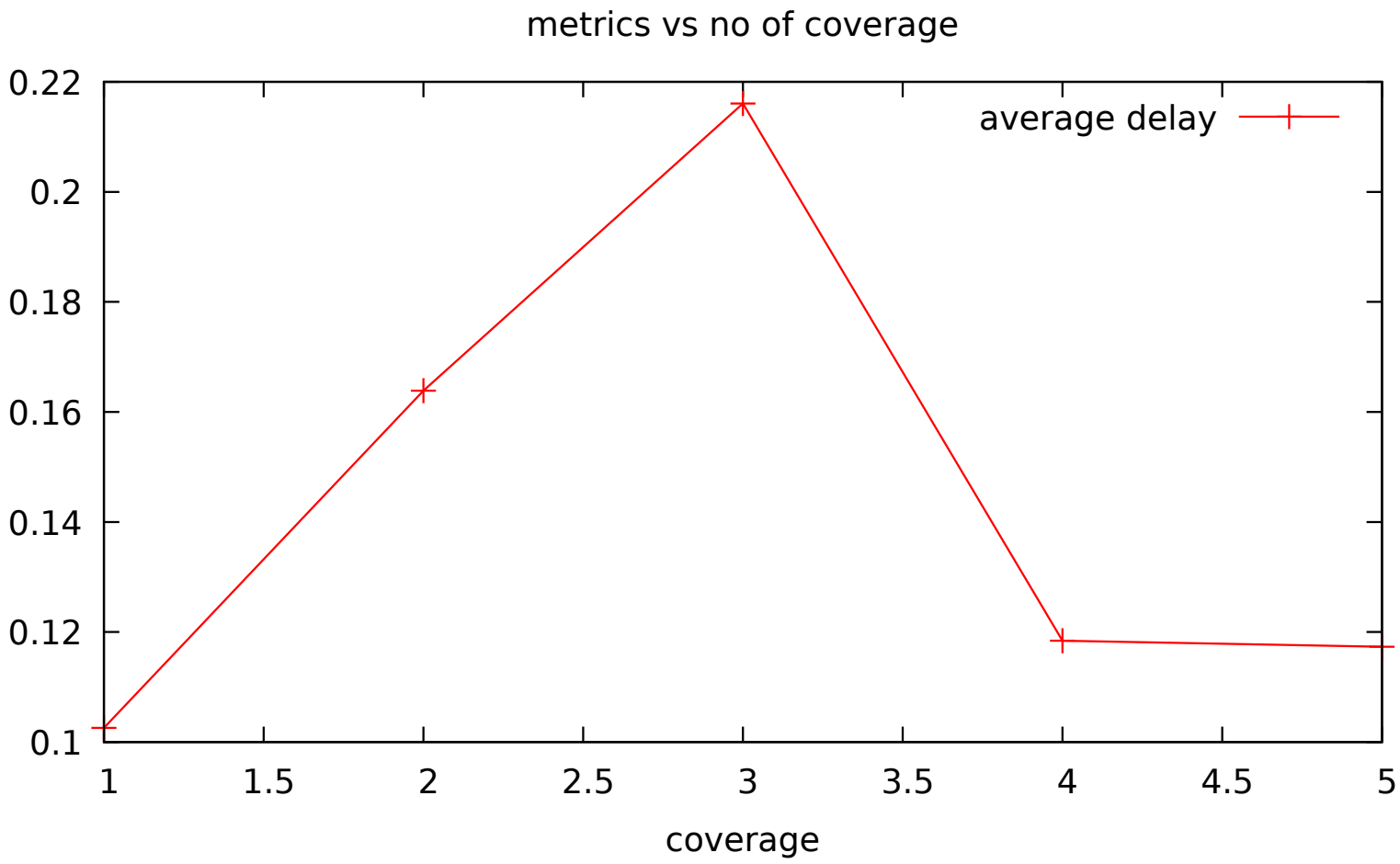

metrics vs no of packets per second
average delay
0.08
0.075
0.07
0.065
0.06
0.055
0.05
0.045
0.04
0.035
0.03
100
150
200
250
300
350
400
450
500
num of packet

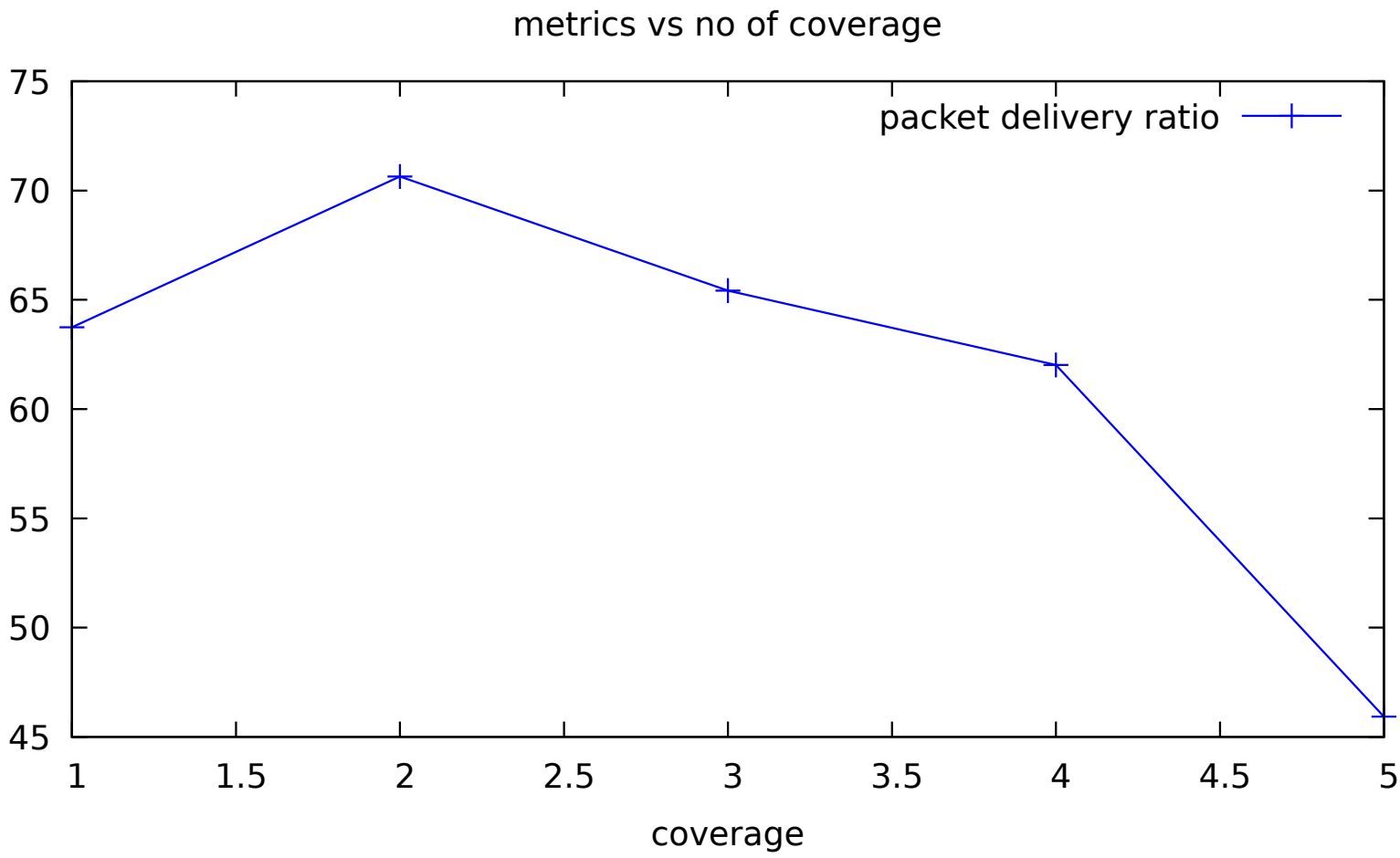

metrics vs no of packets per second
packet delivery ratio
83
82
81
80
79
78
77
76
75
74
100
150
200
250
300
350
400
450
500
num of packet

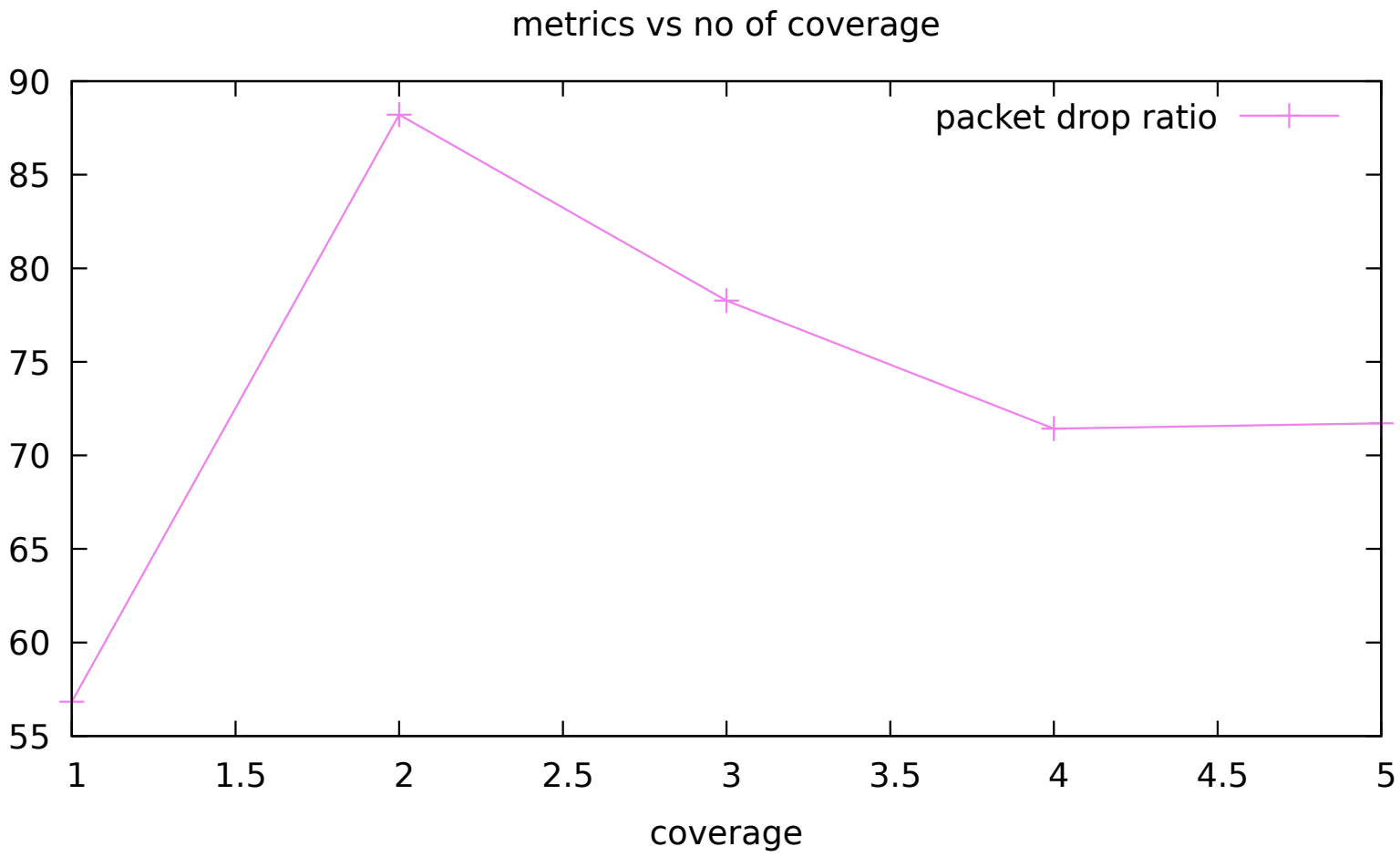

metrics vs no of packets per second
packet drop ratio
65
60
55
50
45
40
100
150
200
250
300
350
400
450
500
num of packet

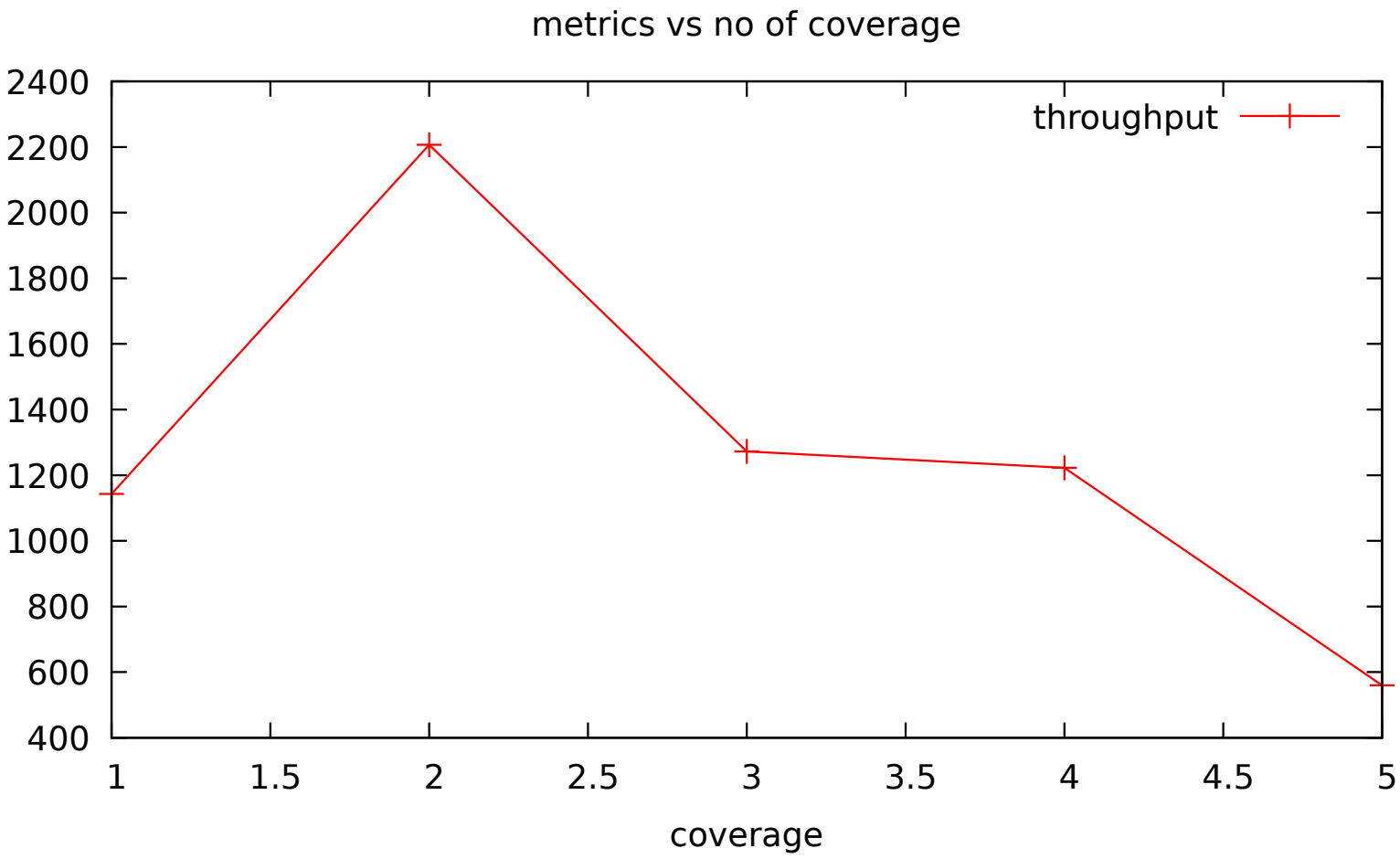

metrics vs no of packets per second
throughput
3600
3400
3200
3000
2800
2600
2400
2200
2000
1800
1600
100
150
200
250
300
350
400
450
500
num of packet

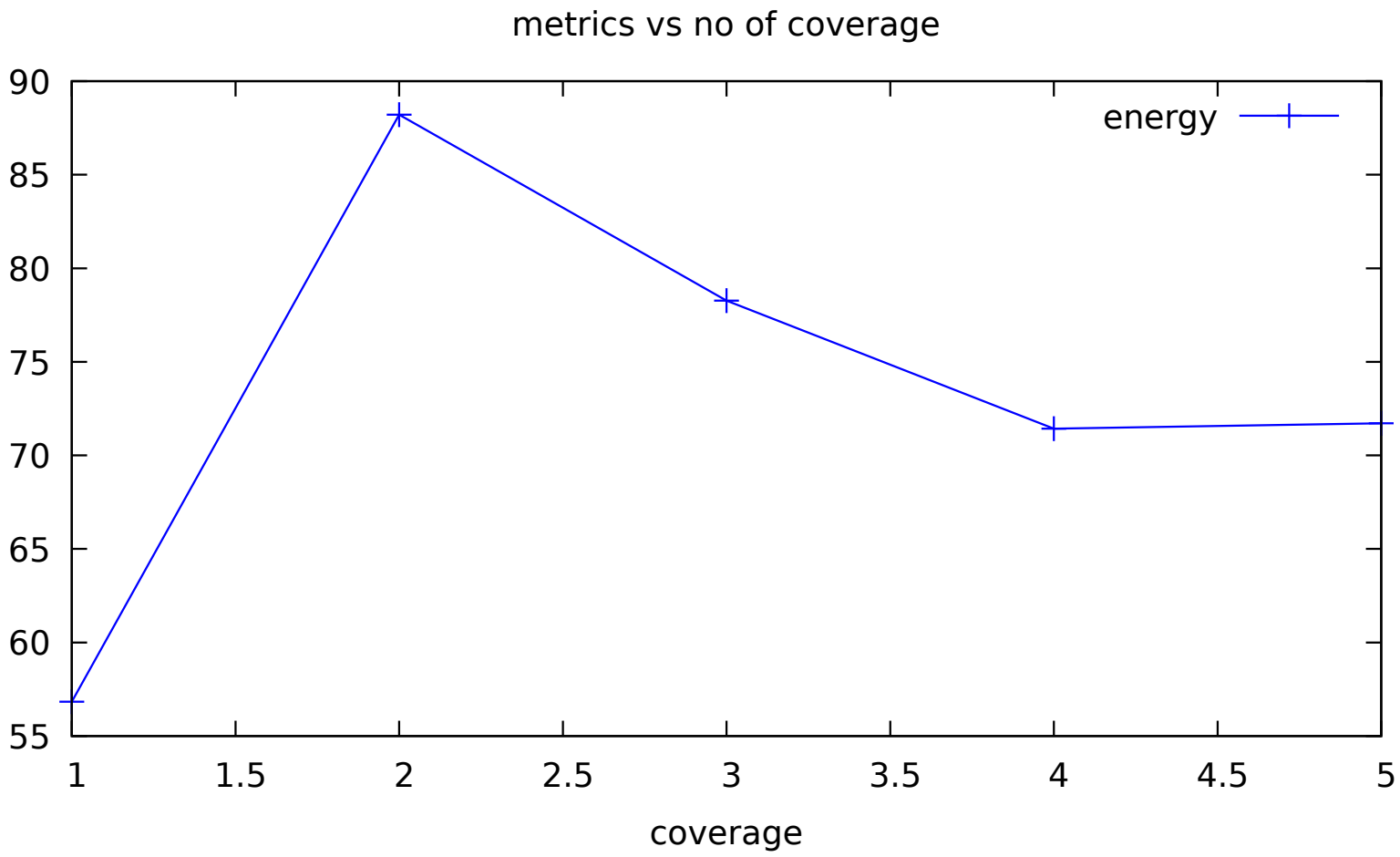

metrics vs no of packets per second
energy
65
60
55
50
45
40
100
150
200
250
300
350
400
450
500
num of packet


metrics vs no of coverage
average delay
0.22
0.2
0.18
0.16
0.14
0.12
0.1
1
1.5
2
2.5
3
3.5
4
4.5
5
coverage


metrics vs no of coverage
packet delivery ratio
75
70
65
60
55
50
45
1
1.5
2
2.5
3
3.5
4
4.5
5
coverage

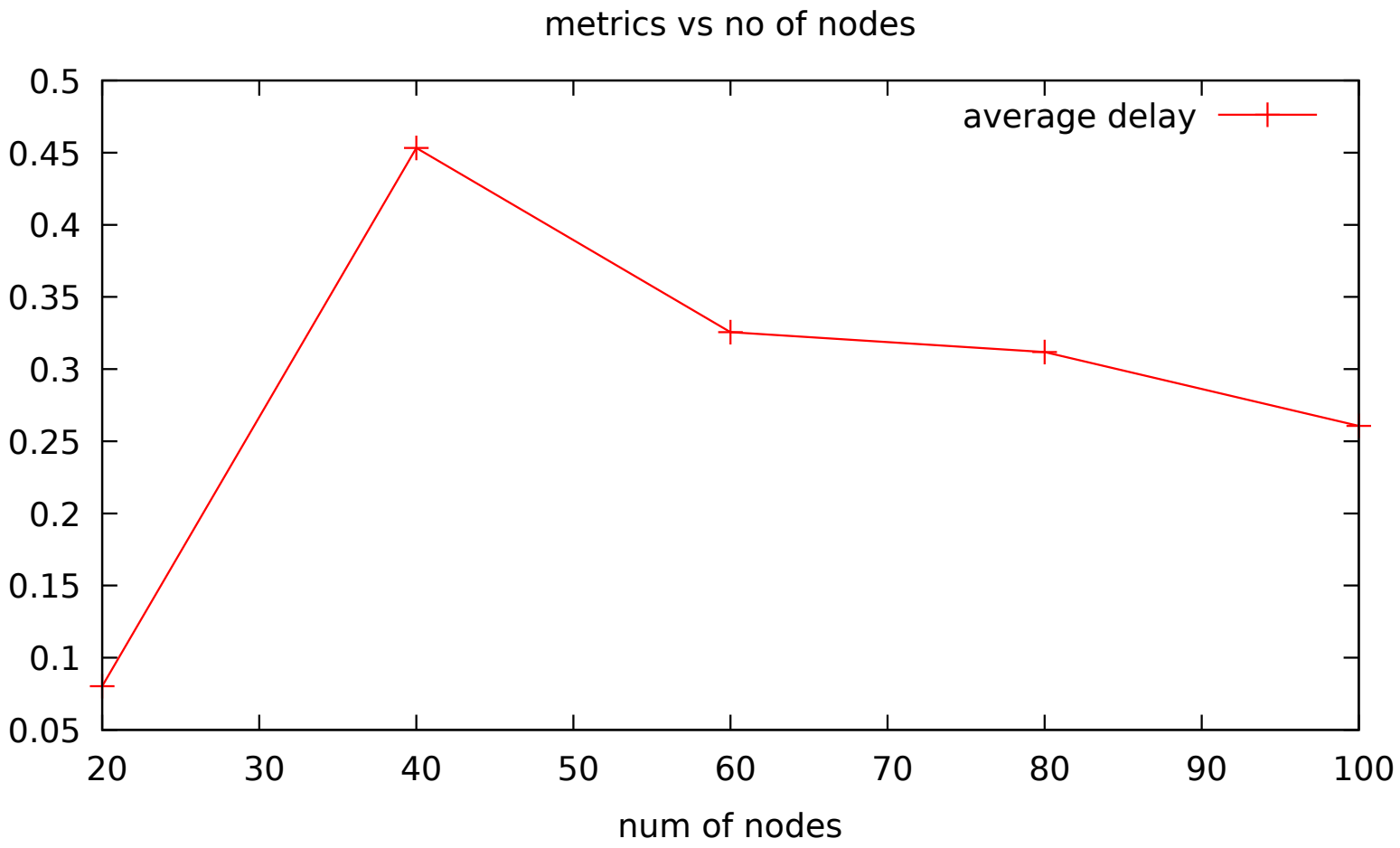

metrics vs no of coverage
packet drop ratio
90
85
80
75
70
65
60
55
1
1.5
2
2.5
3
3.5
4
4.5
5
coverage

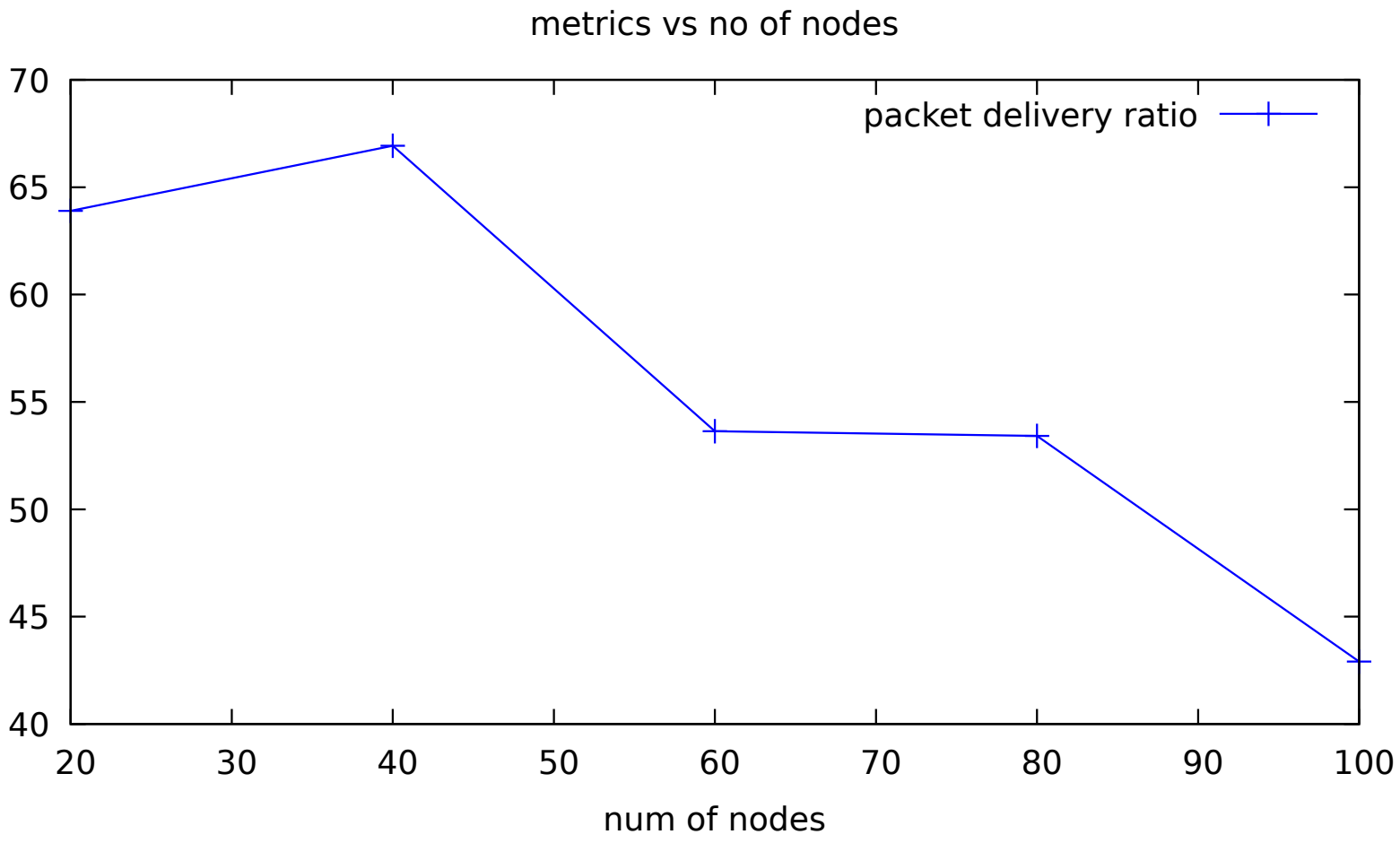

metrics vs no of coverage
throughput
2400
2200
2000
1800
1600
1400
1200
1000
800
600
400
1
1.5
2
2.5
3
3.5
4
4.5
5
coverage

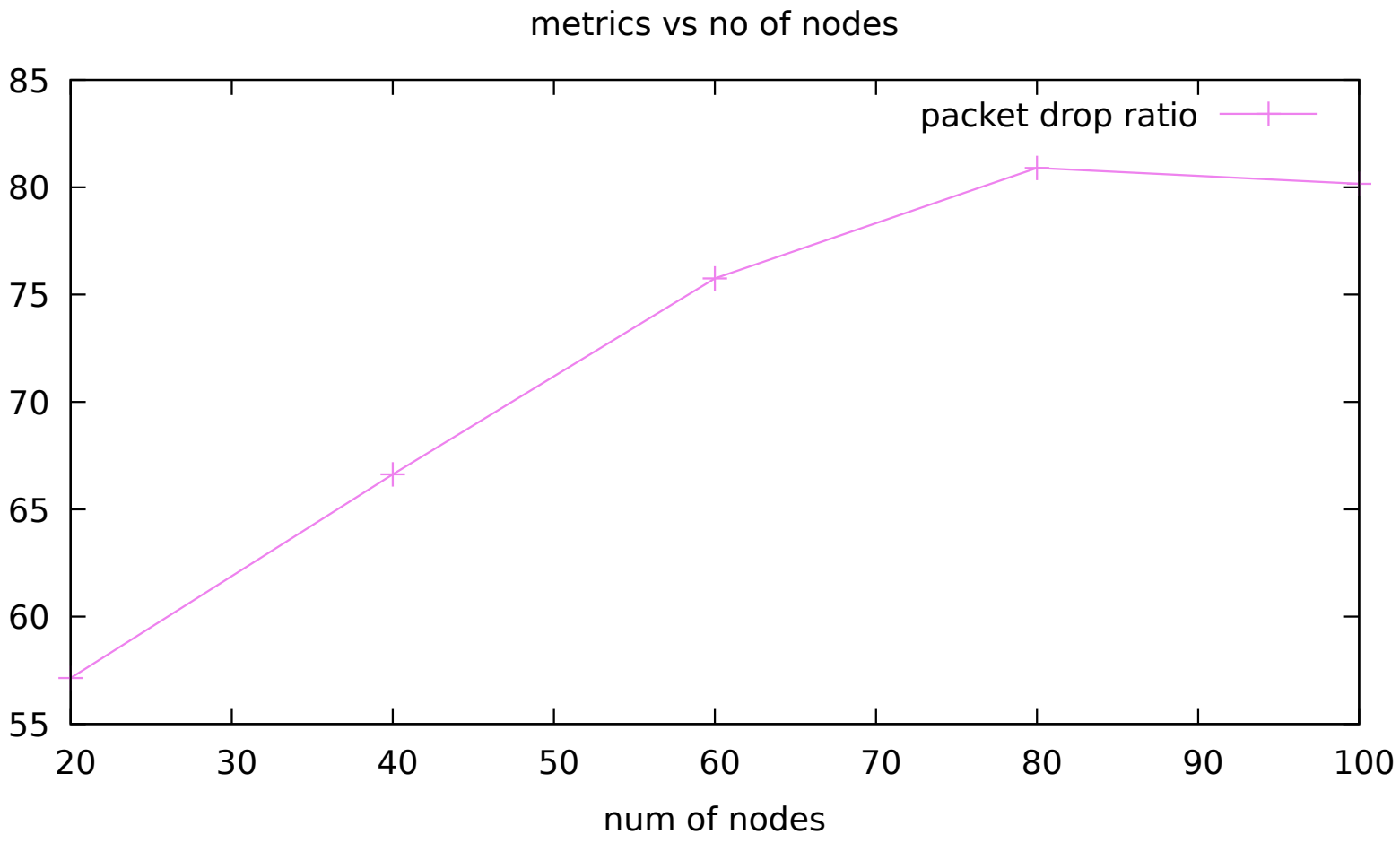

metrics vs no of coverage
energy
90
85
80
75
70
65
60
55
1
1.5
2
2.5
3
3.5
4
4.5
5
coverage

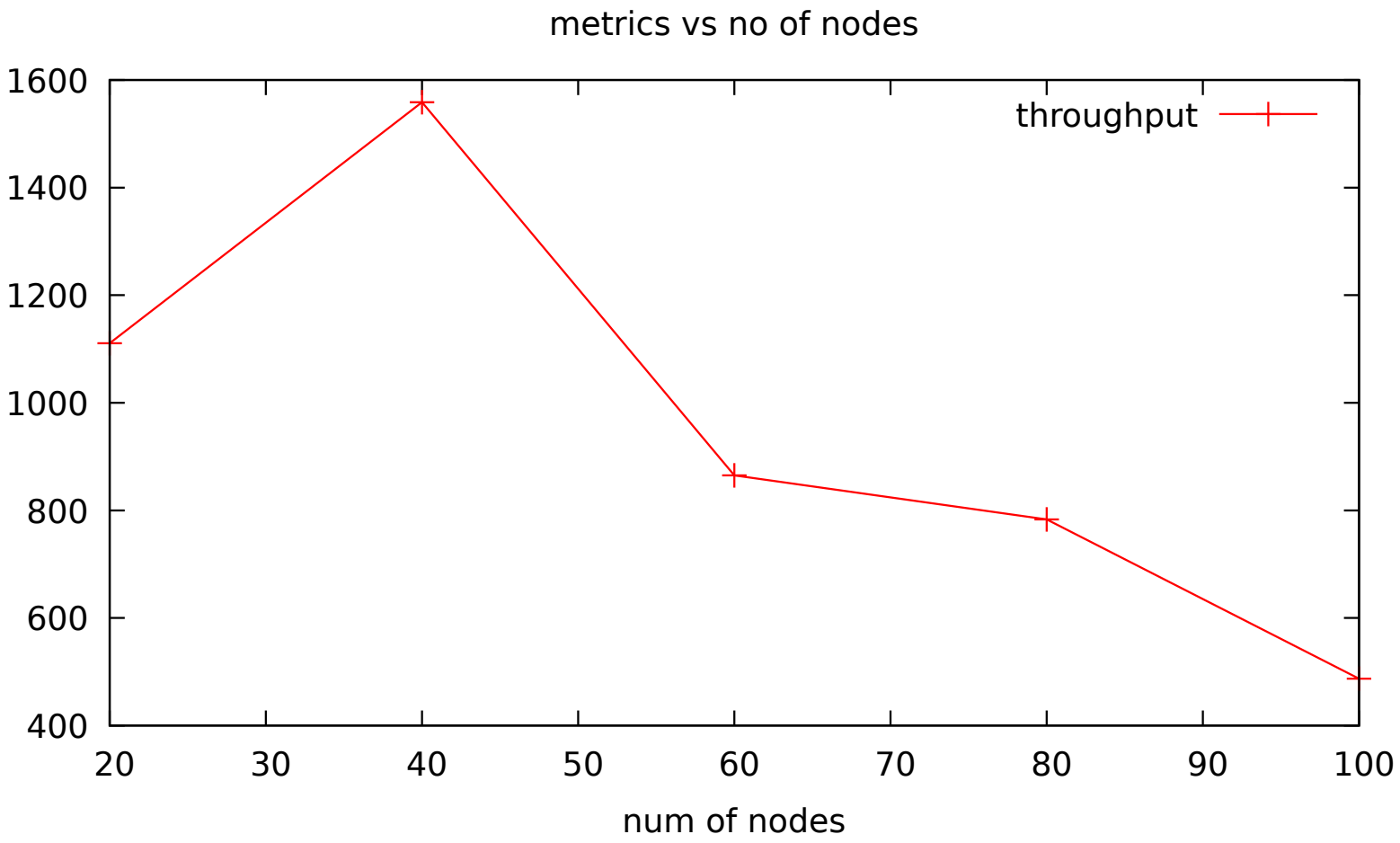
6.4 802.15.04 Static Modified

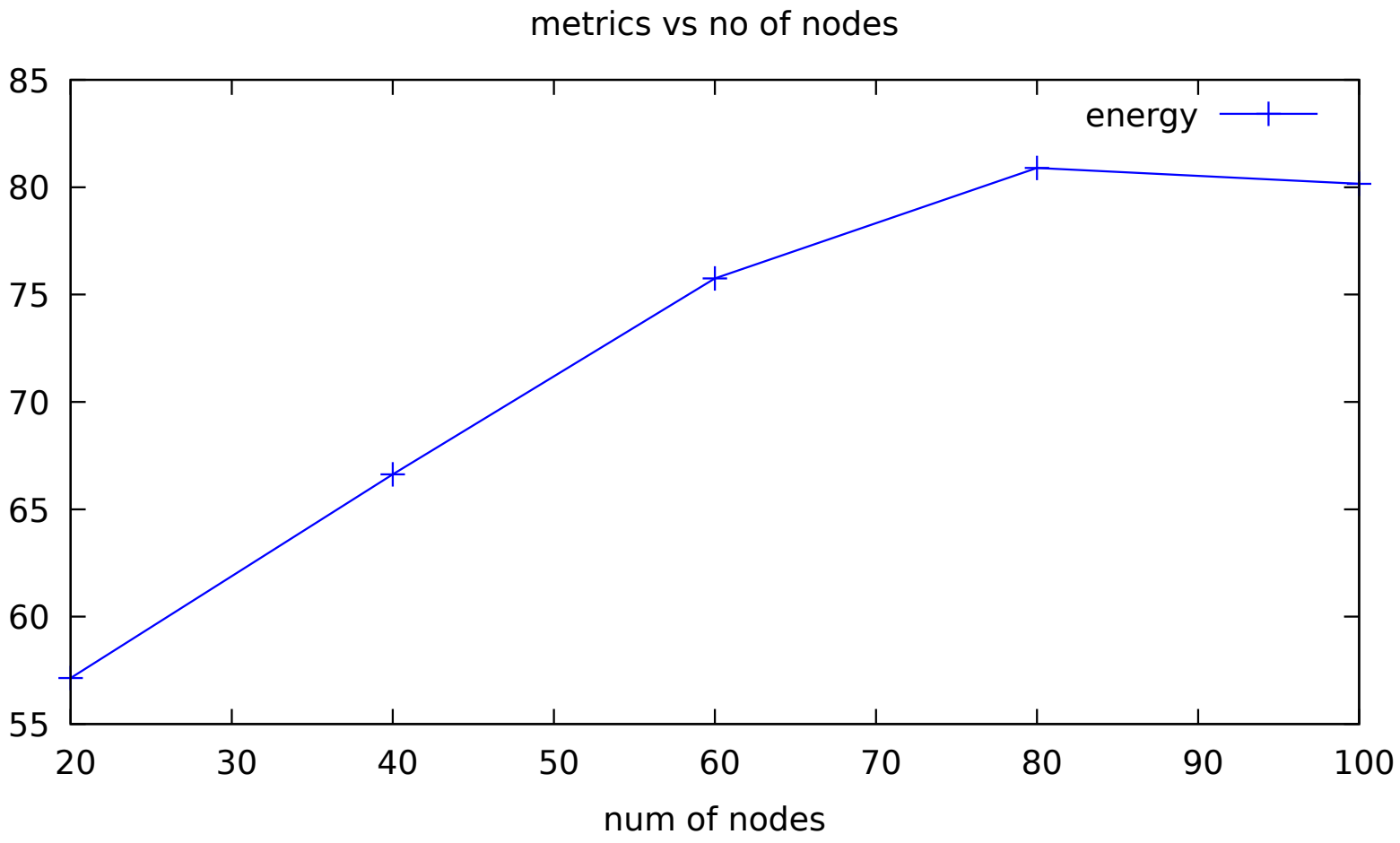
After modifying tcp and AODV protocol, the following graphs are generated.
Here AODV routing protocol is used.

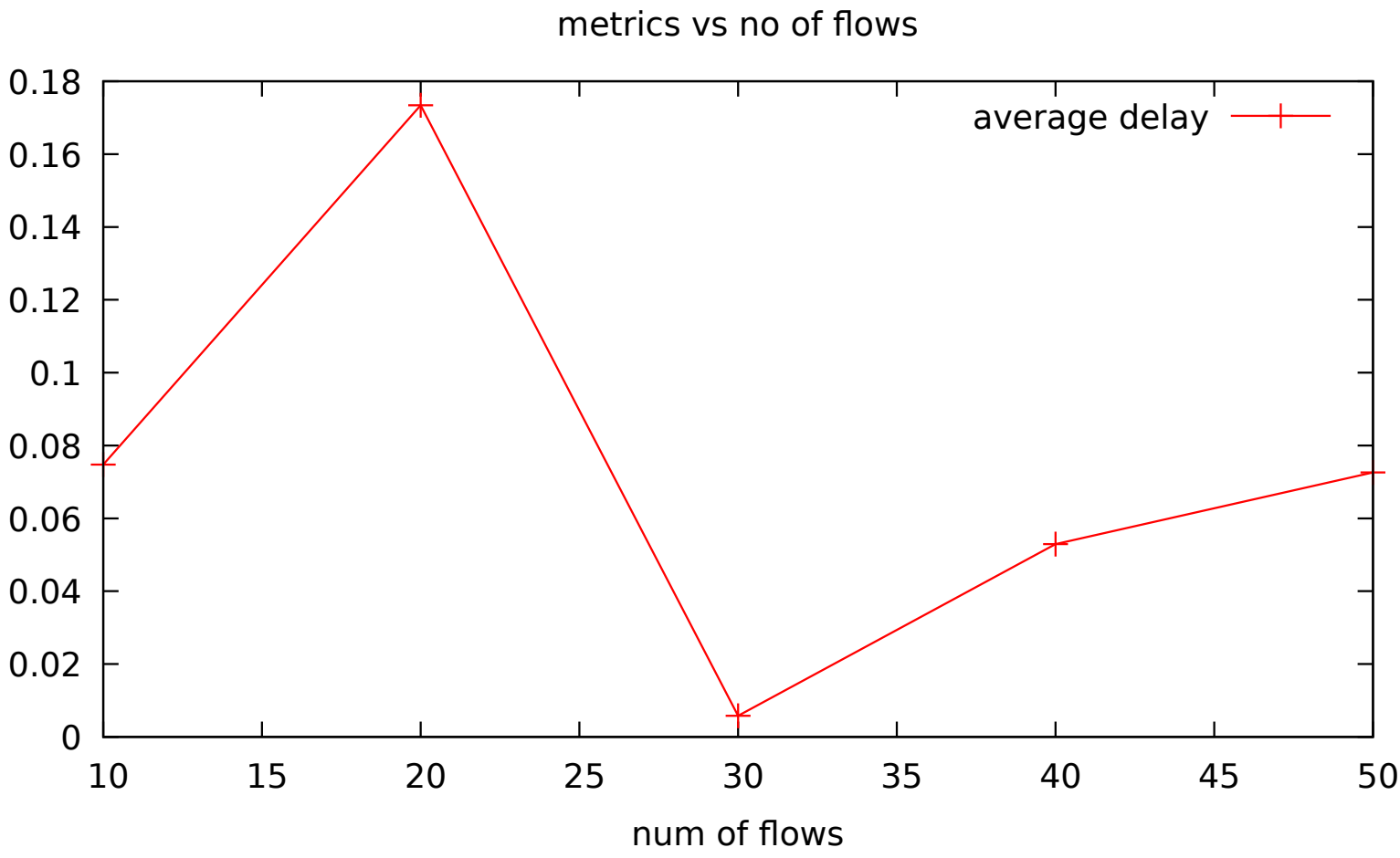

metrics vs no of nodes
average delay
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
20
30
40
50
60
70
80
90
100
num of nodes

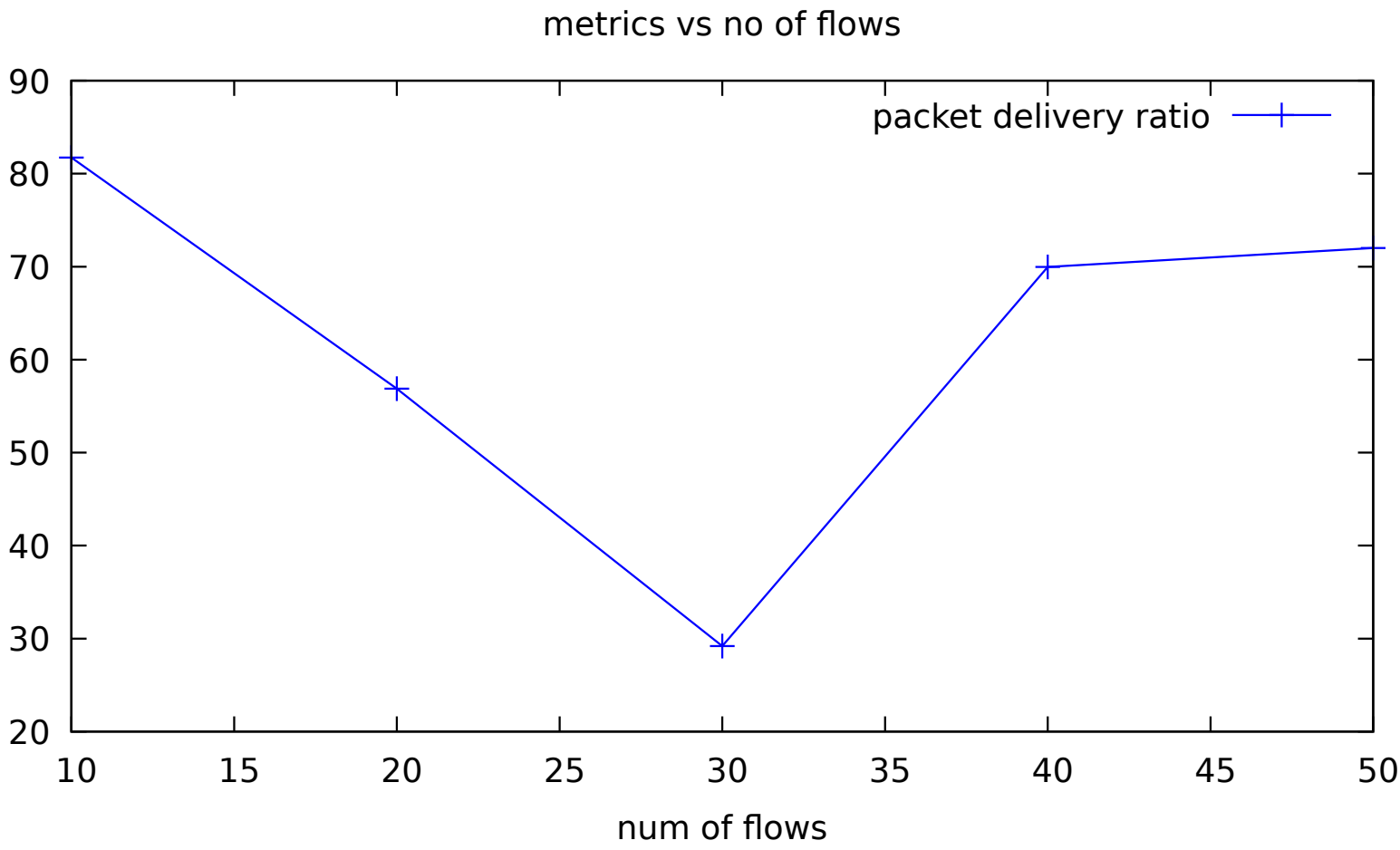

metrics vs no of nodes
packet delivery ratio
70
65
60
55
50
45
40
20
30
40
50
60
70
80
90
100
num of nodes

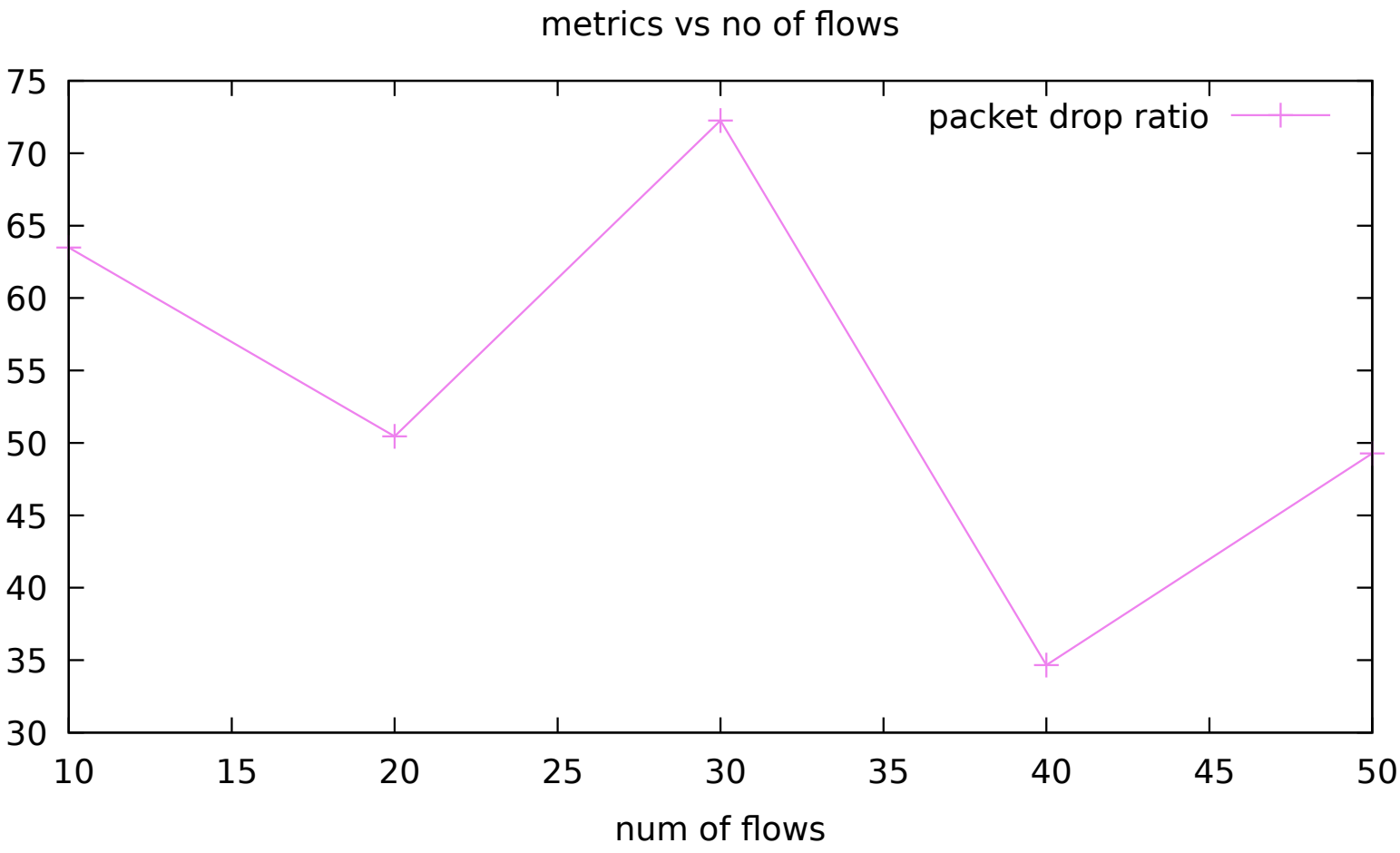

metrics vs no of nodes
packet drop ratio
85
80
75
70
65
60
55
20
30
40
50
60
70
80
90
100
num of nodes

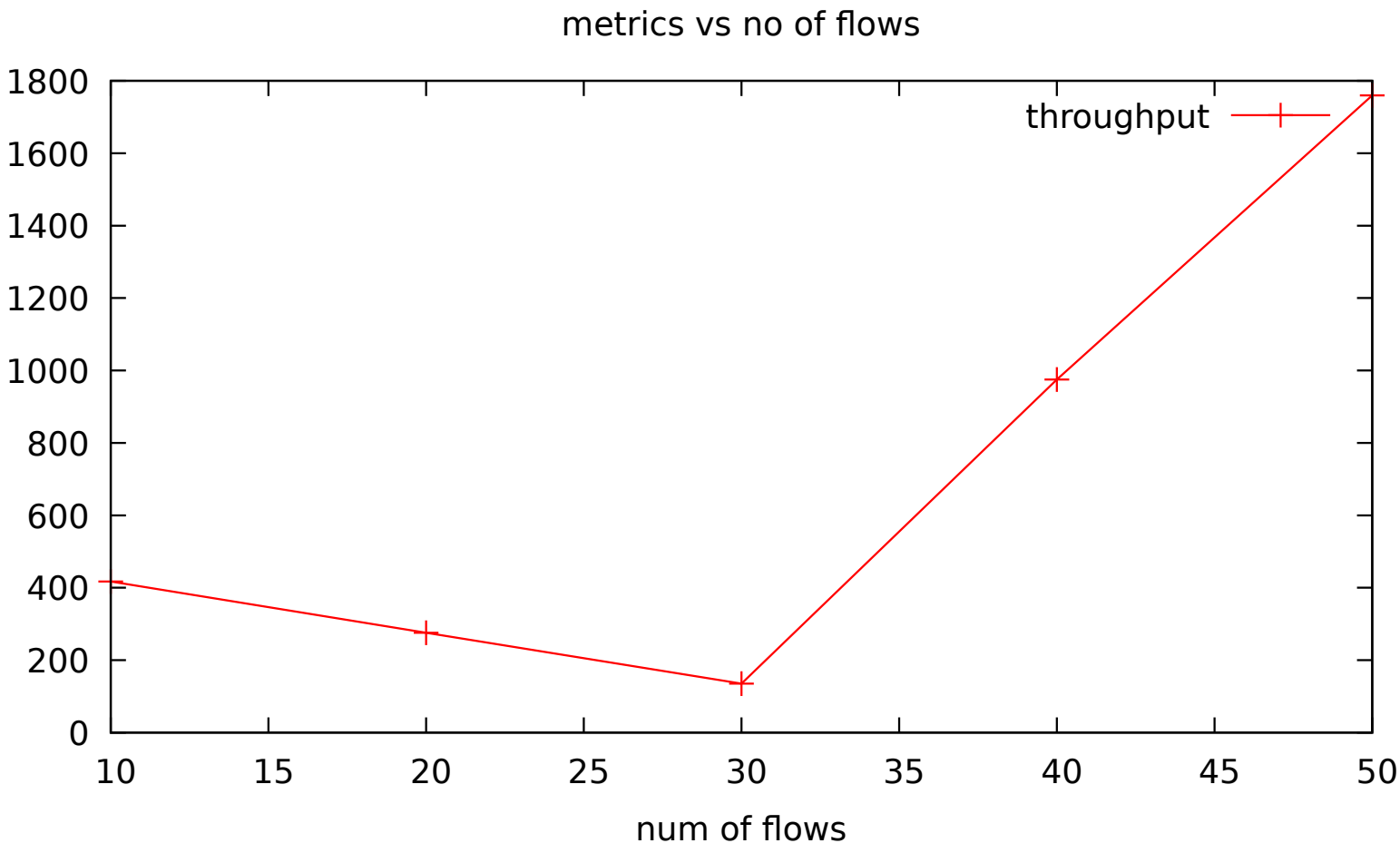

metrics vs no of nodes
throughput
1600
1400
1200
1000
800
600
400
20
30
40
50
60
70
80
90
100
num of nodes

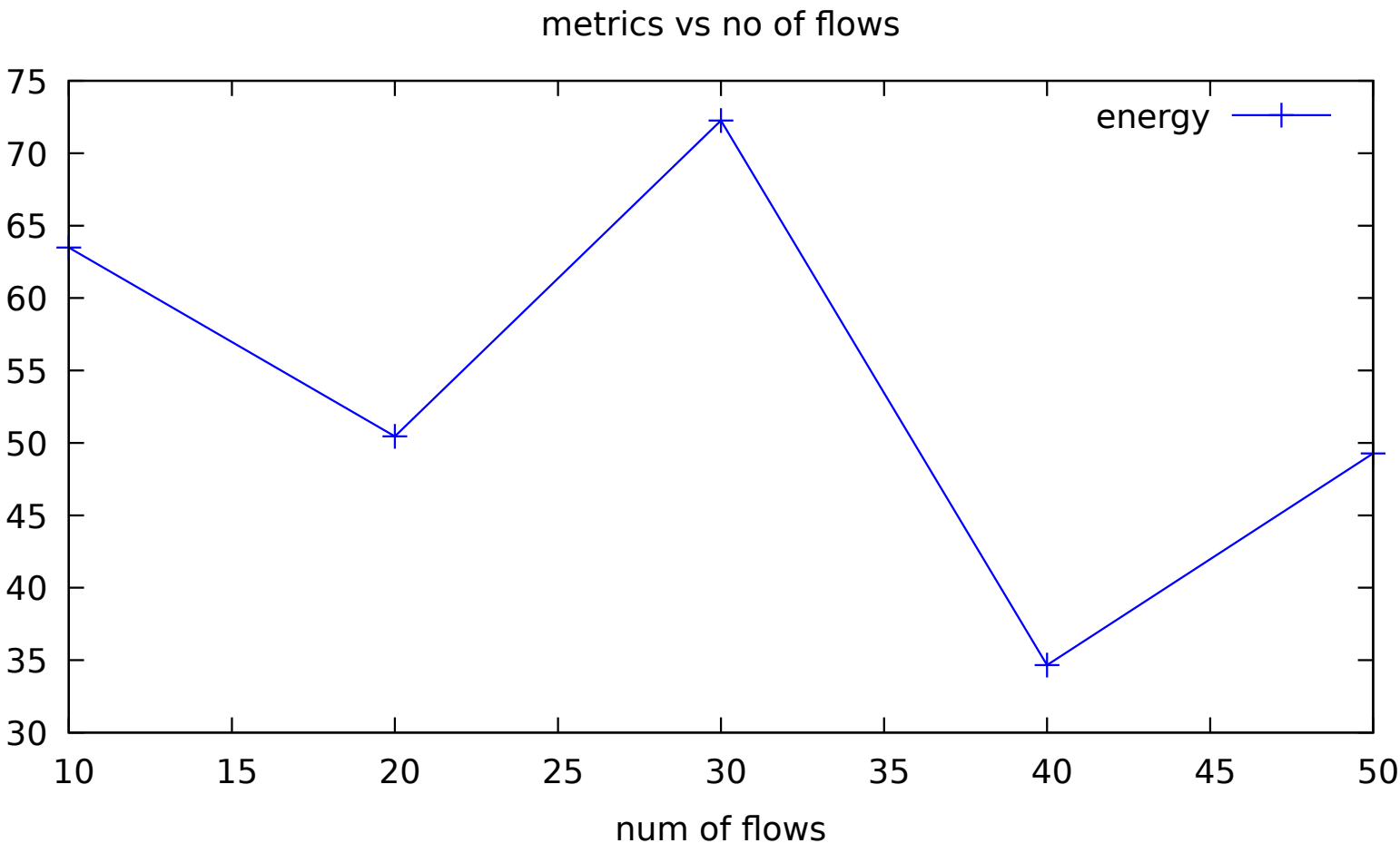

metrics vs no of nodes
energy
85
80
75
70
65
60
55
20
30
40
50
60
70
80
90
100
num of nodes

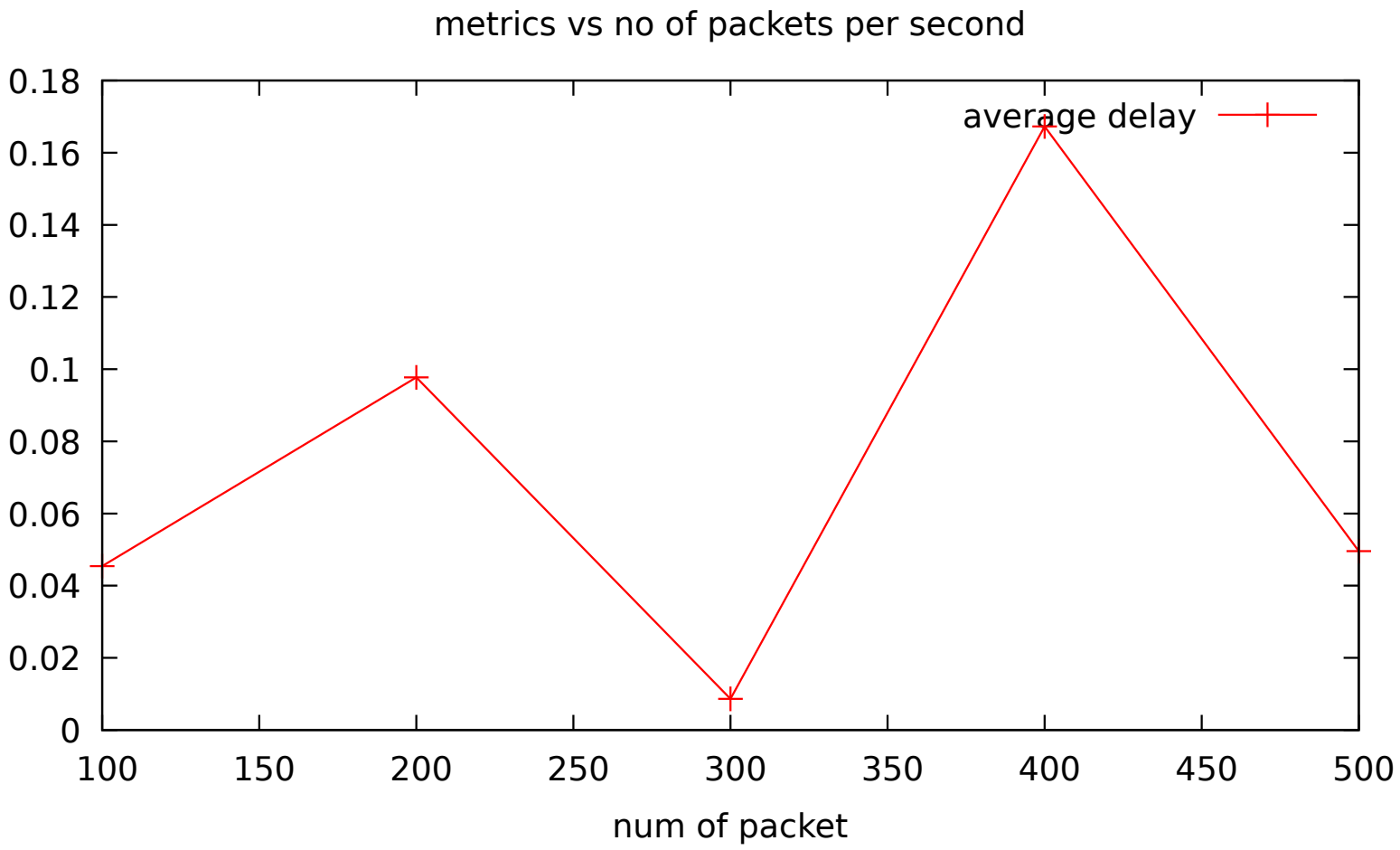

metrics vs no of flows
average delay
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
10
15
20
25
30
35
40
45
50
num of flows

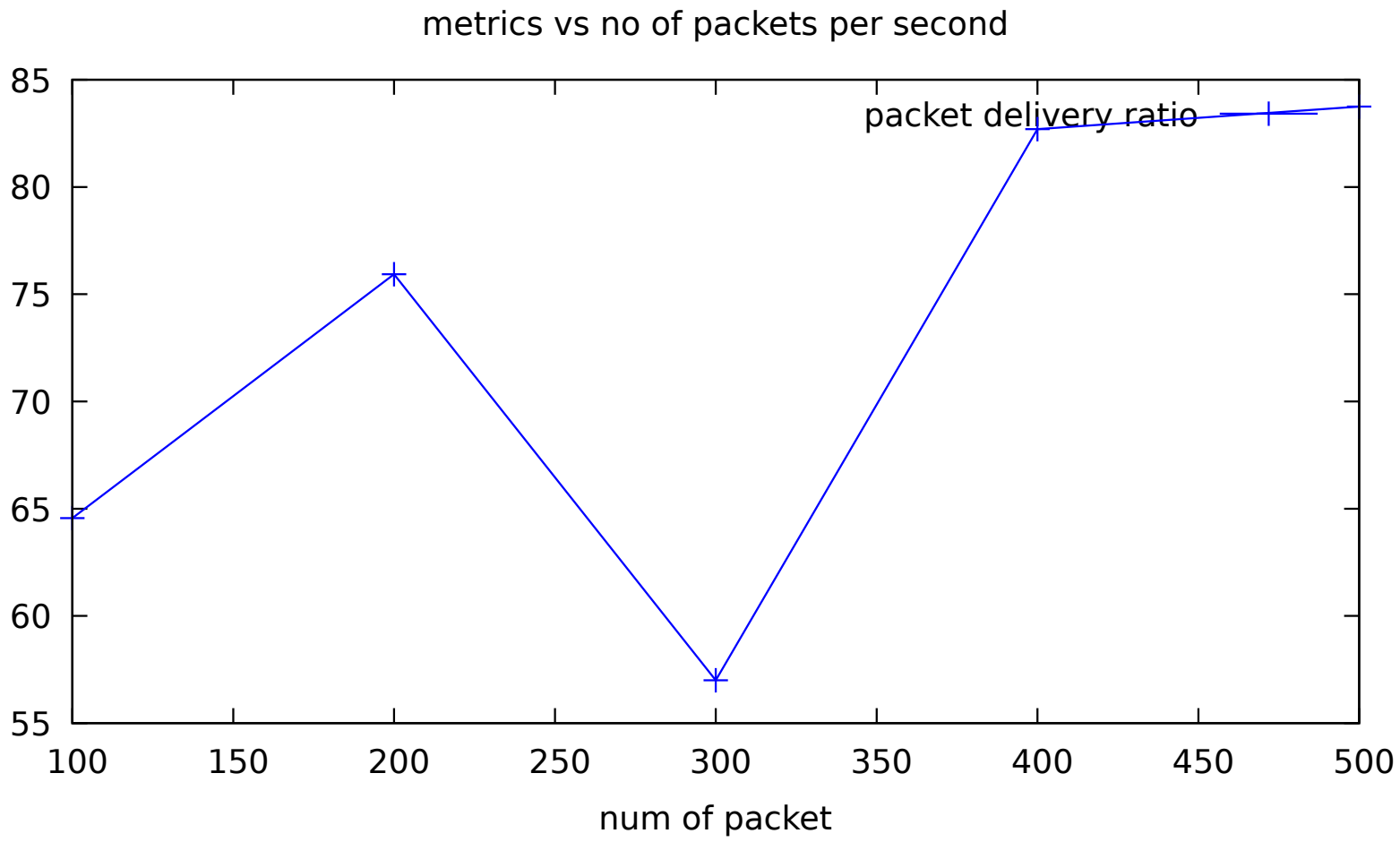

metrics vs no of flows
packet delivery ratio
90
80
70
60
50
40
30
20
10
15
20
25
30
35
40
45
50
num of flows

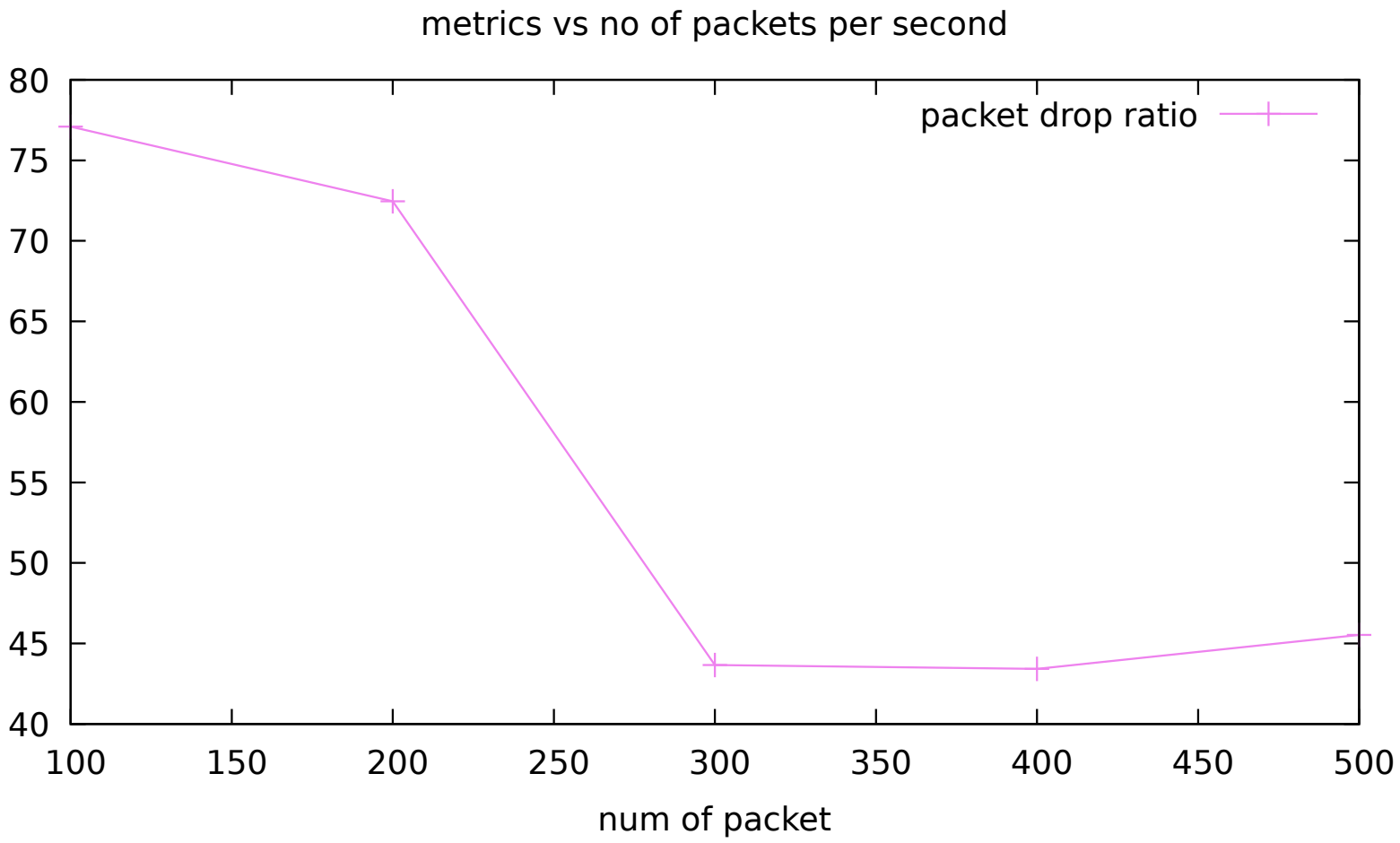

metrics vs no of flows
packet drop ratio
75
70
65
60
55
50
45
40
35
30
10
15
20
25
30
35
40
45
50
num of flows

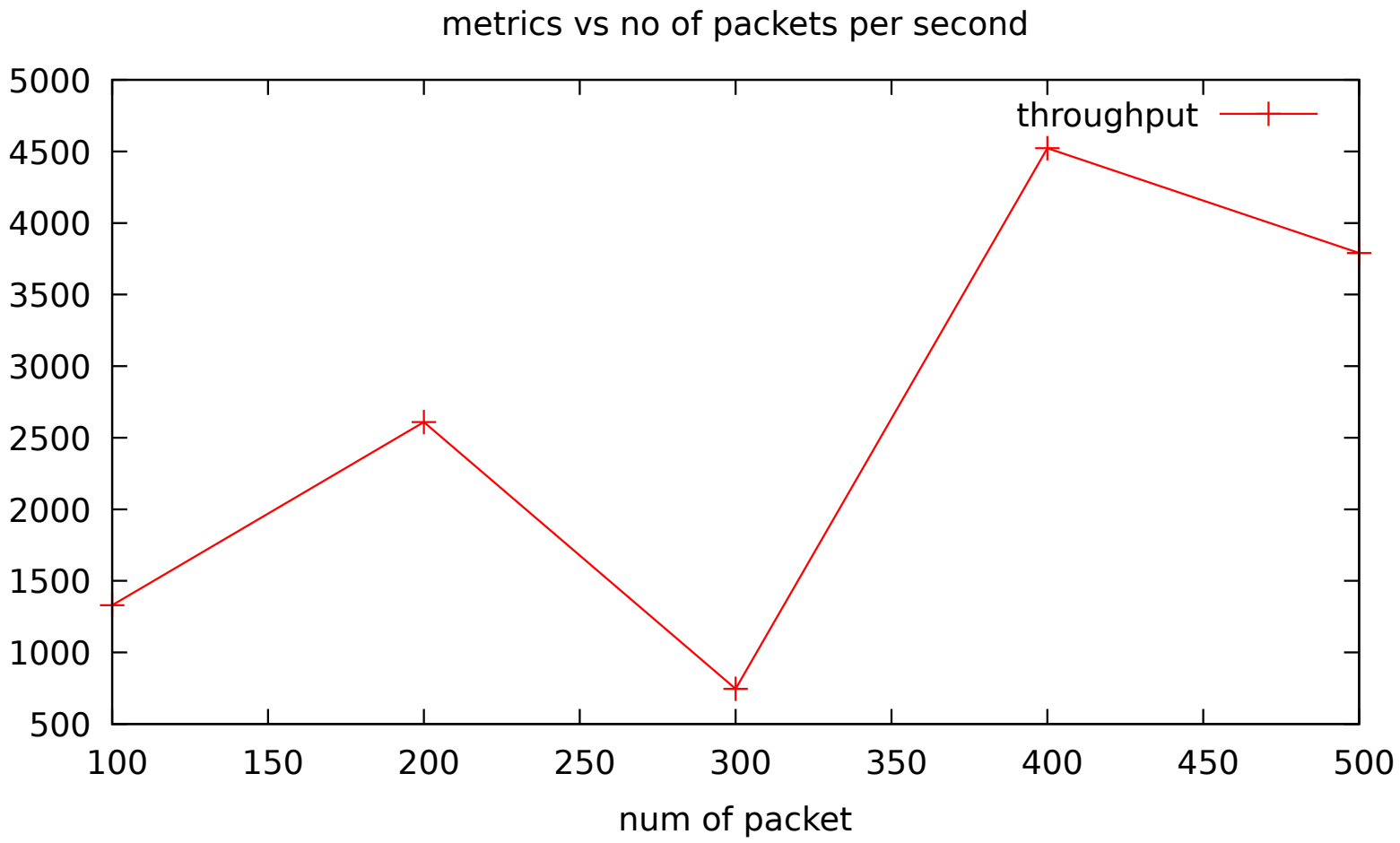

metrics vs no of flows
throughput
1800
1600
1400
1200
1000
800
600
400
200
0
10
15
20
25
30
35
40
45
50
num of flows

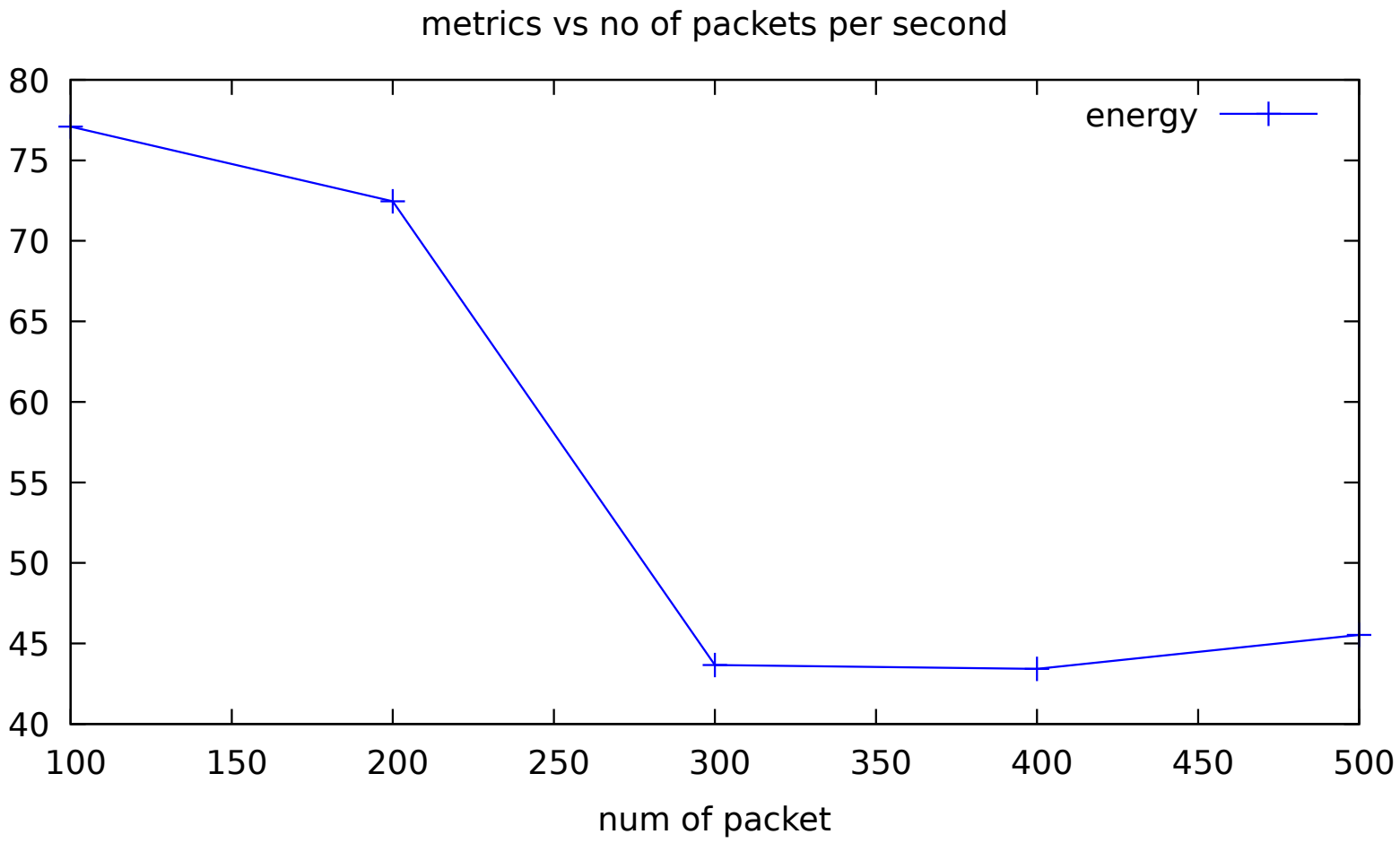

metrics vs no of flows
energy
75
70
65
60
55
50
45
40
35
30
10
15
20
25
30
35
40
45
50
num of flows

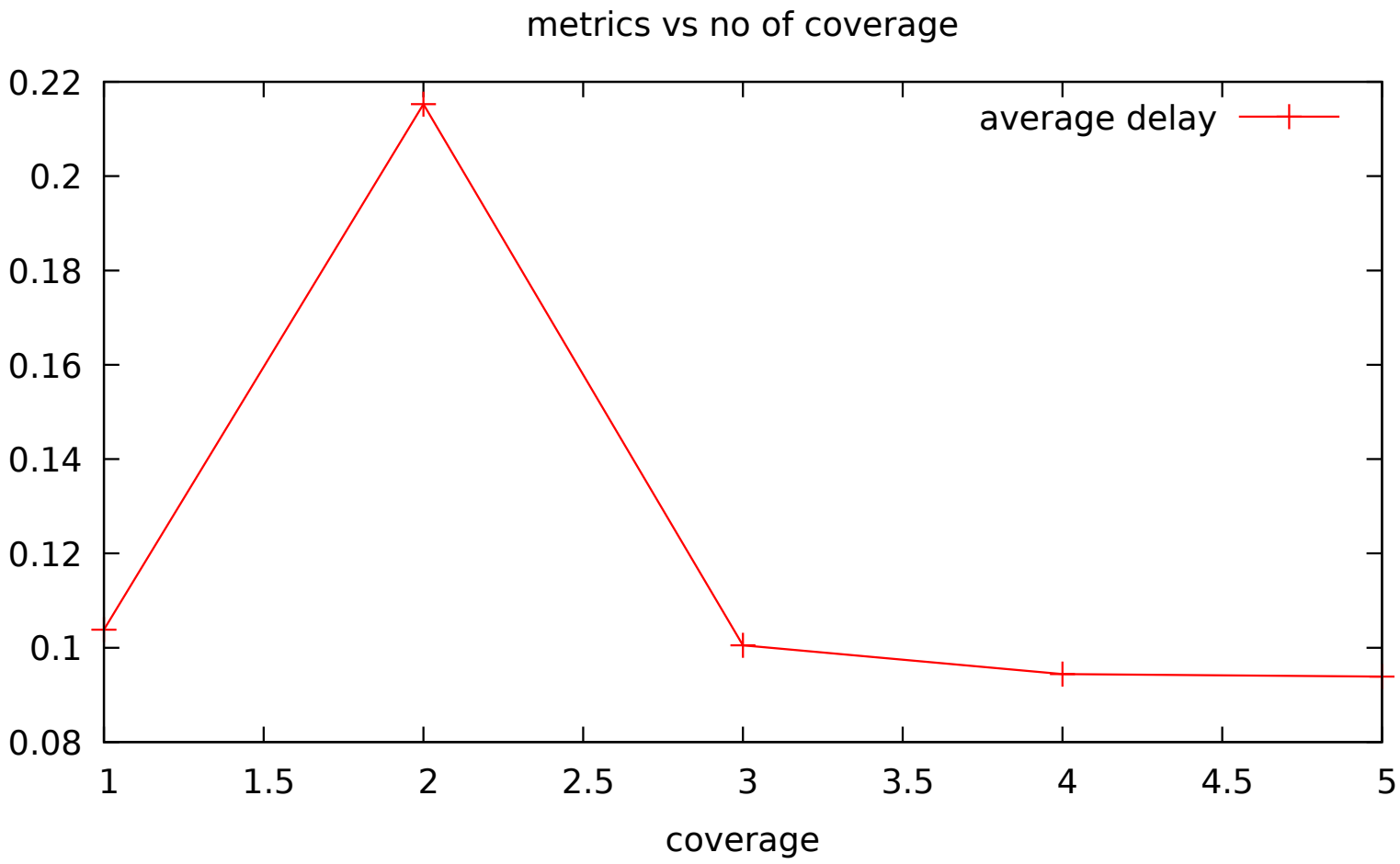

metrics vs no of packets per second
average delay
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
100
150
200
250
300
350
400
450
500
num of packet

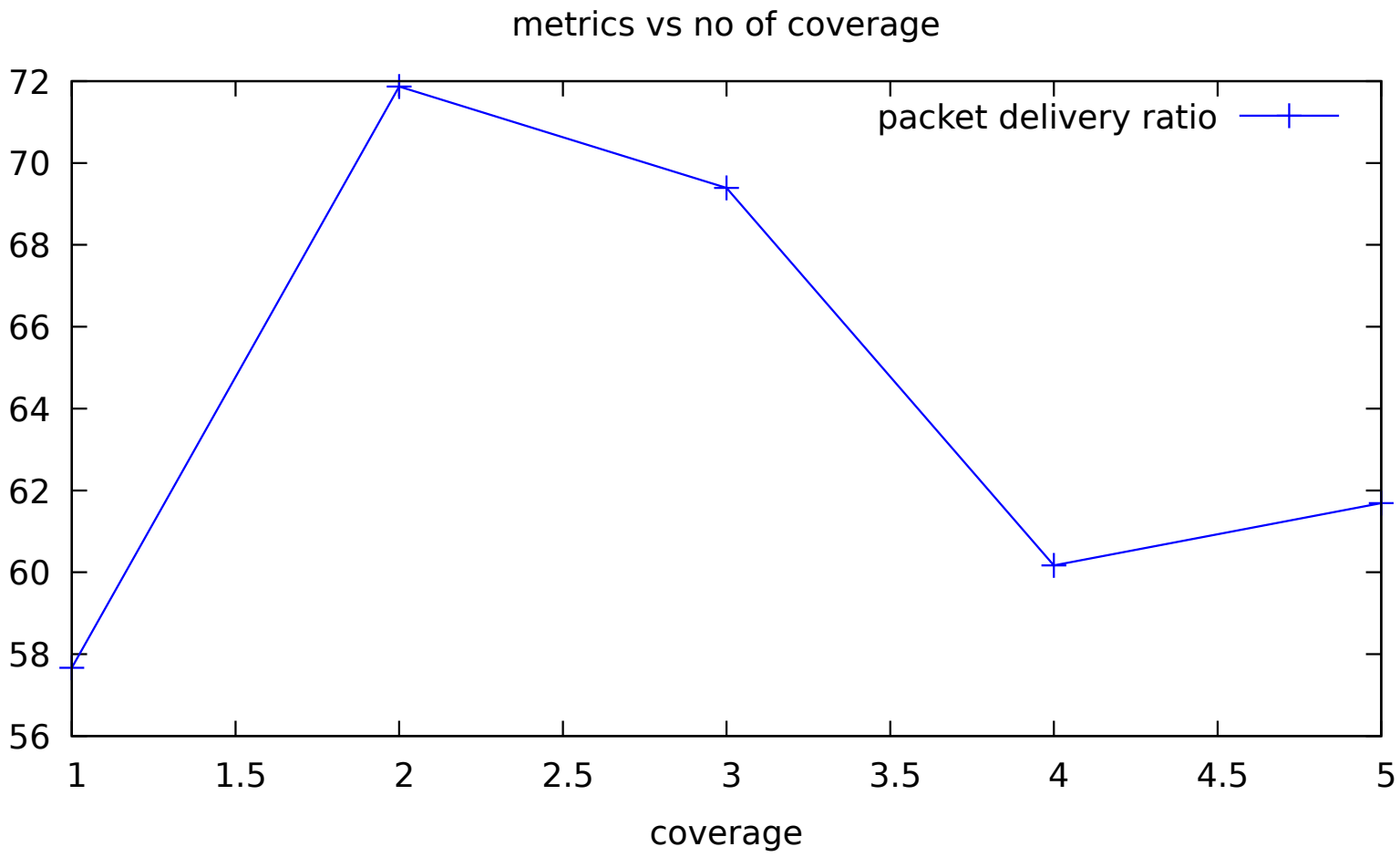

metrics vs no of packets per second
packet delivery ratio
85
80
75
70
65
60
55
100
150
200
250
300
350
400
450
500
num of packet

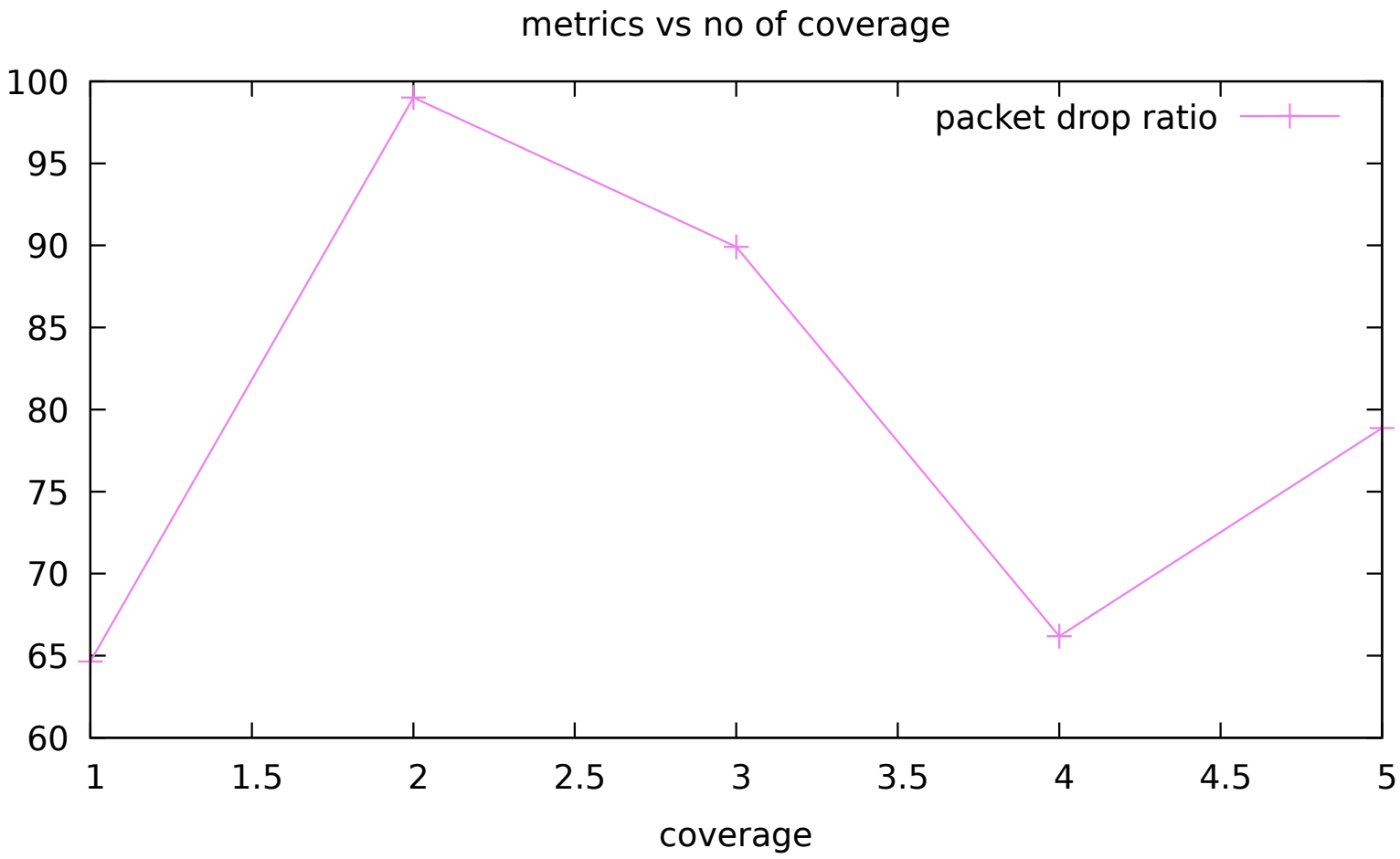

metrics vs no of packets per second
packet drop ratio
80
75
70
65
60
55
50
45
40
100
150
200
250
300
350
400
450
500
num of packet

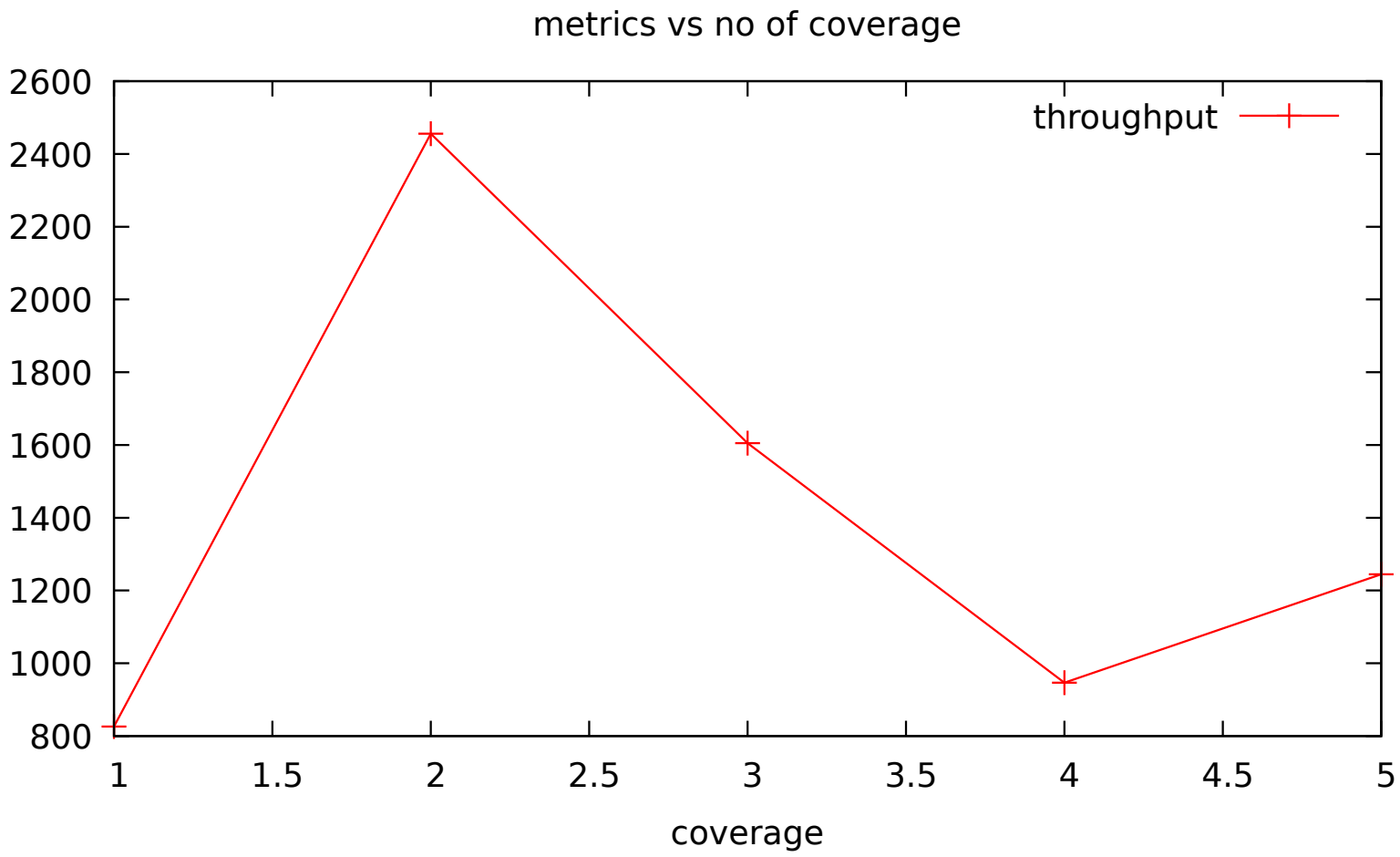

metrics vs no of packets per second
throughput
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
100
150
200
250
300
350
400
450
500
num of packet

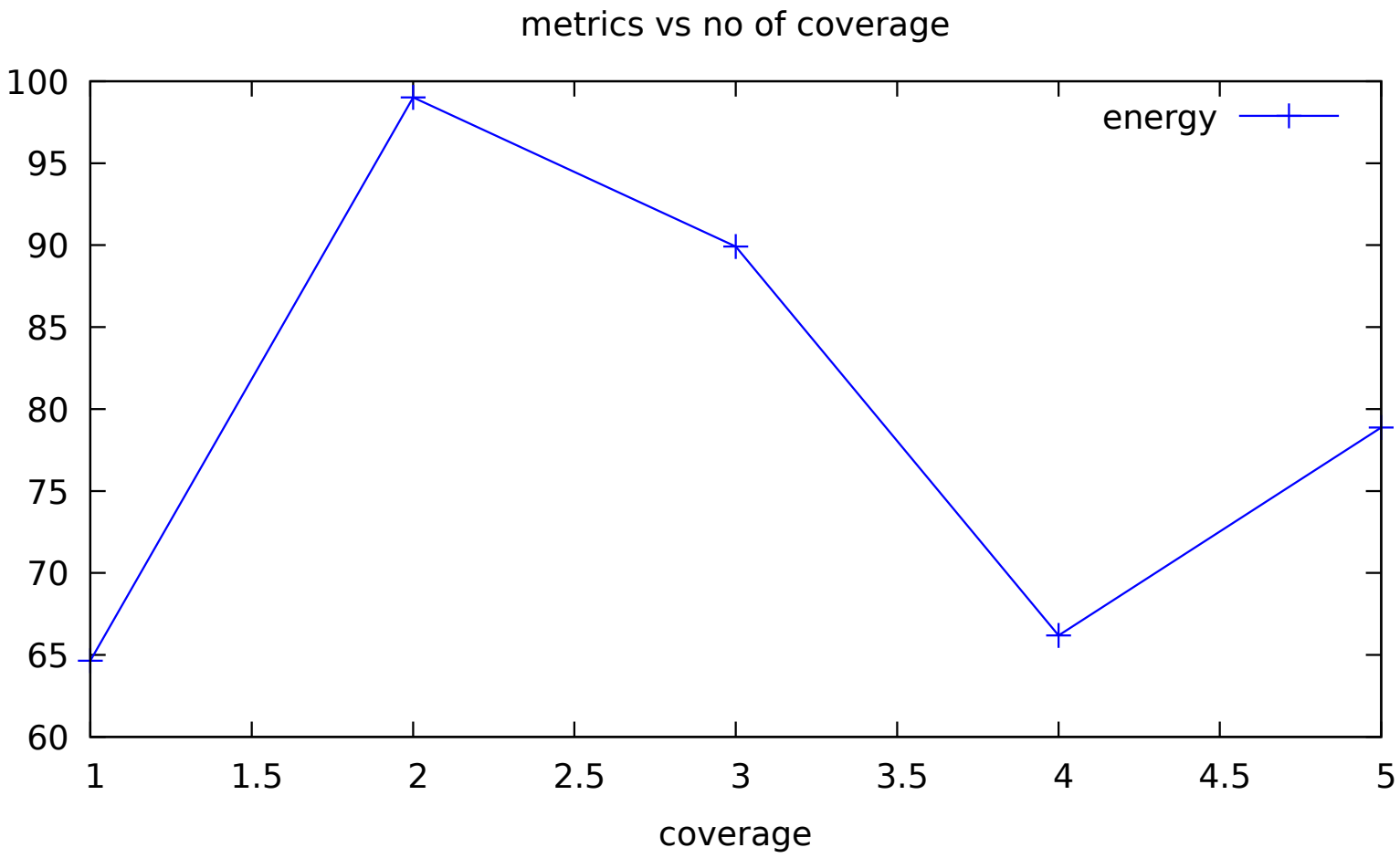

metrics vs no of packets per second
energy
80
75
70
65
60
55
50
45
40
100
150
200
250
300
350
400
450
500
num of packet


metrics vs no of coverage
average delay
0.22
0.2
0.18
0.16
0.14
0.12
0.1
0.08
1
1.5
2
2.5
3
3.5
4
4.5
5
coverage


metrics vs no of coverage
packet delivery ratio
72
70
68
66
64
62
60
58
56
1
1.5
2
2.5
3
3.5
4
4.5
5
coverage


metrics vs no of coverage
packet drop ratio
100
95
90
85
80
75
70
65
60
1
1.5
2
2.5
3
3.5
4
4.5
5
coverage


metrics vs no of coverage
throughput
2600
2400
2200
2000
1800
1600
1400
1200
1000
800
1
1.5
2
2.5
3
3.5
4
4.5
5
coverage

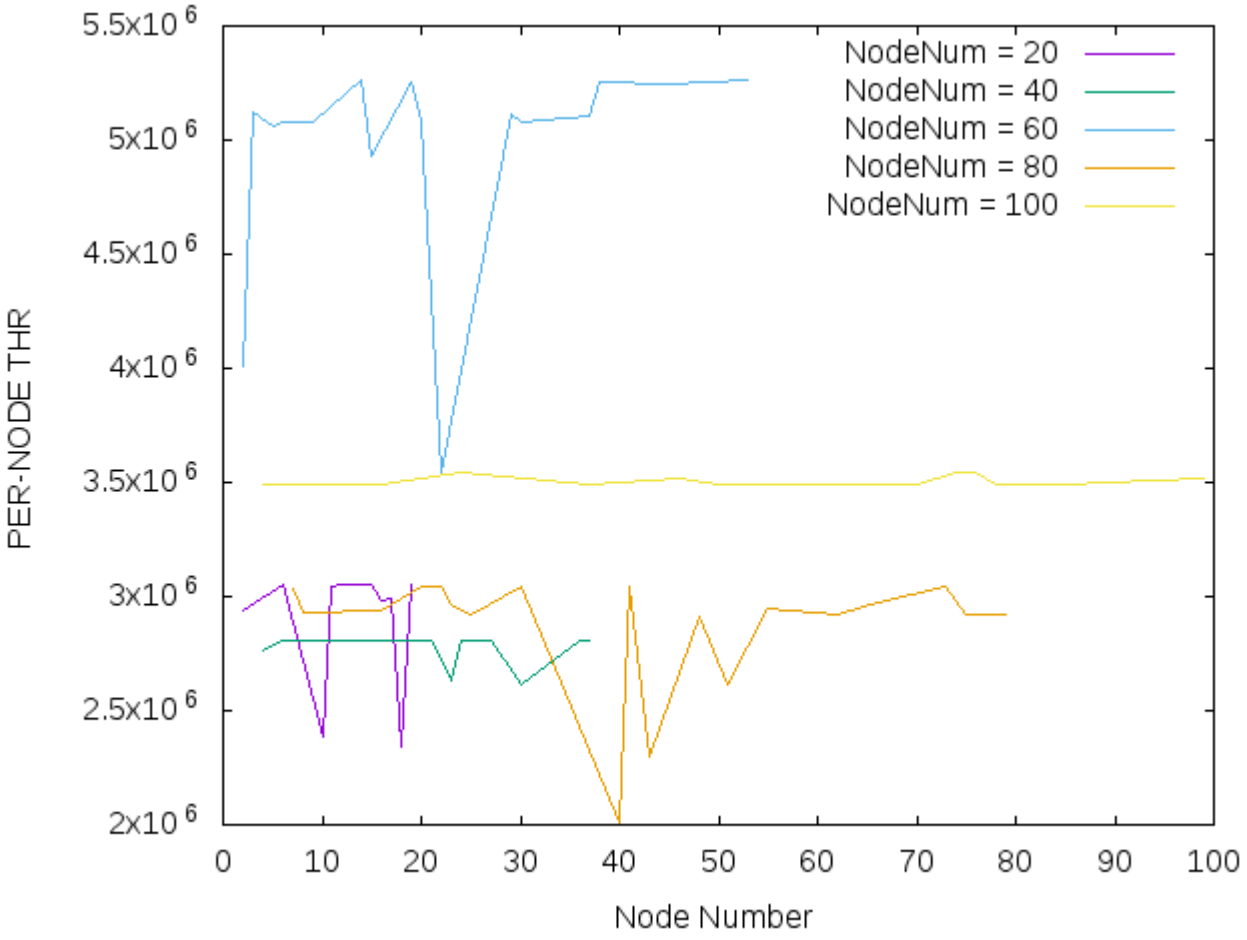

metrics vs no of coverage
energy
100
95
90
85
80
75
70
65
60
1
1.5
2
2.5
3
3.5
4
4.5
5
coverage

7 Bonus Section

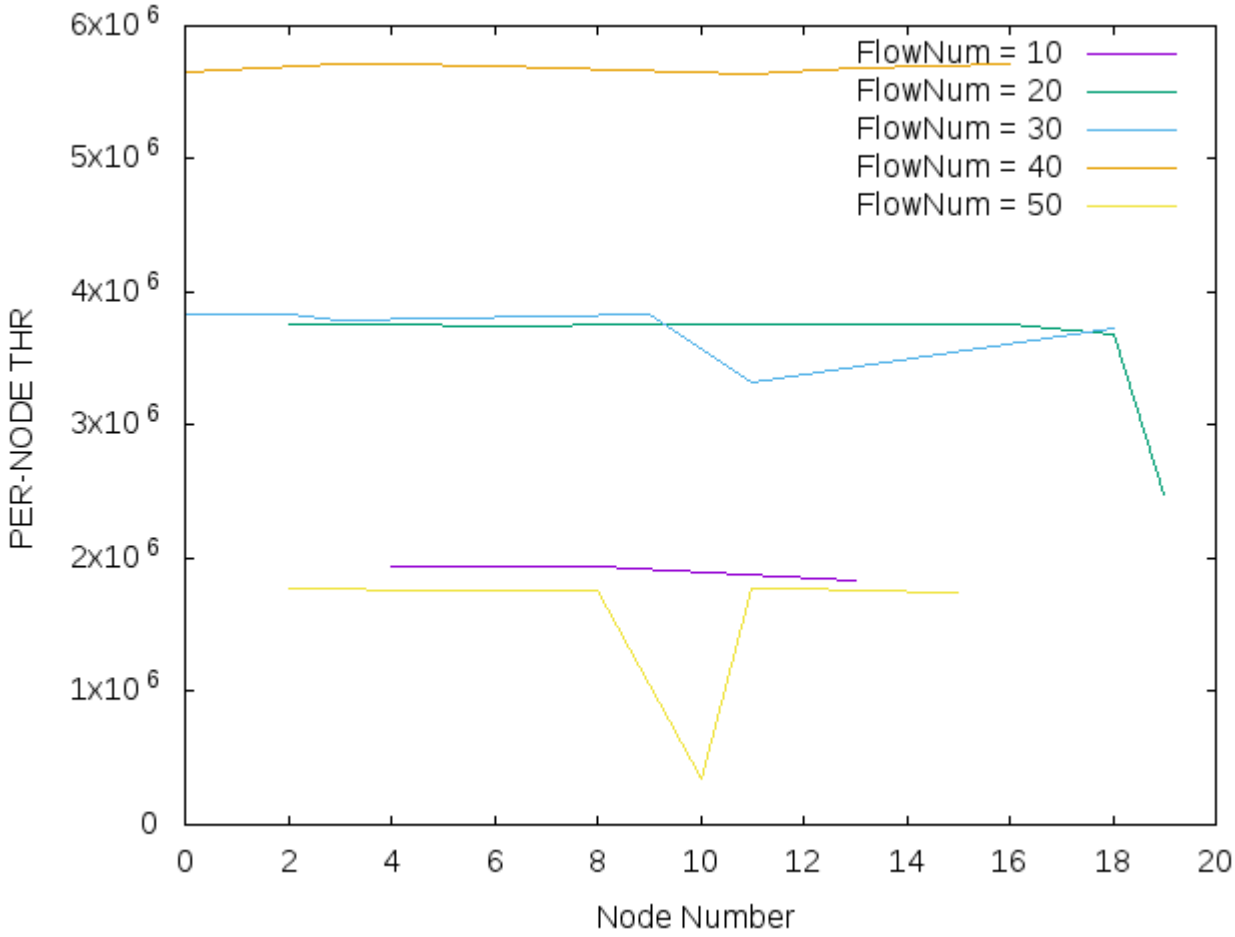
7.1 Wireless Mobile

- We've calculated per node throughput varying value of no of nodes ,no of flows,speed of nodes, queue size etc. The graphs for wireless mobile are attached below.
- We've also varied queue size and thus calculated other parameters. Related graphs are attached below for wireless mobile.

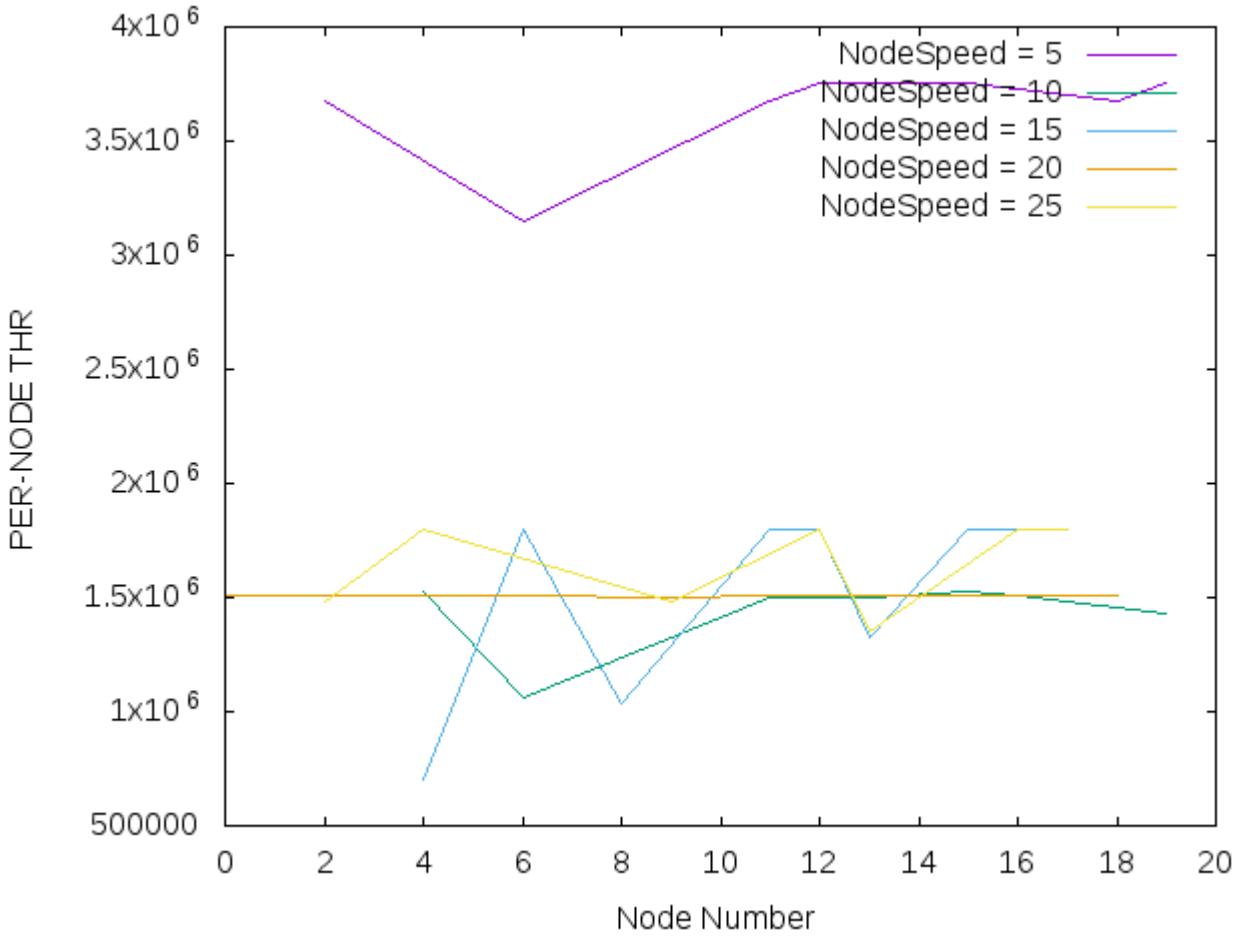
THR VS NodeNum.png



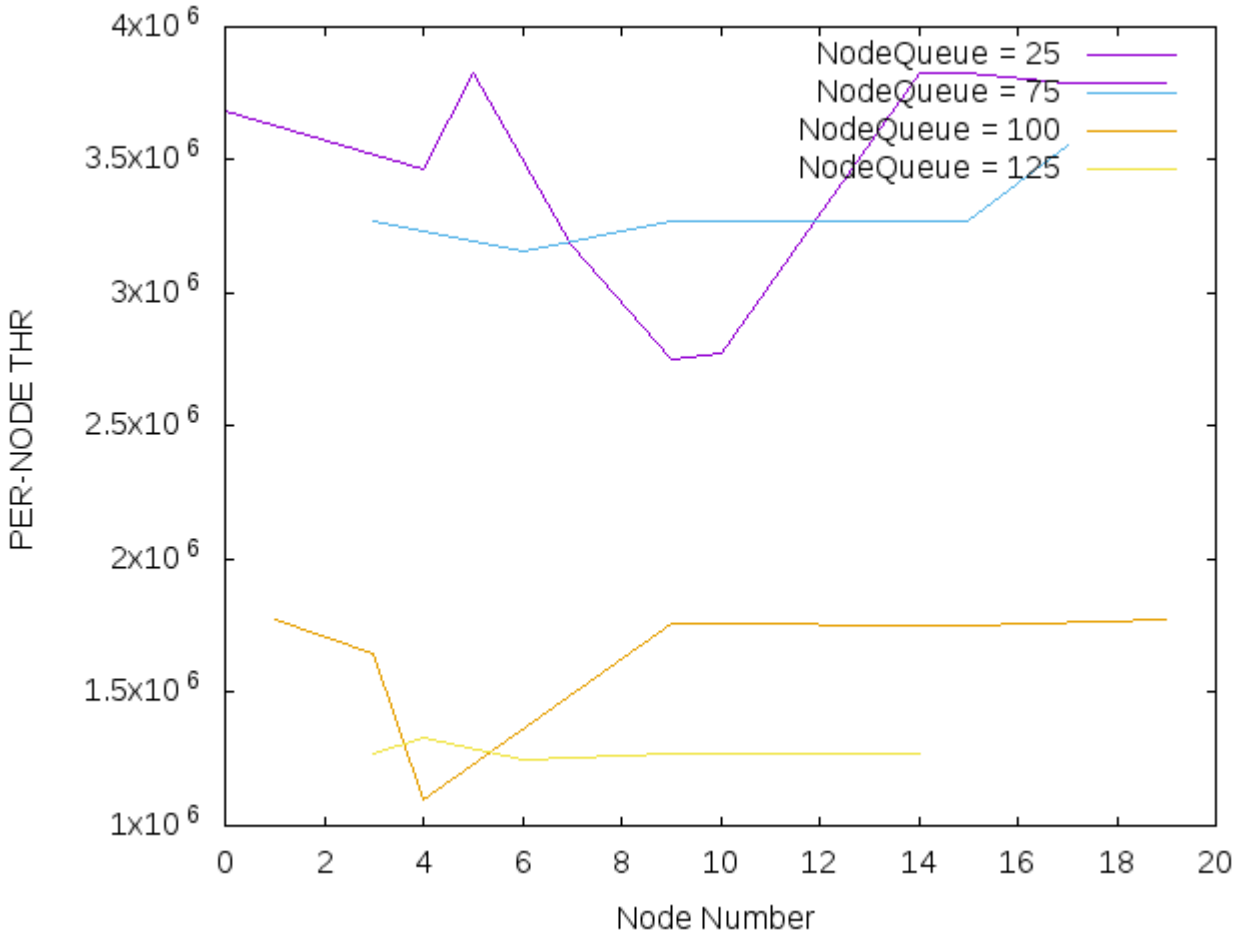
THR VS FlowNum.png



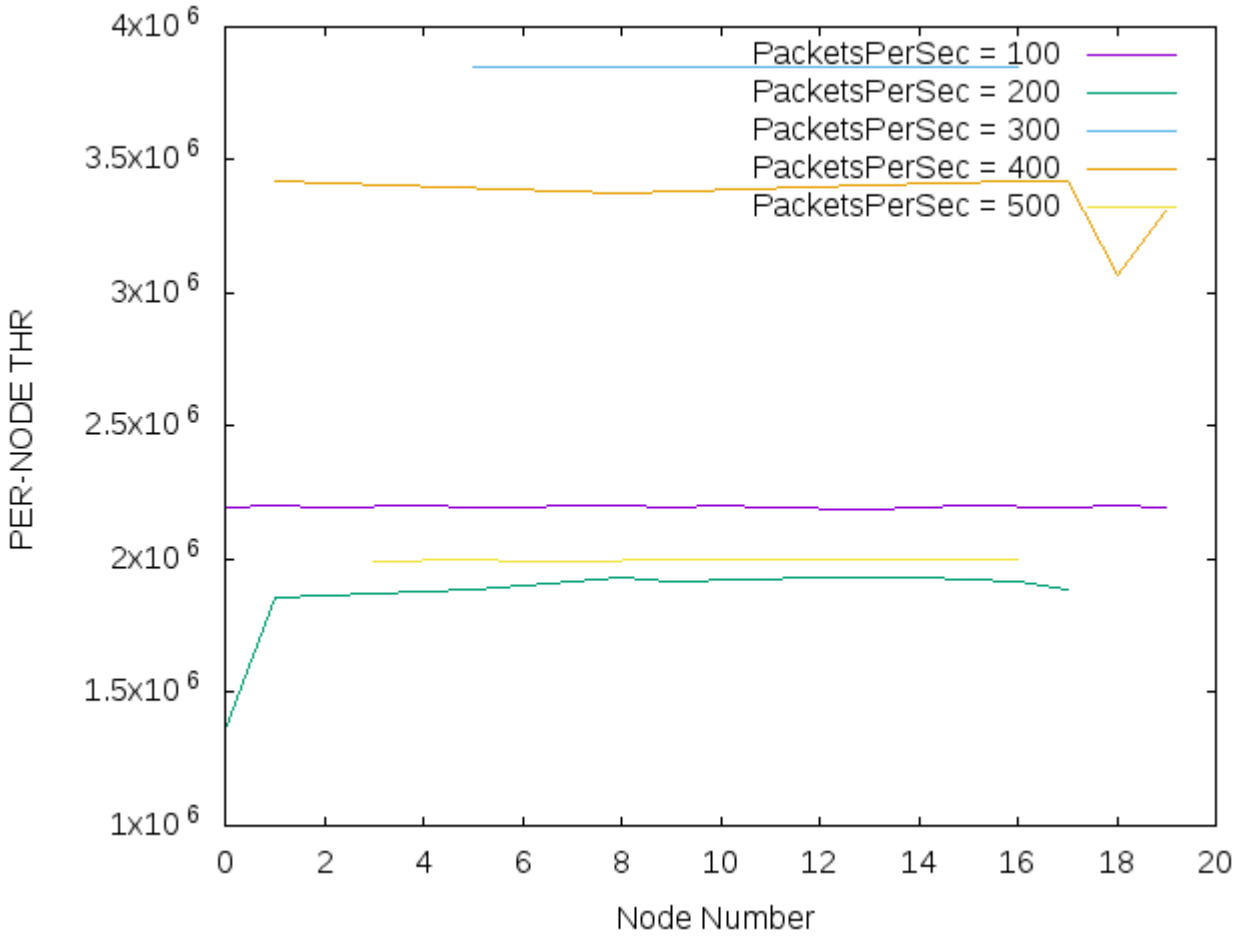
THR VS NodeSpeed.png



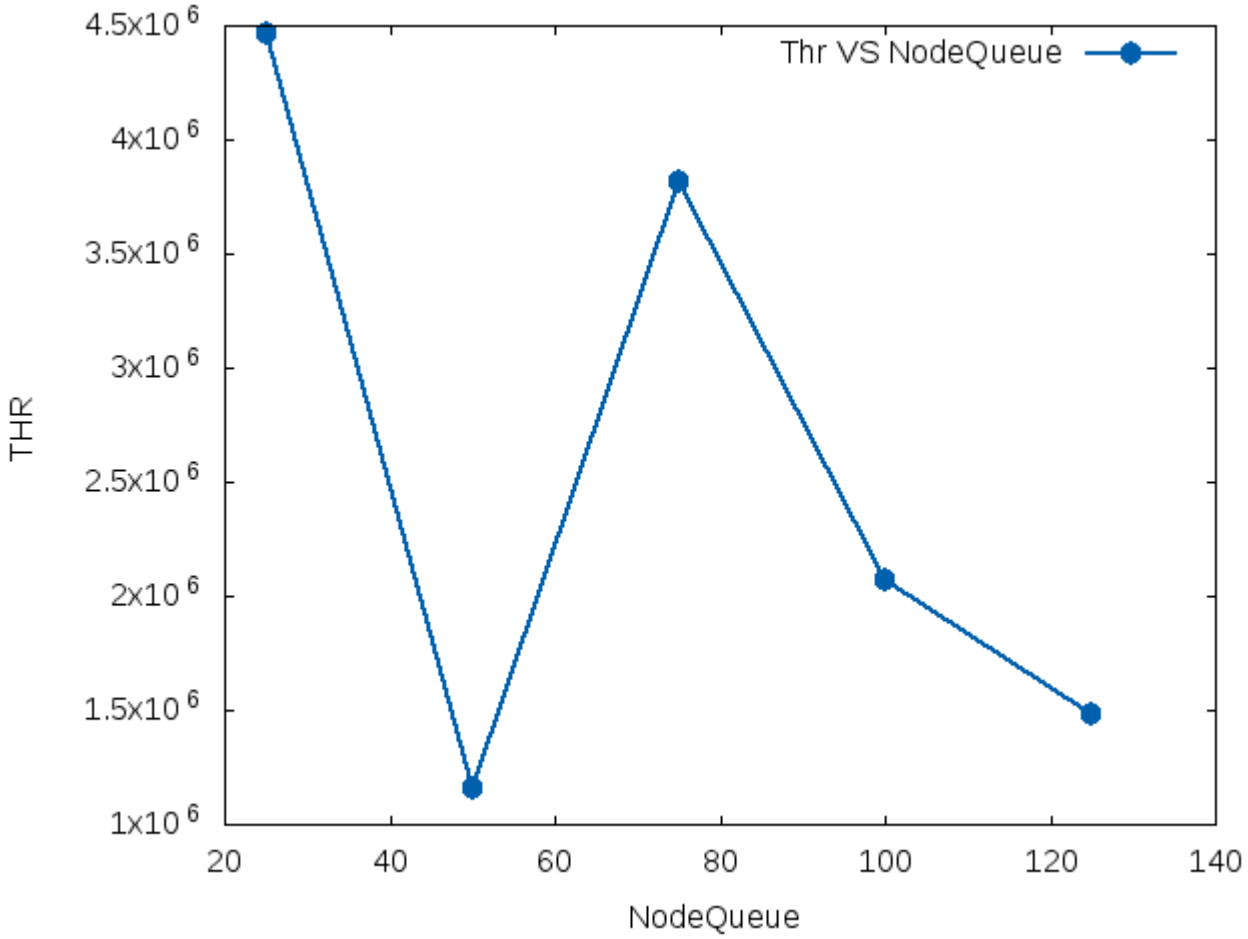
THR VS NodeQueue.png



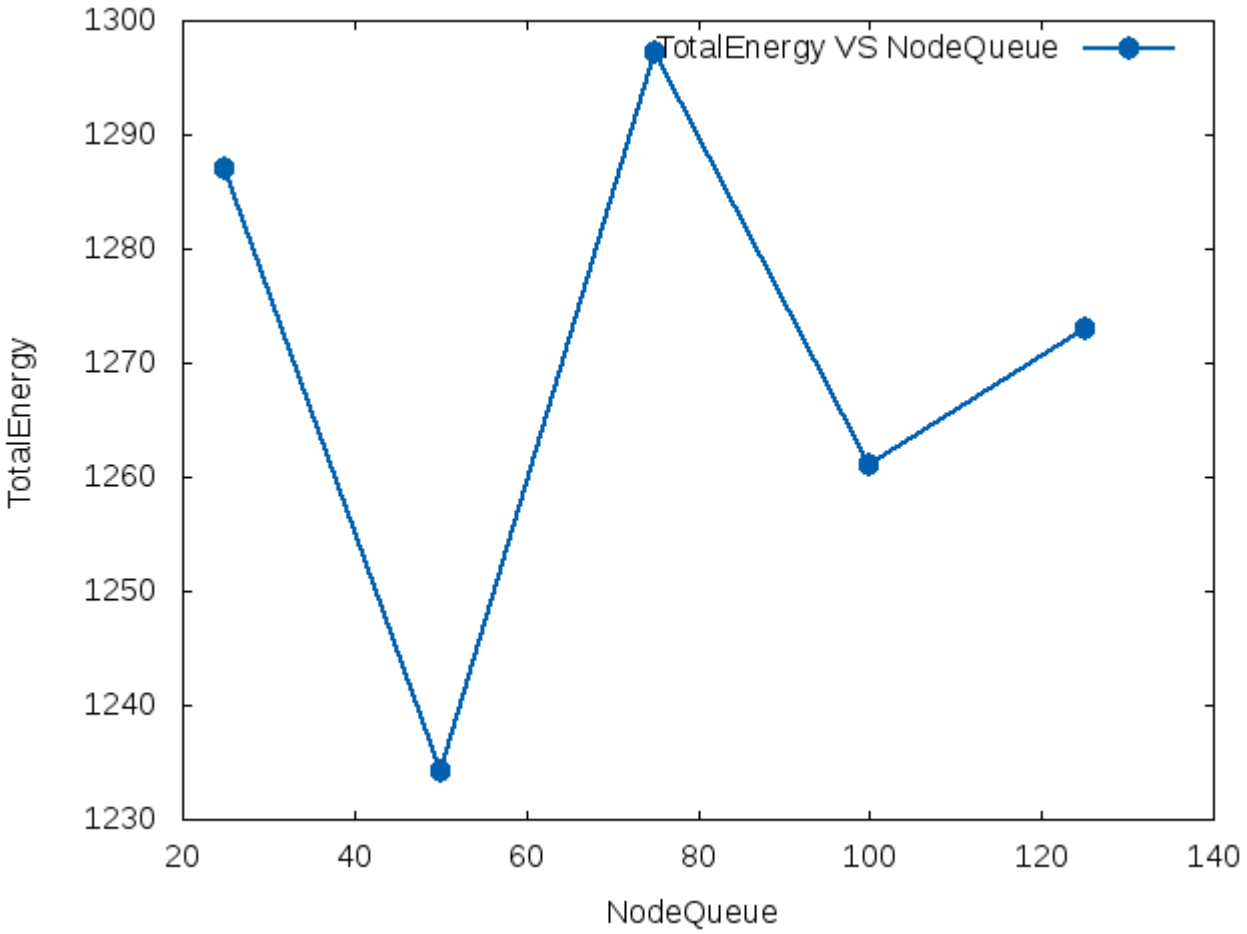
THR VS PacketsPerSec.png



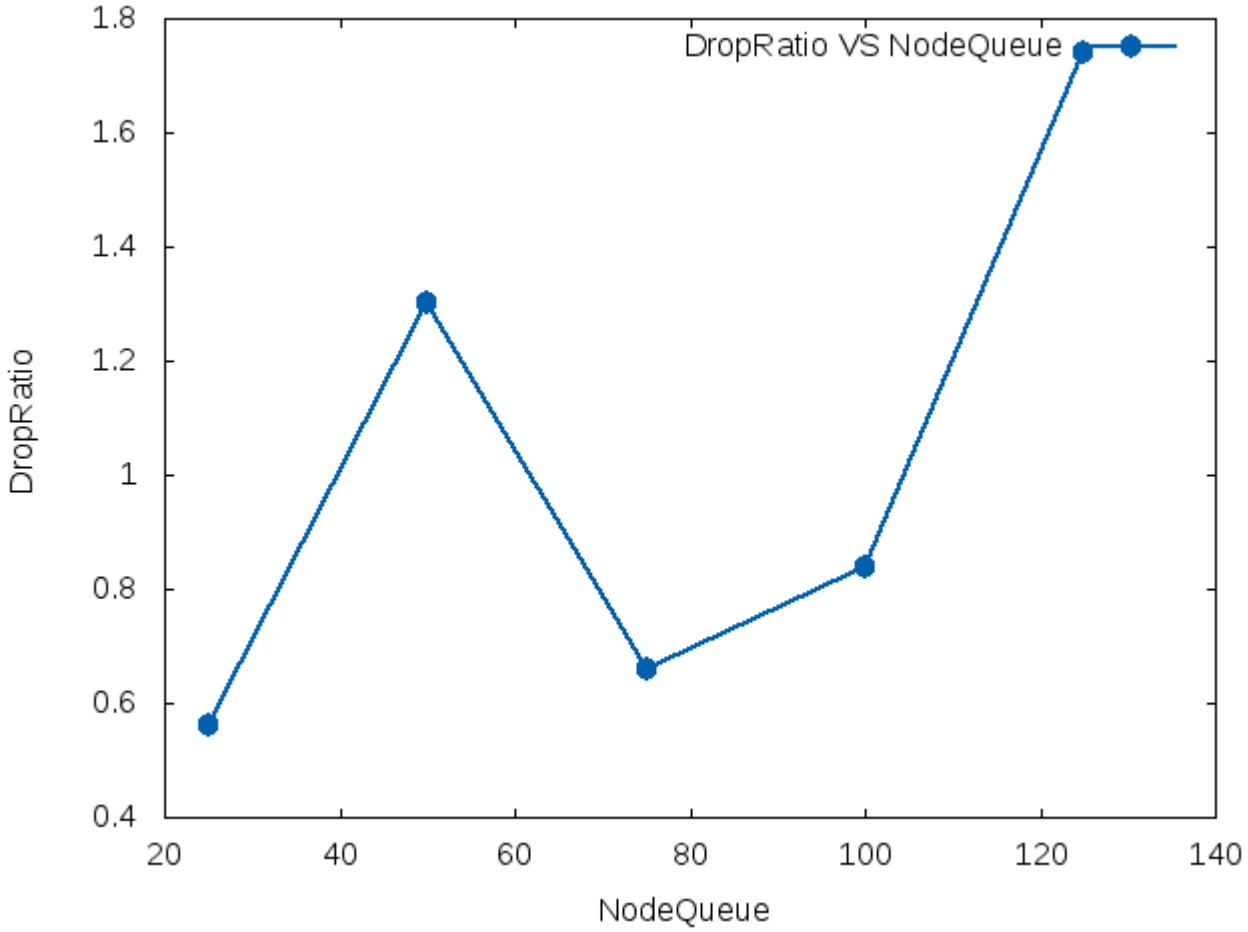
VS NodeQueue.png



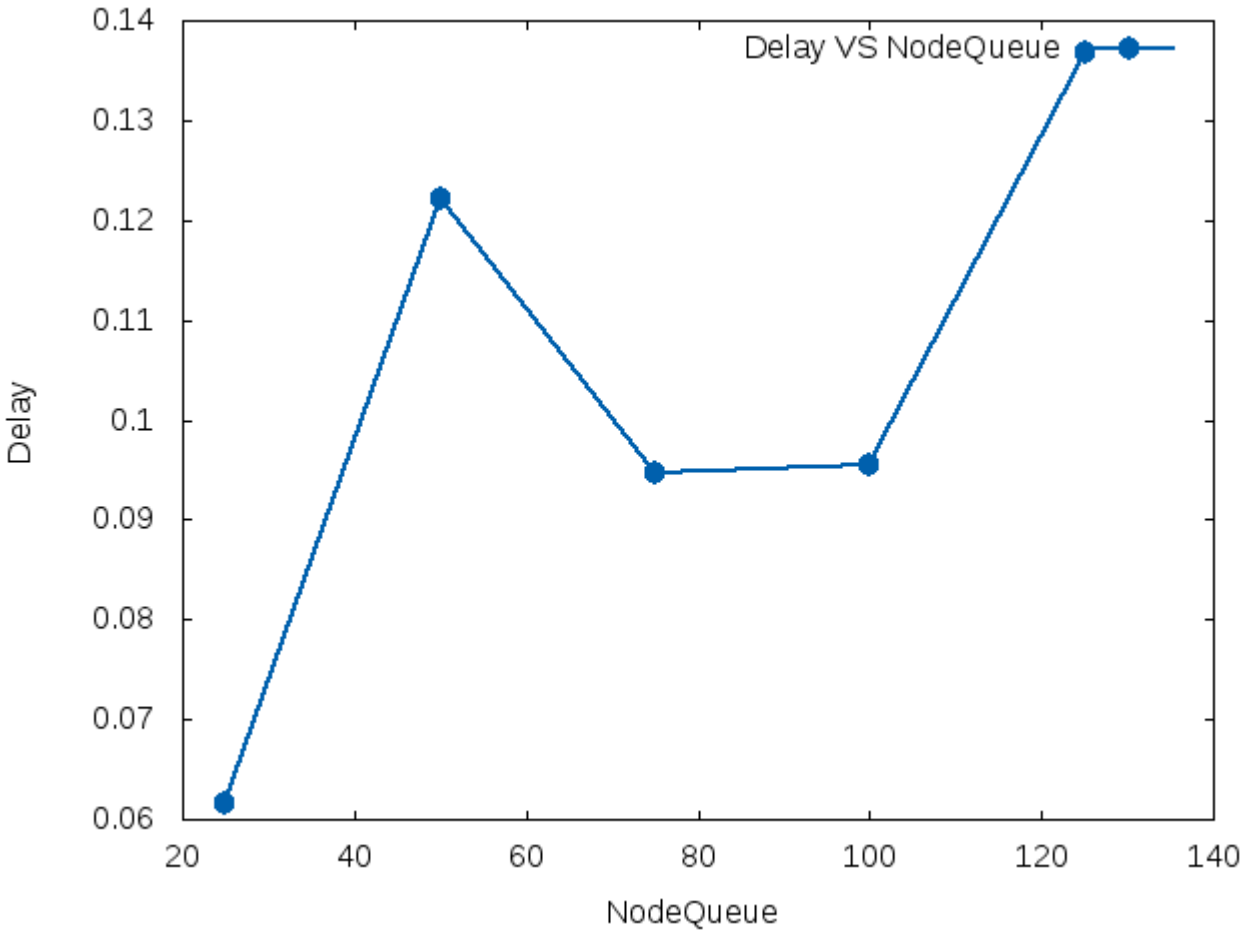
VS NodeQueue.png



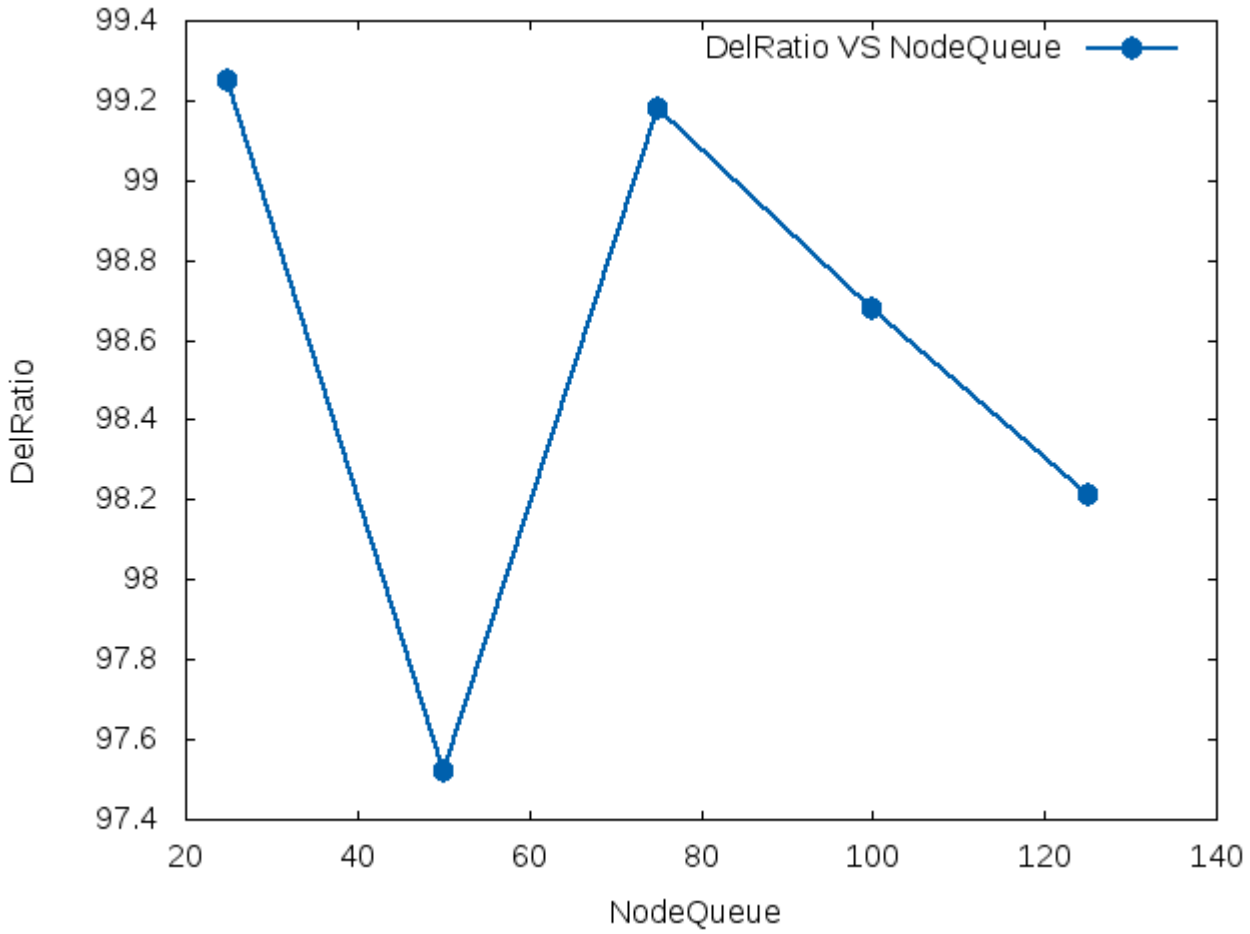
VS NodeQueue.png



VS NodeQueue.png

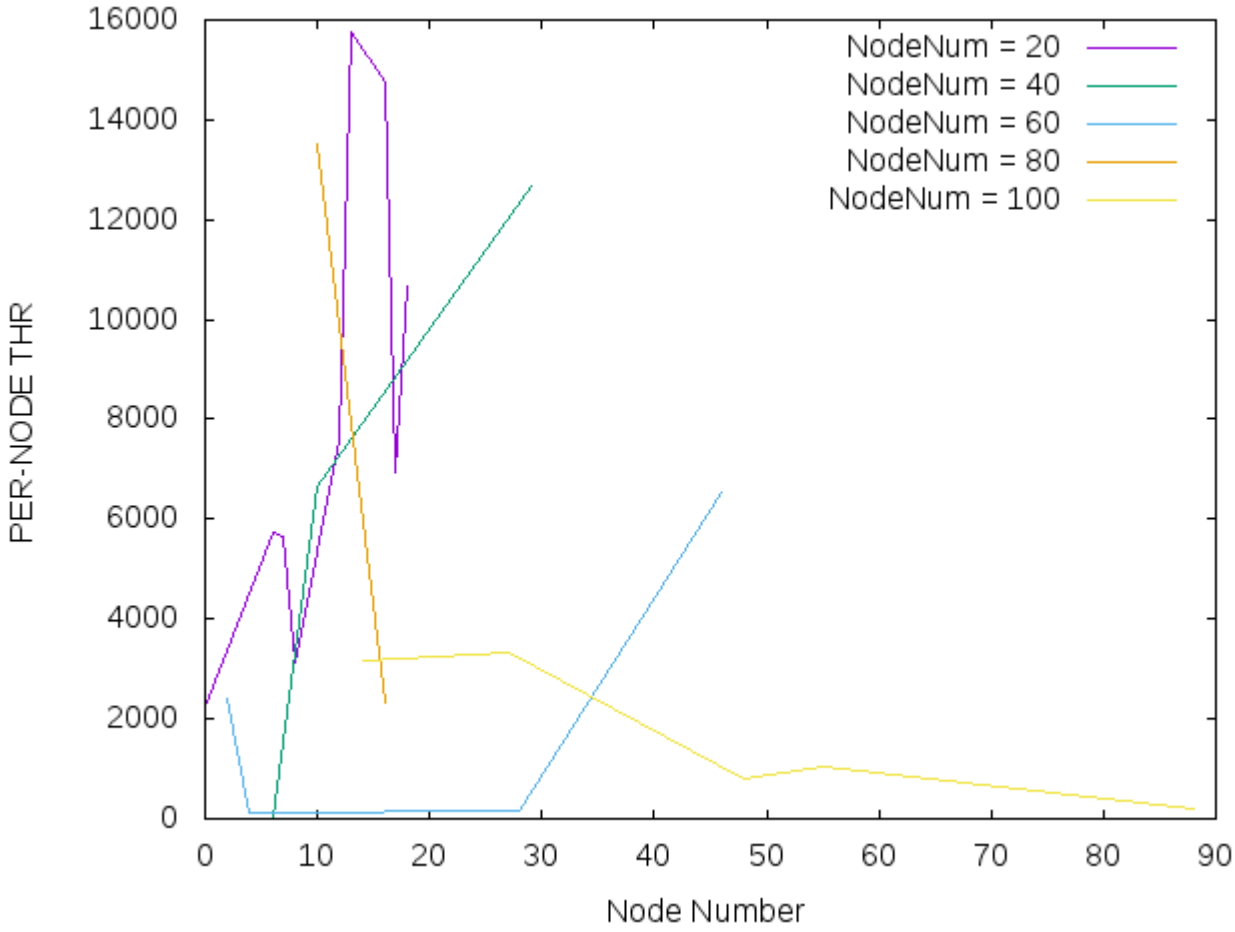


VS NodeQueue.png

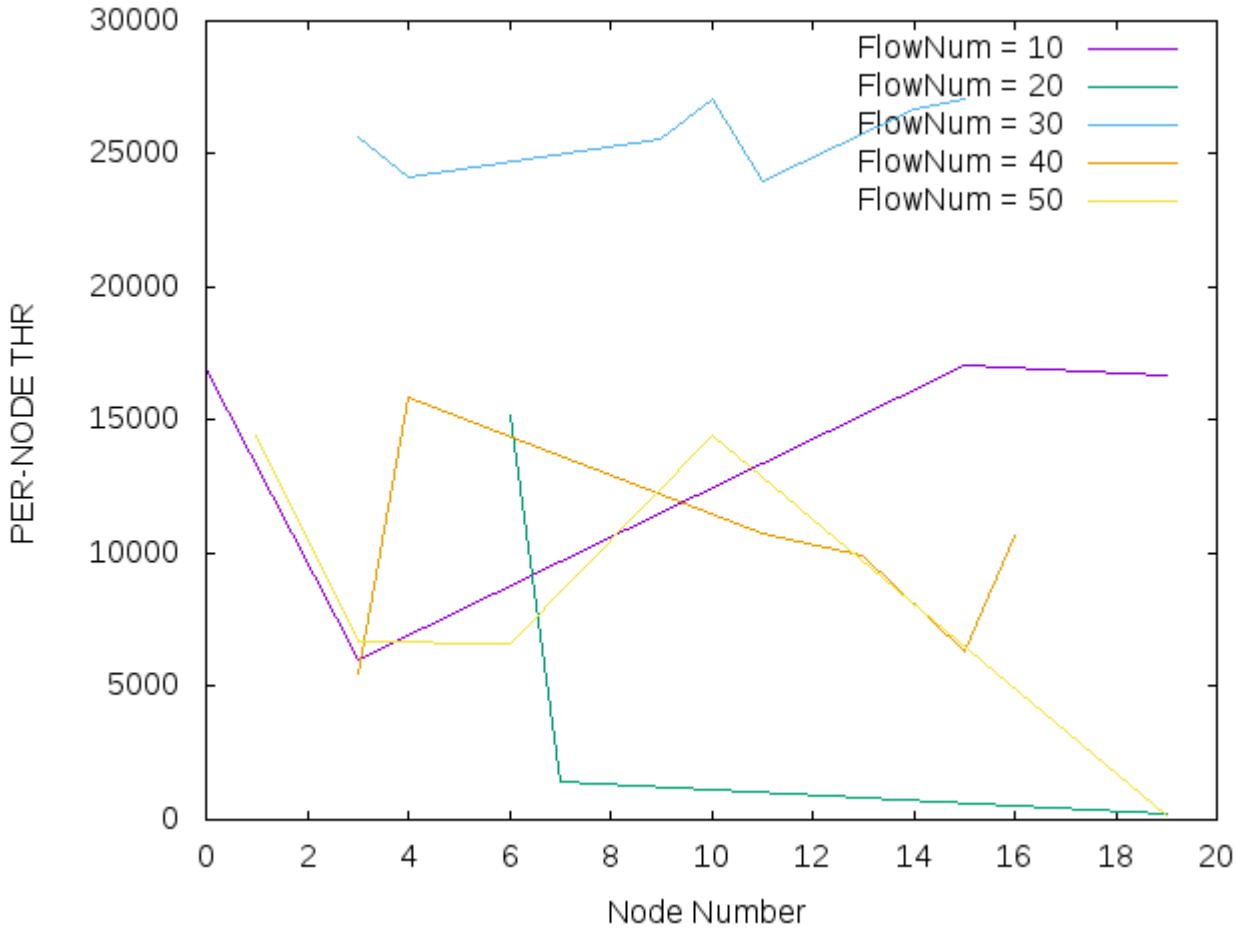

DelRatio VS NodeQueue
DelRatio
NodeQueue
99.4
99.2
99
98.8
98.6
98.4
98.2
98
97.8
97.6
97.4
20
40
60
80
100
120
140

7.2 Wireless Static

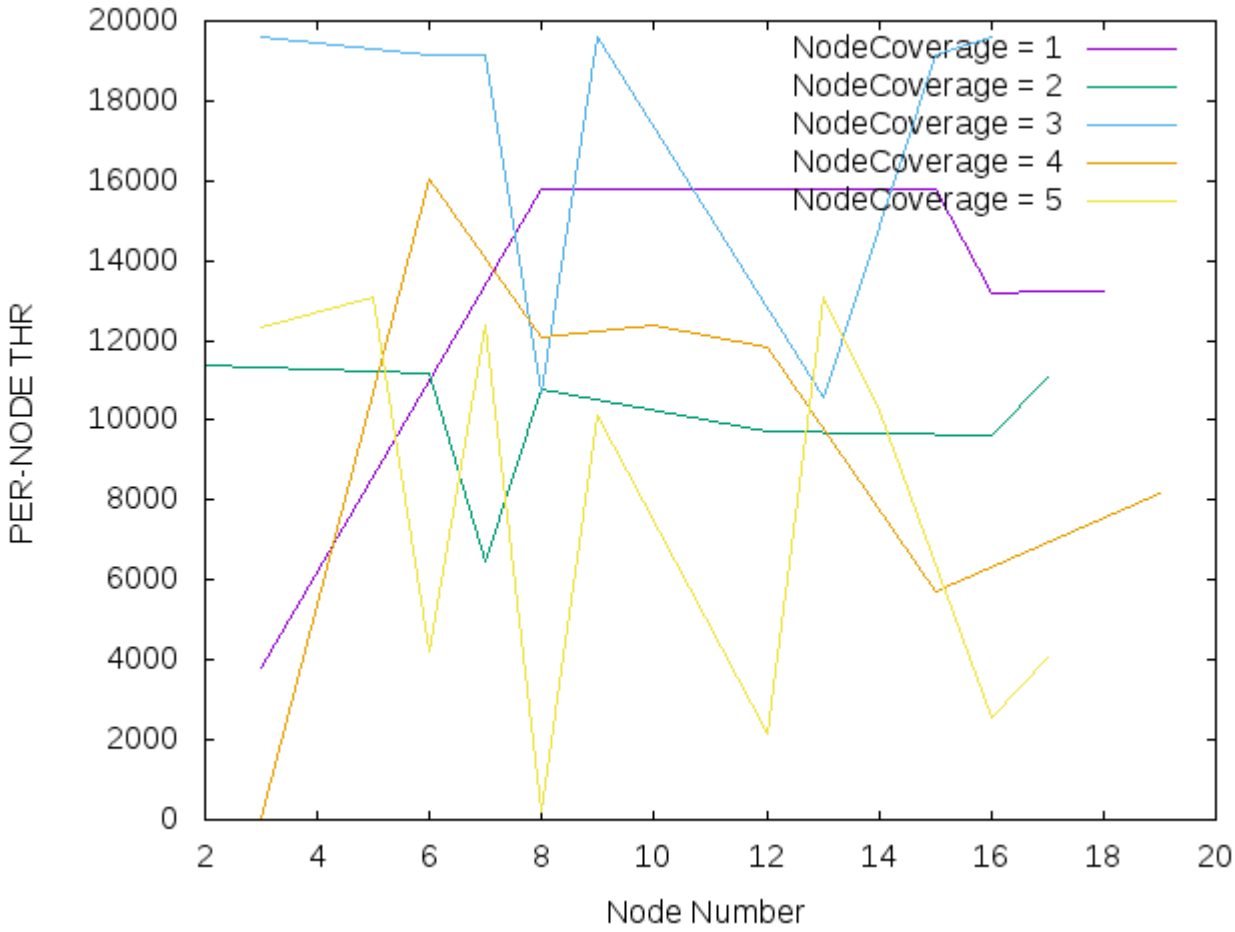
THR VS NodeNumS.png



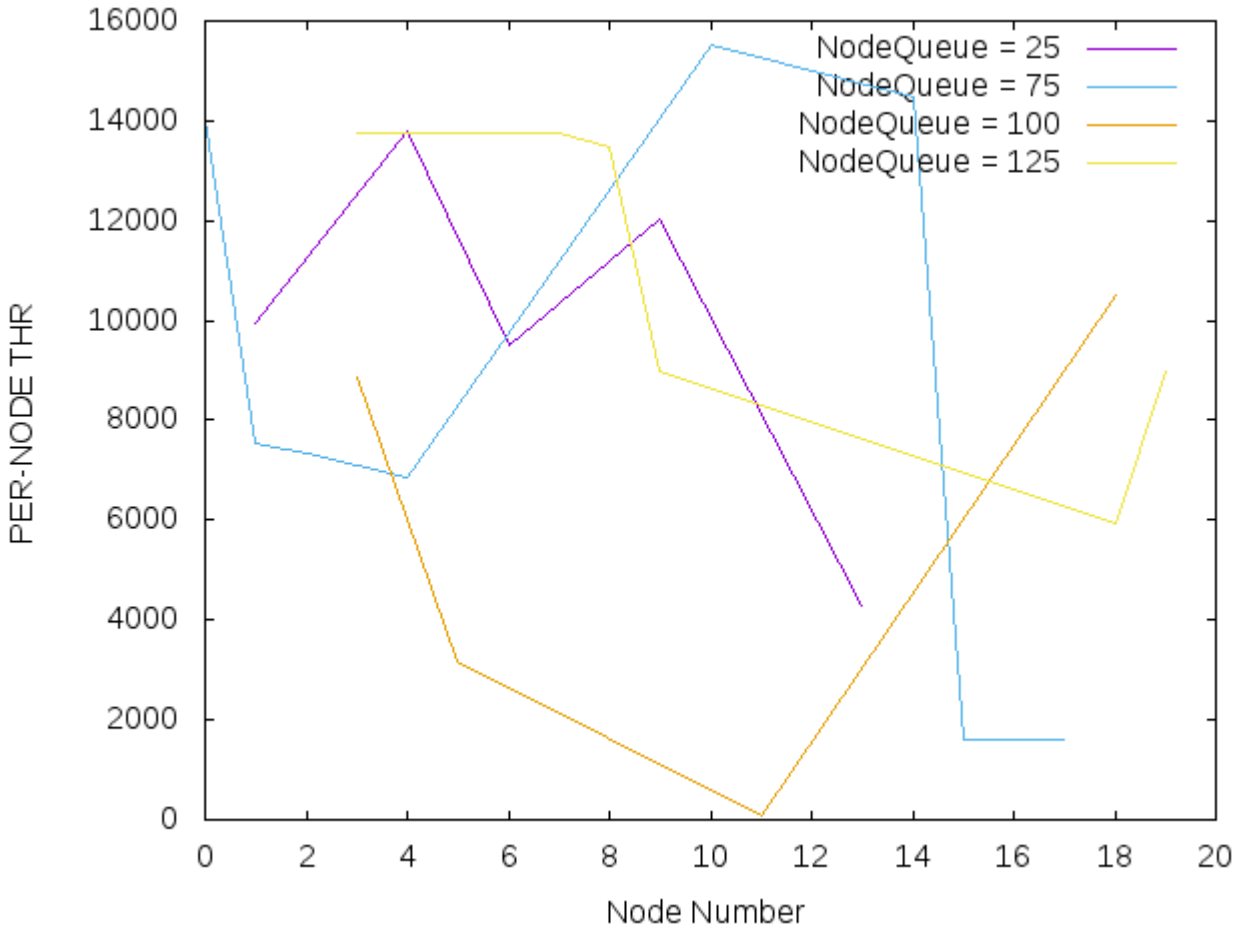
THR VS FlowNumS.png



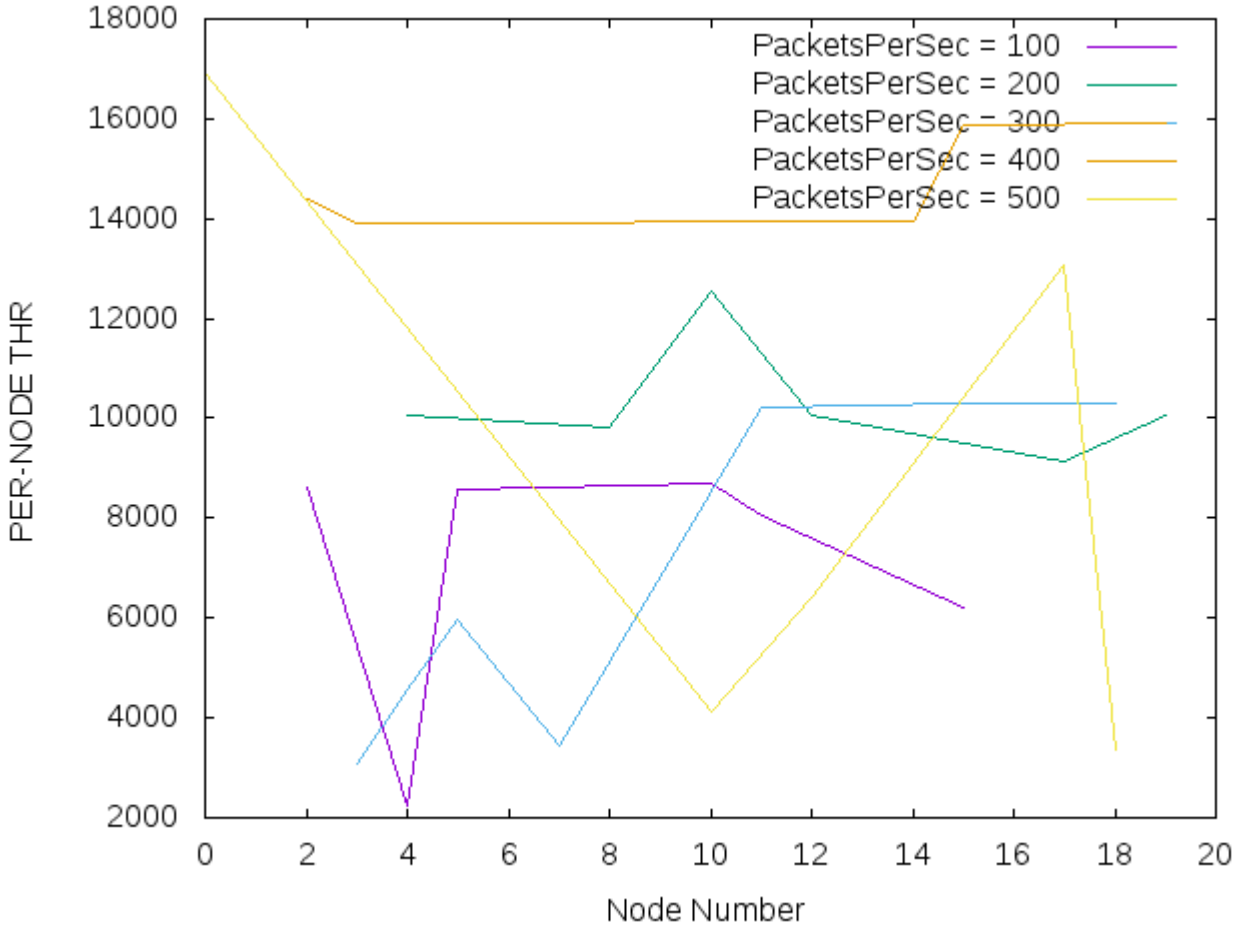
THR VS NodeCoverageS.png



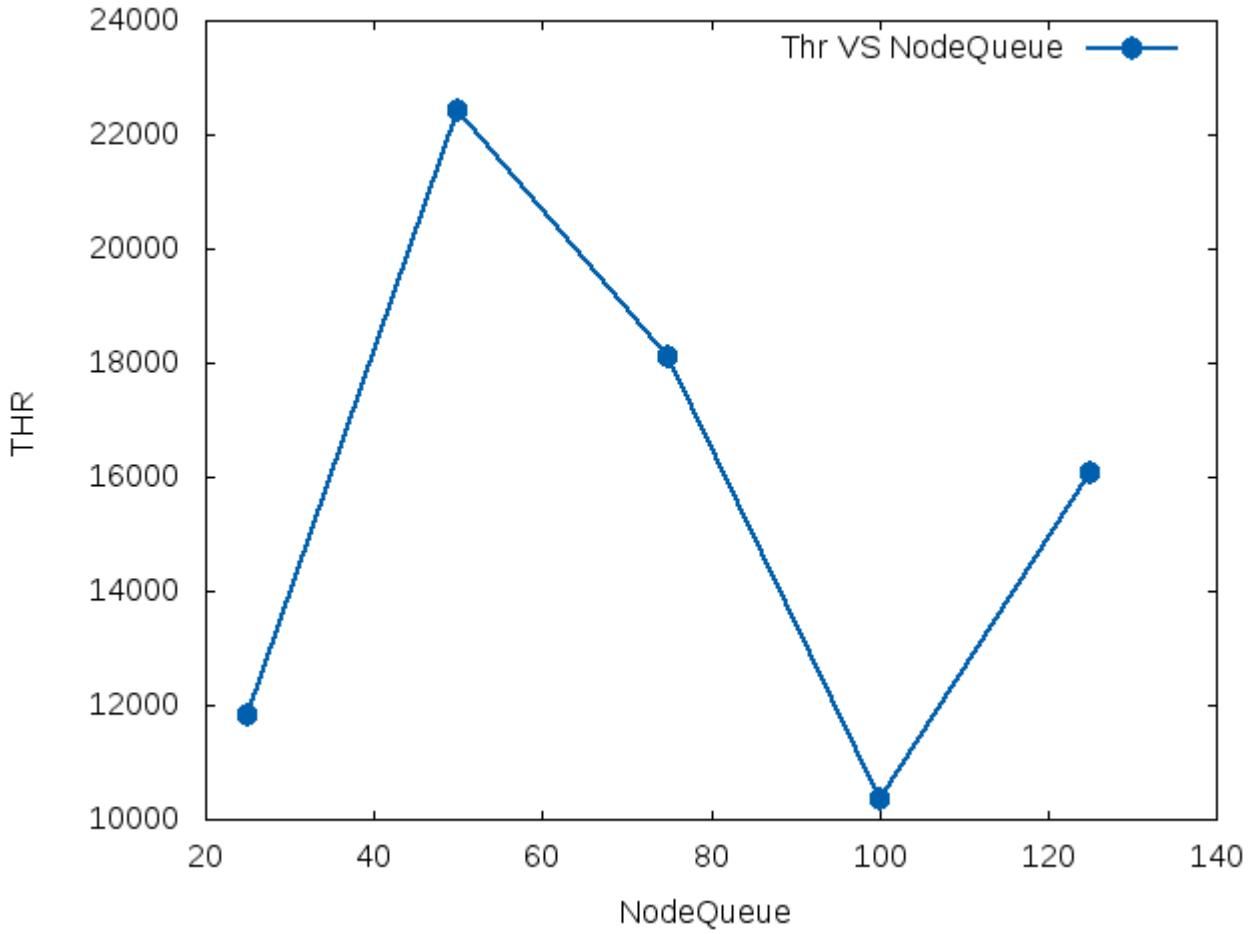
THR VS NodeQueueS.png



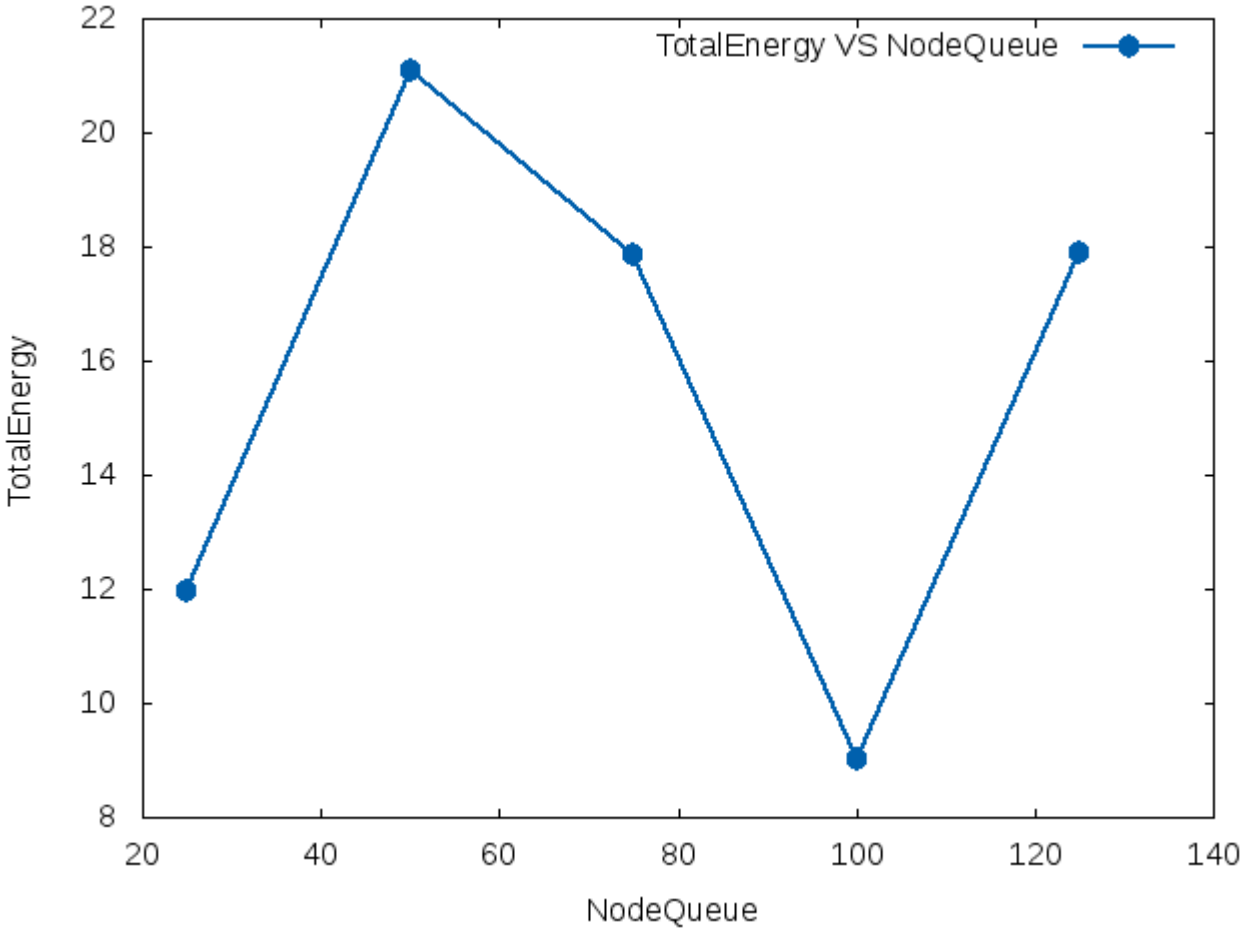
THR VS PacketsPerSecS.png



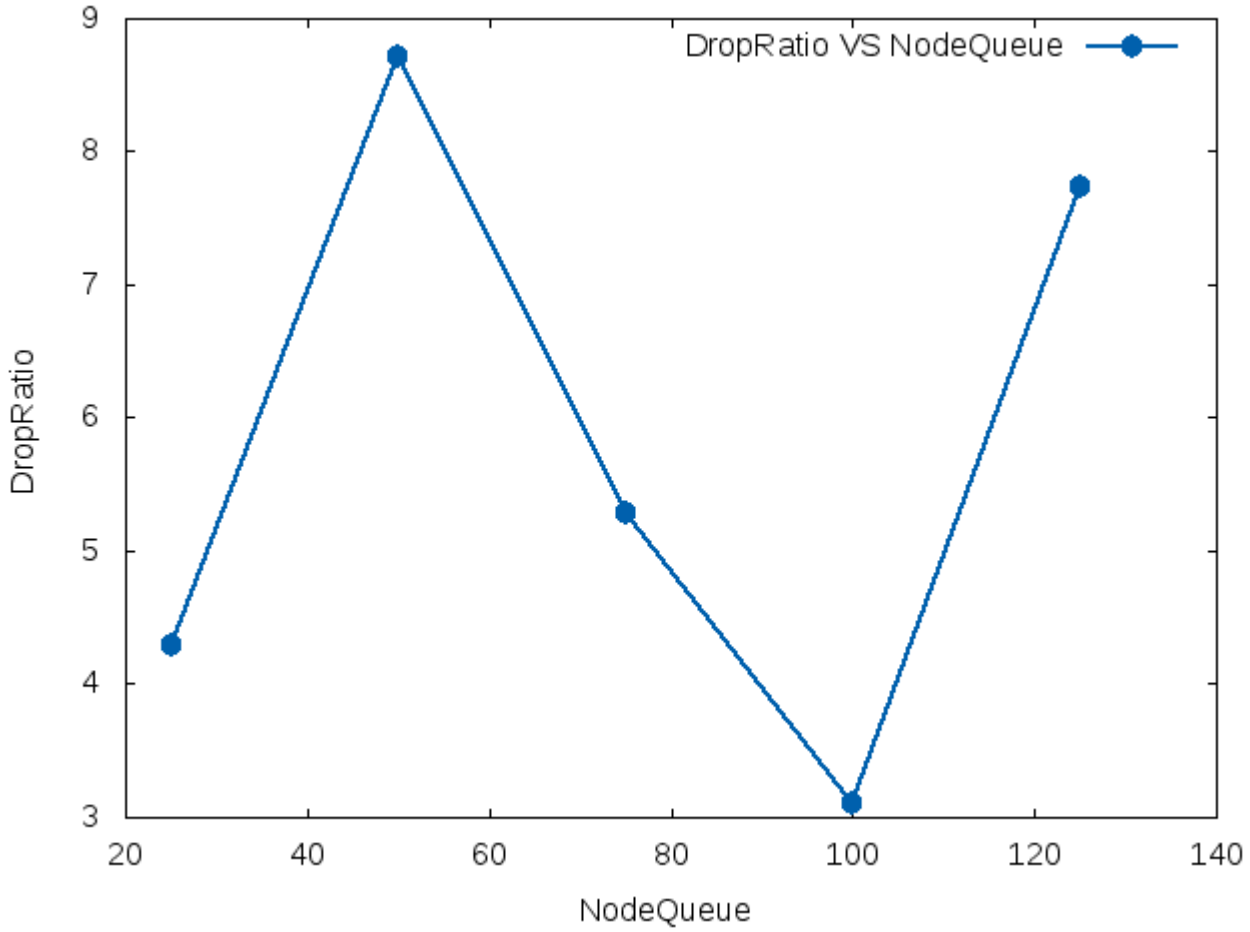
VS NodeQueueS.png



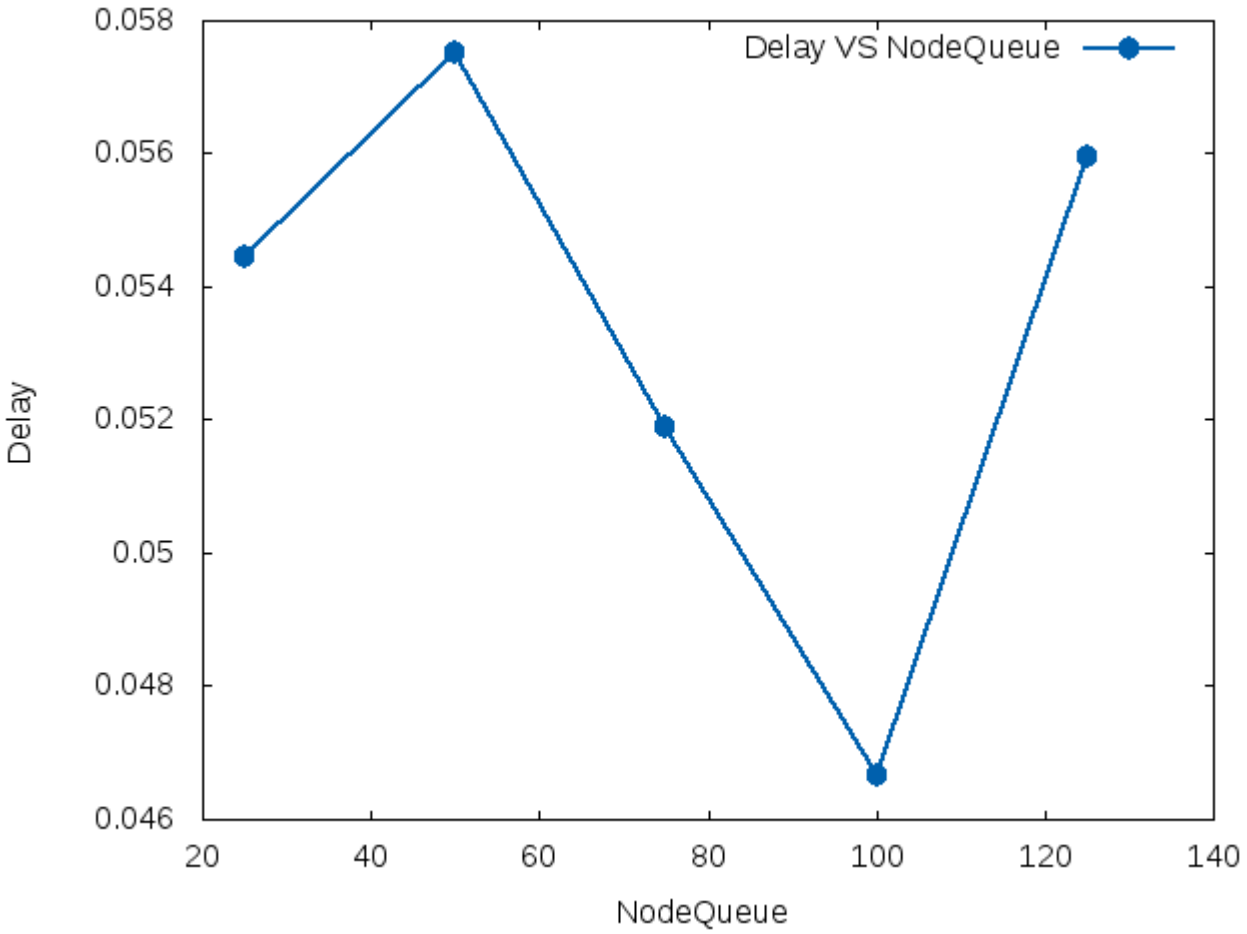
VS NodeQueueS.png



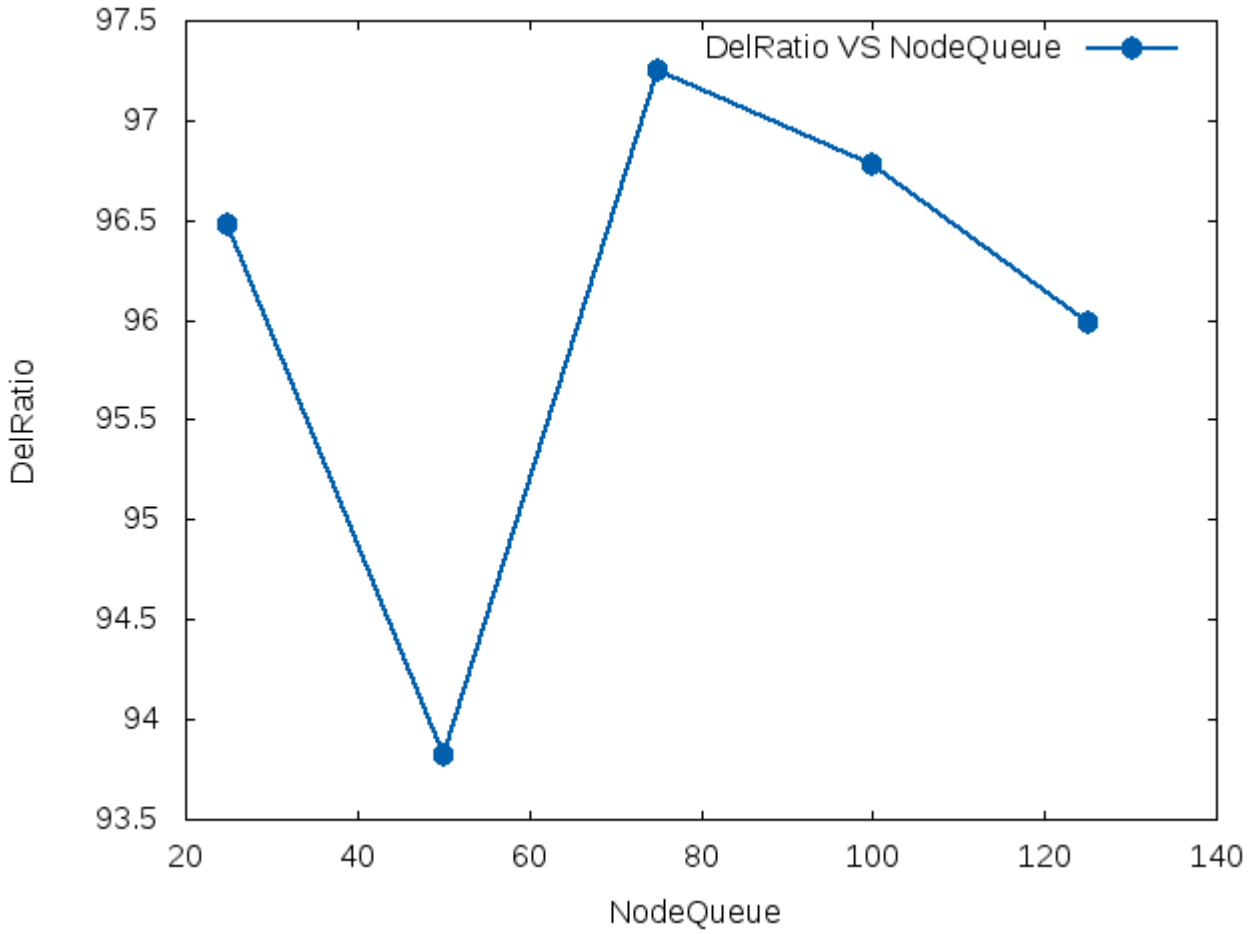
VS NodeQueueS.png



VS NodeQueueS.png

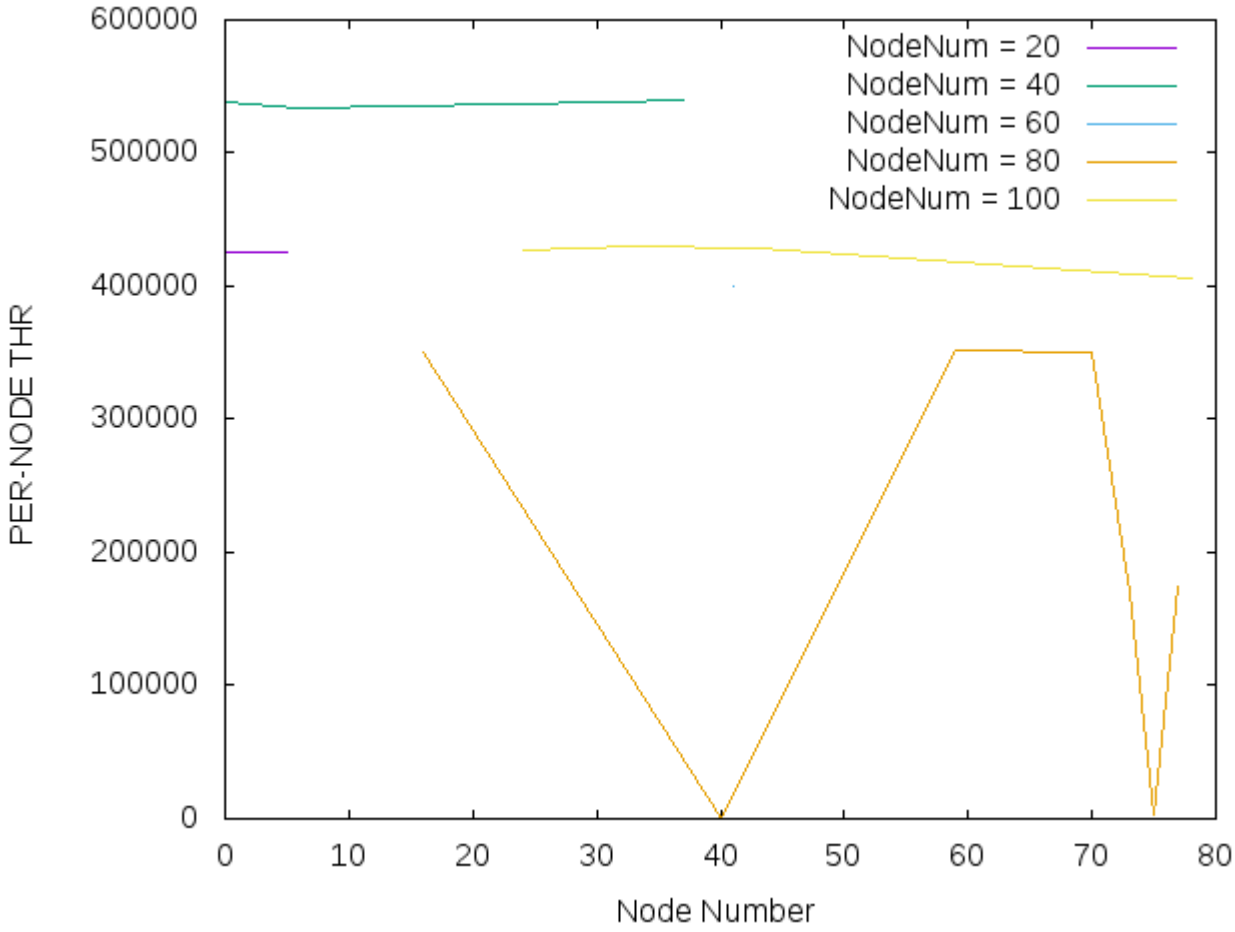


VS NodeQueueS.png

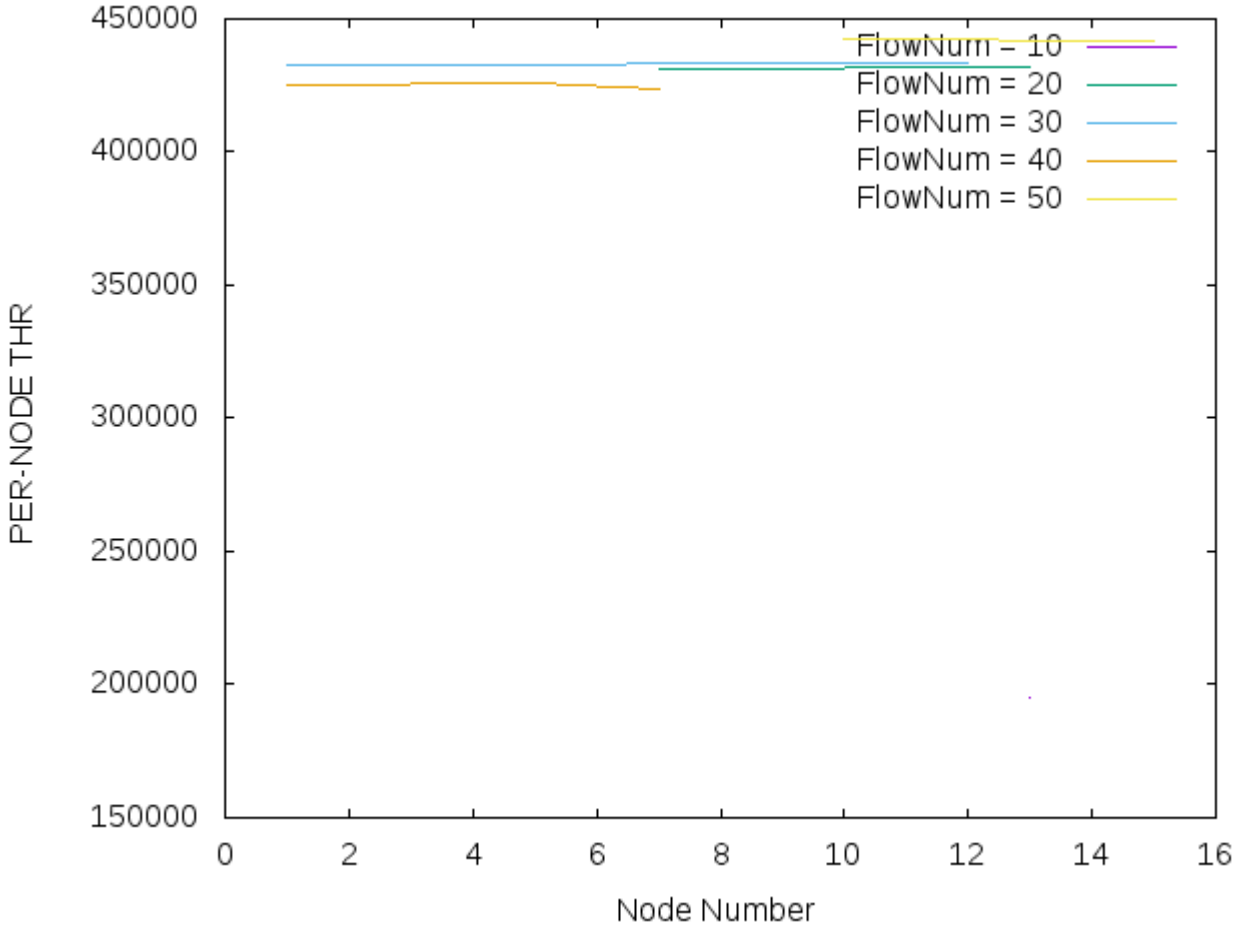

DelRatio VS NodeQueue
DelRatio
NodeQueue
97.5
97
96.5
96
95.5
95
94.5
94
93.5
20
40
60
80
100
120
140

7.3 Wireless Mobile Modified

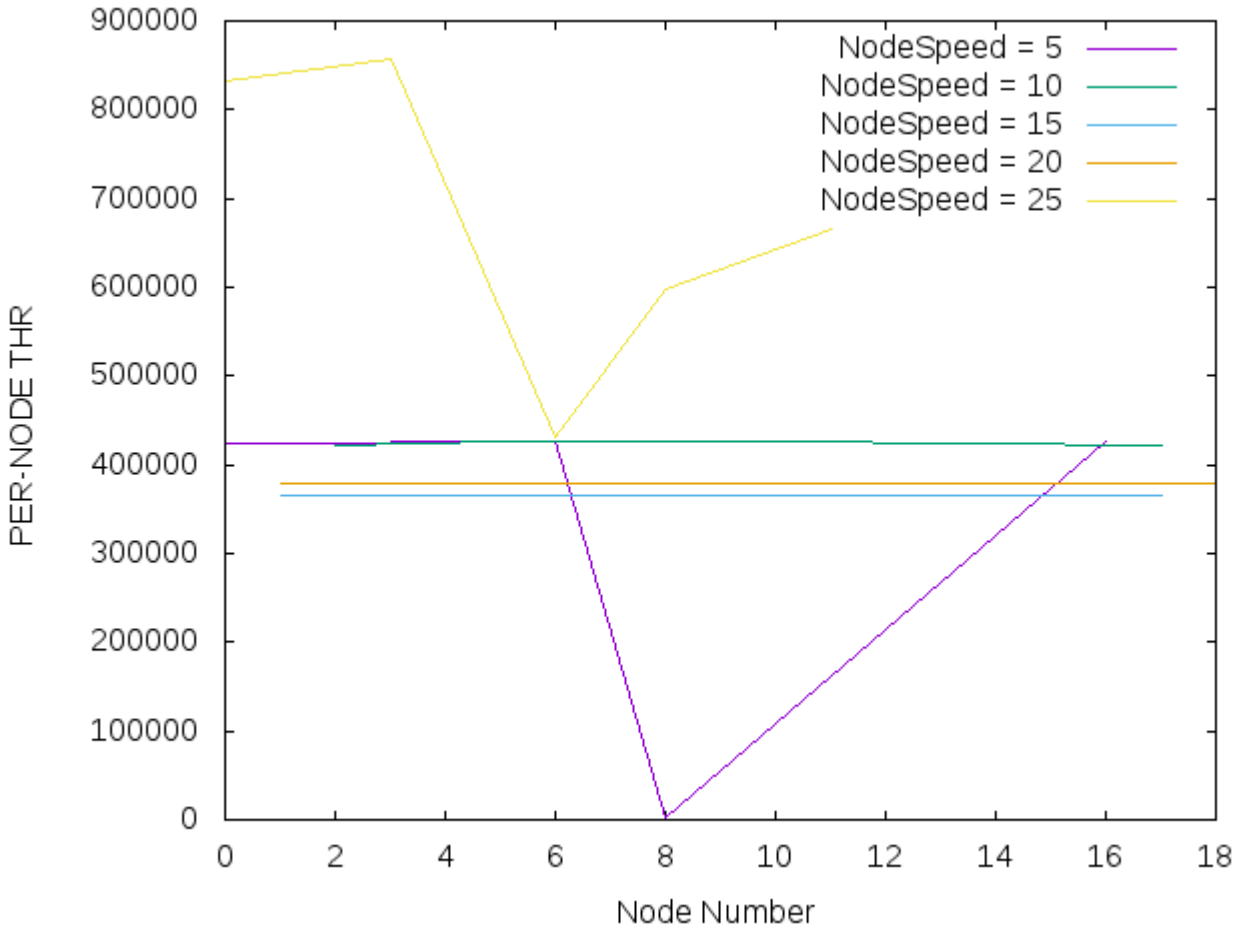
THR VS NodeNumMOD.png



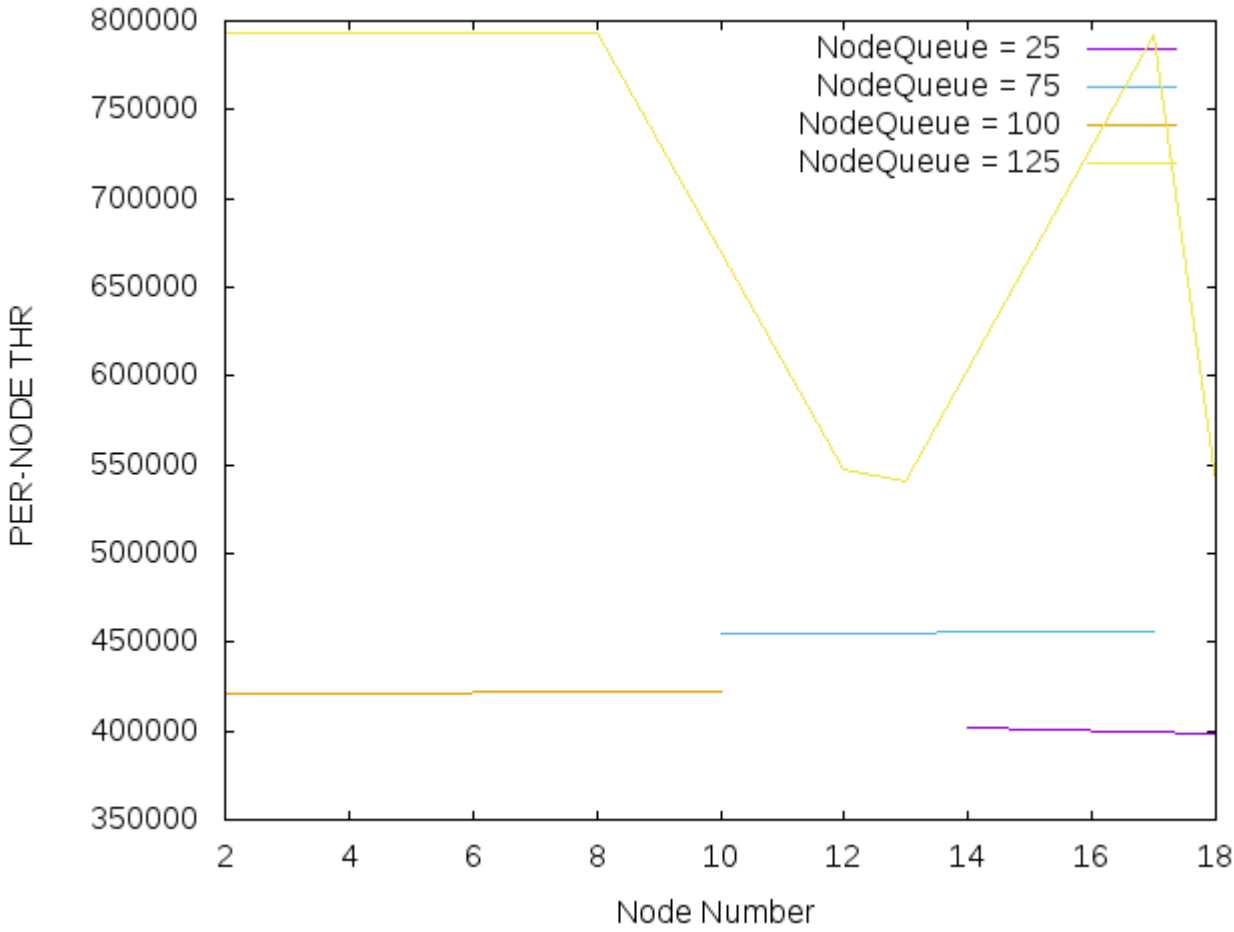
THR VS FlowNumMOD.png



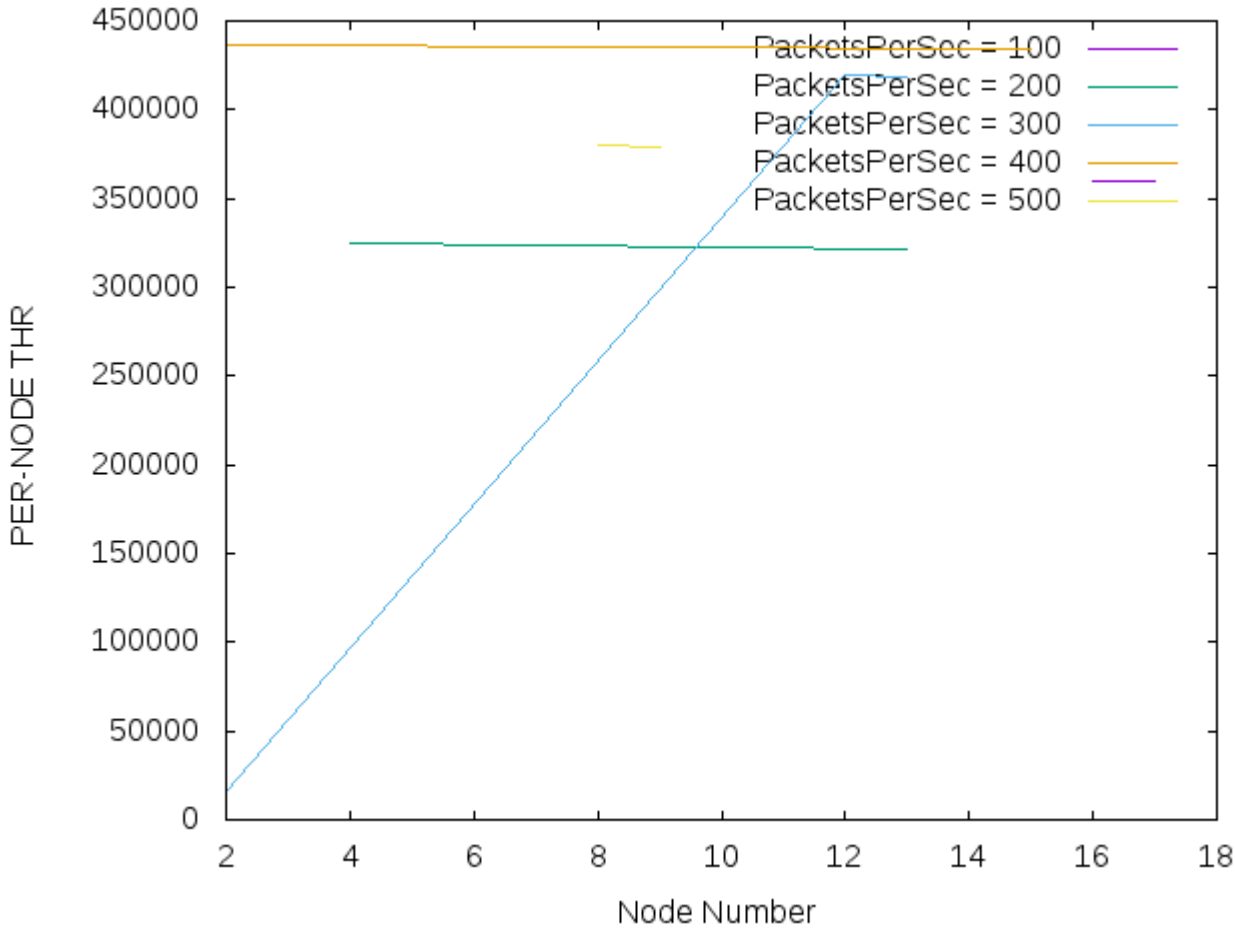
THR VS NodeSpeedMOD.png



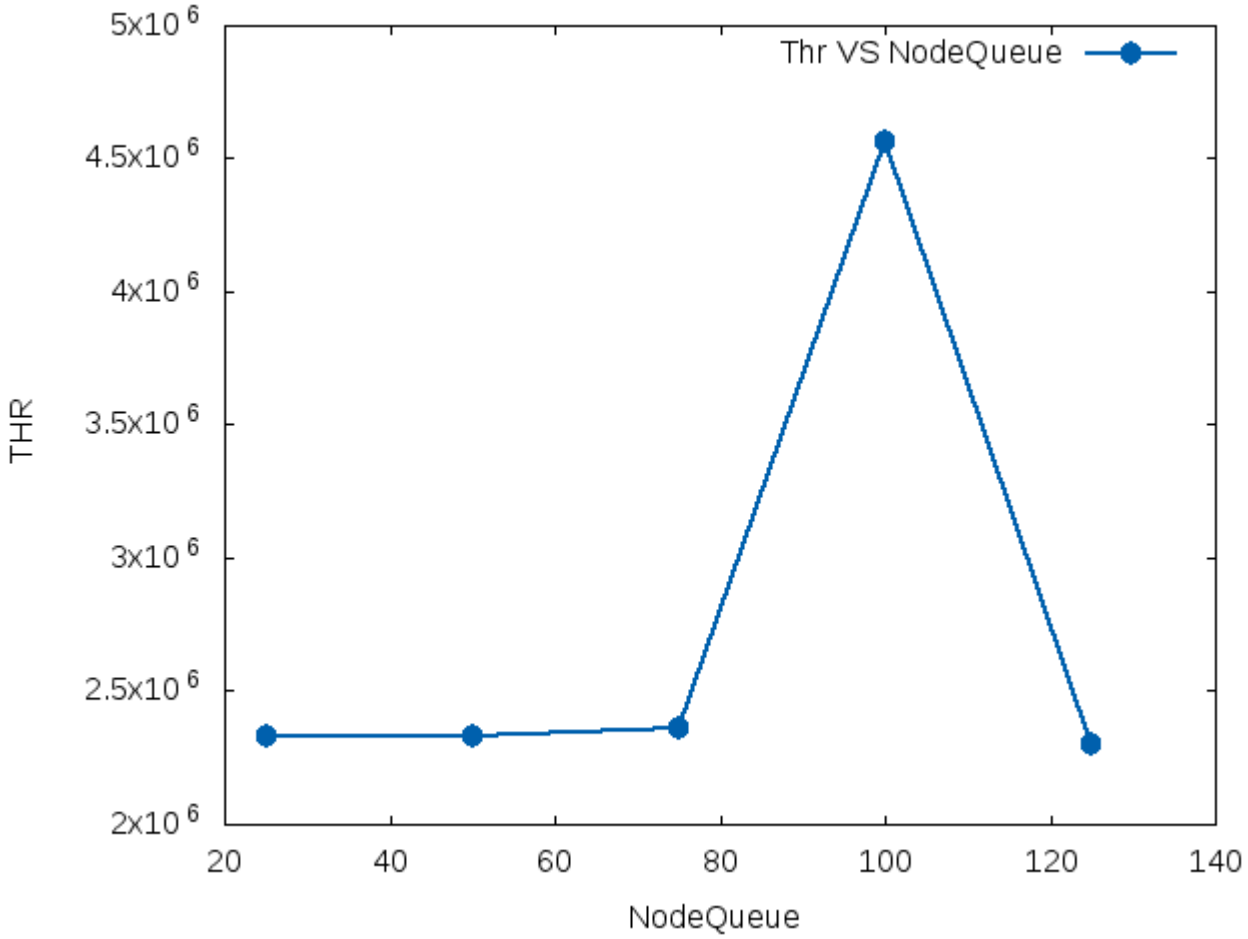
THR VS NodeQueueMOD.png



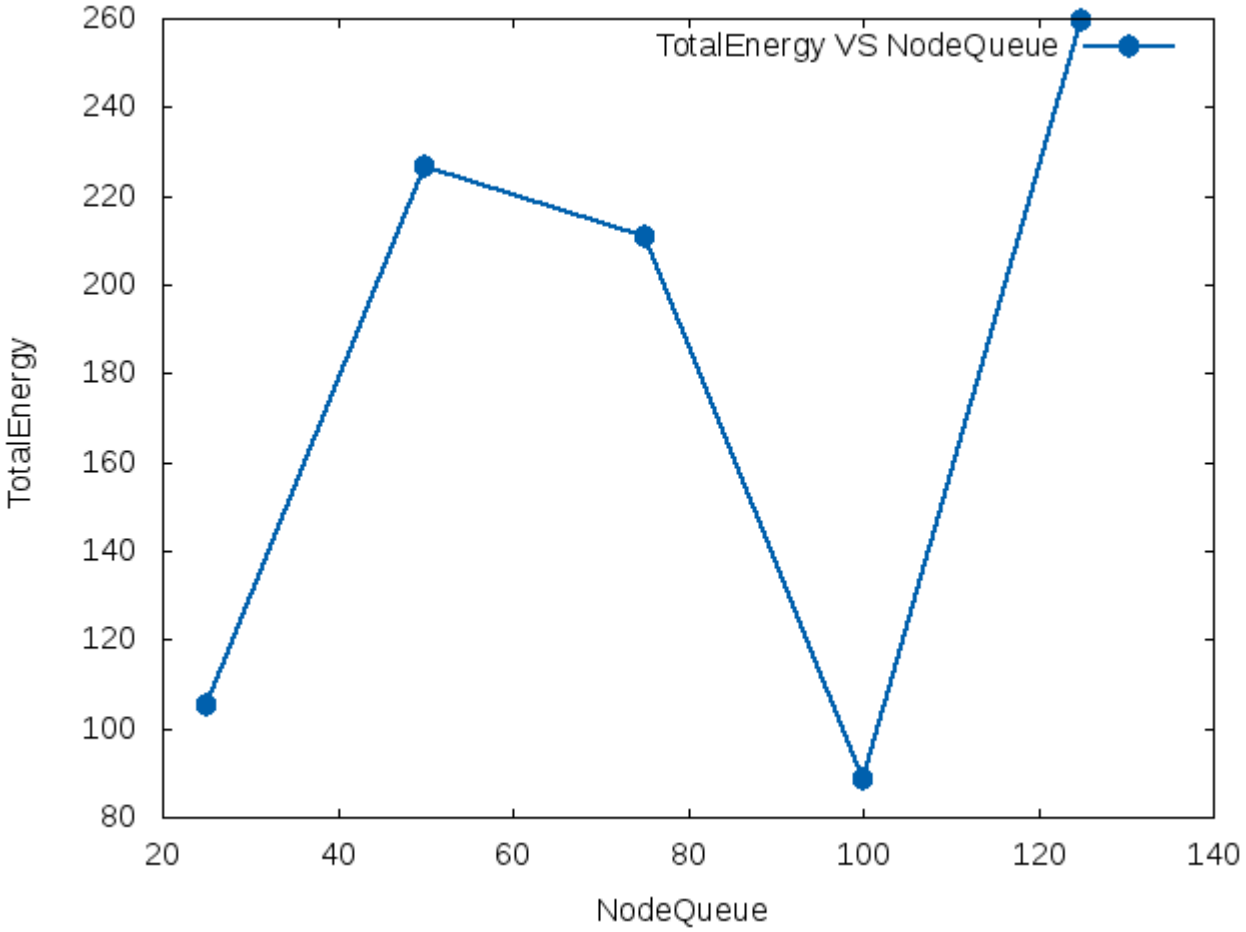
THR VS PacketsPerSecMOD.png



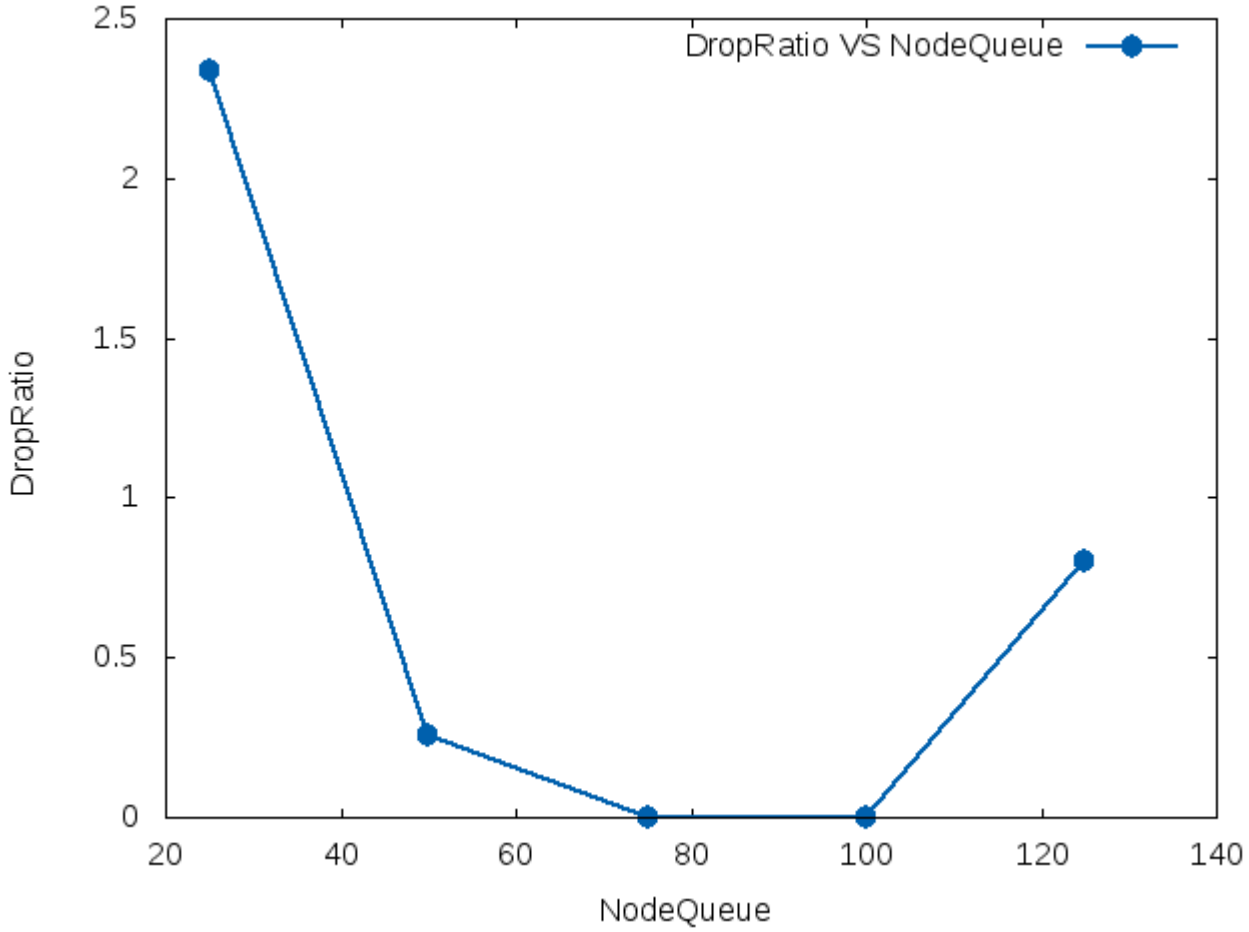
VS NodeQueueMOD.png



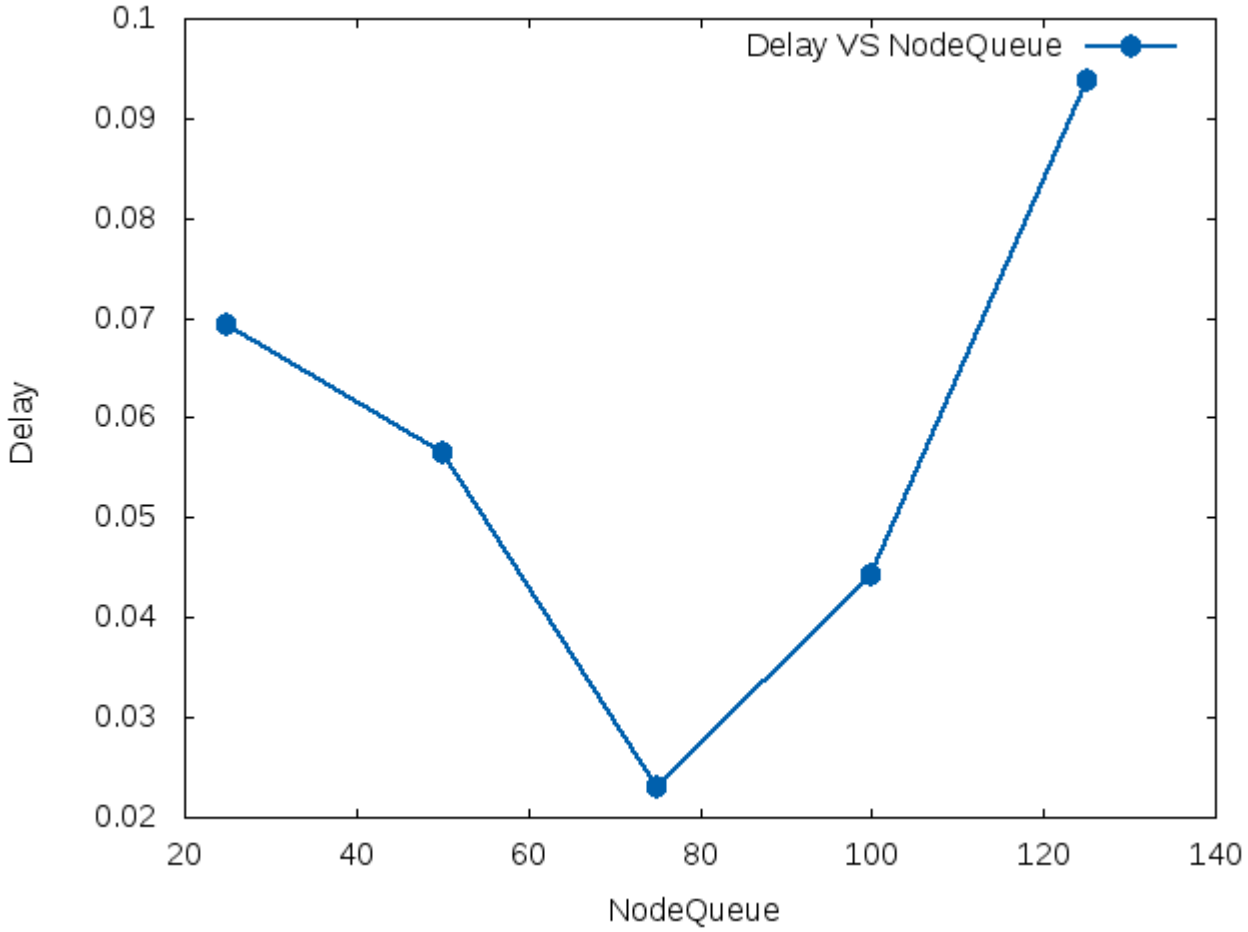
VS NodeQueueMOD.png



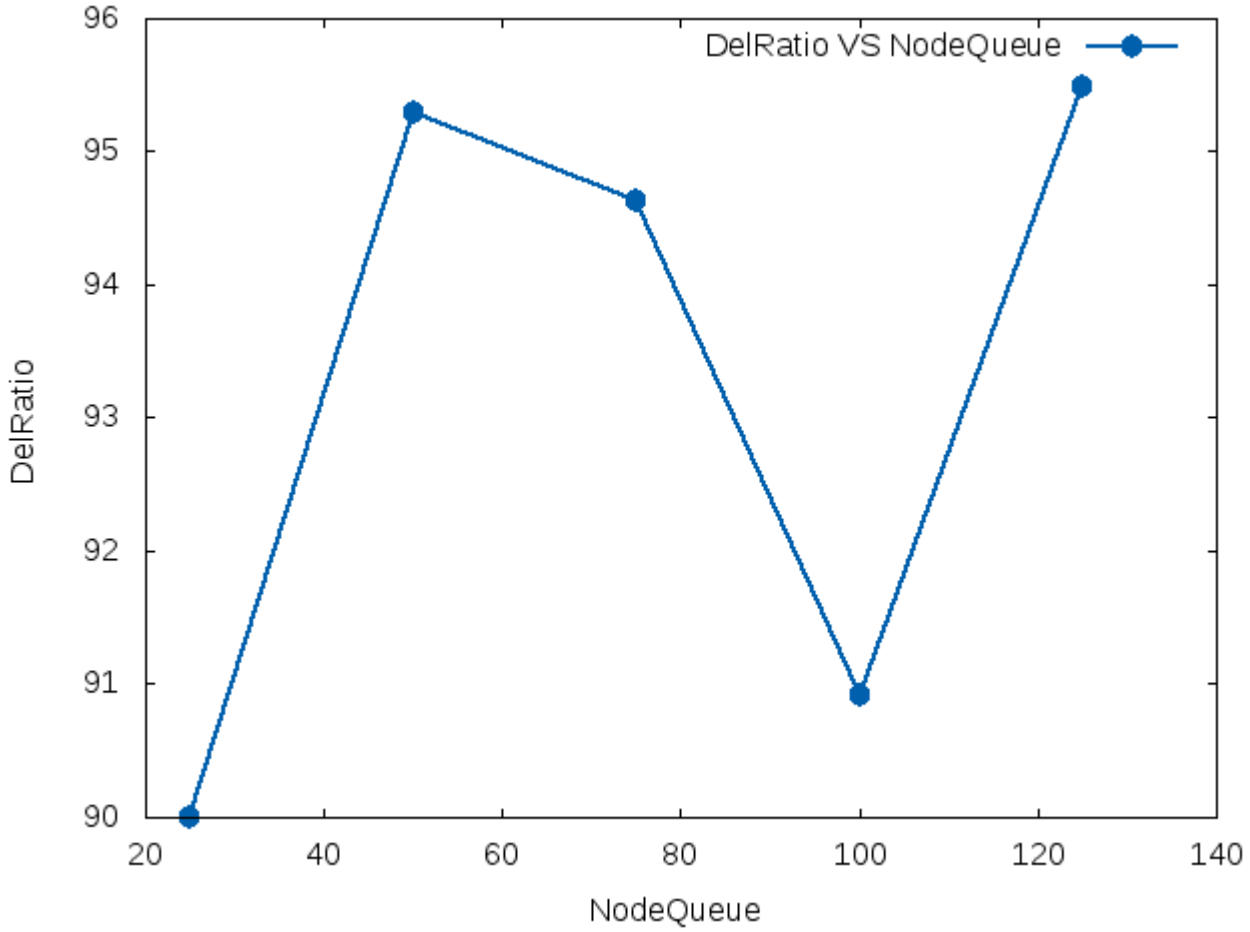
VS NodeQueueMOD.png



VS NodeQueueMOD.png

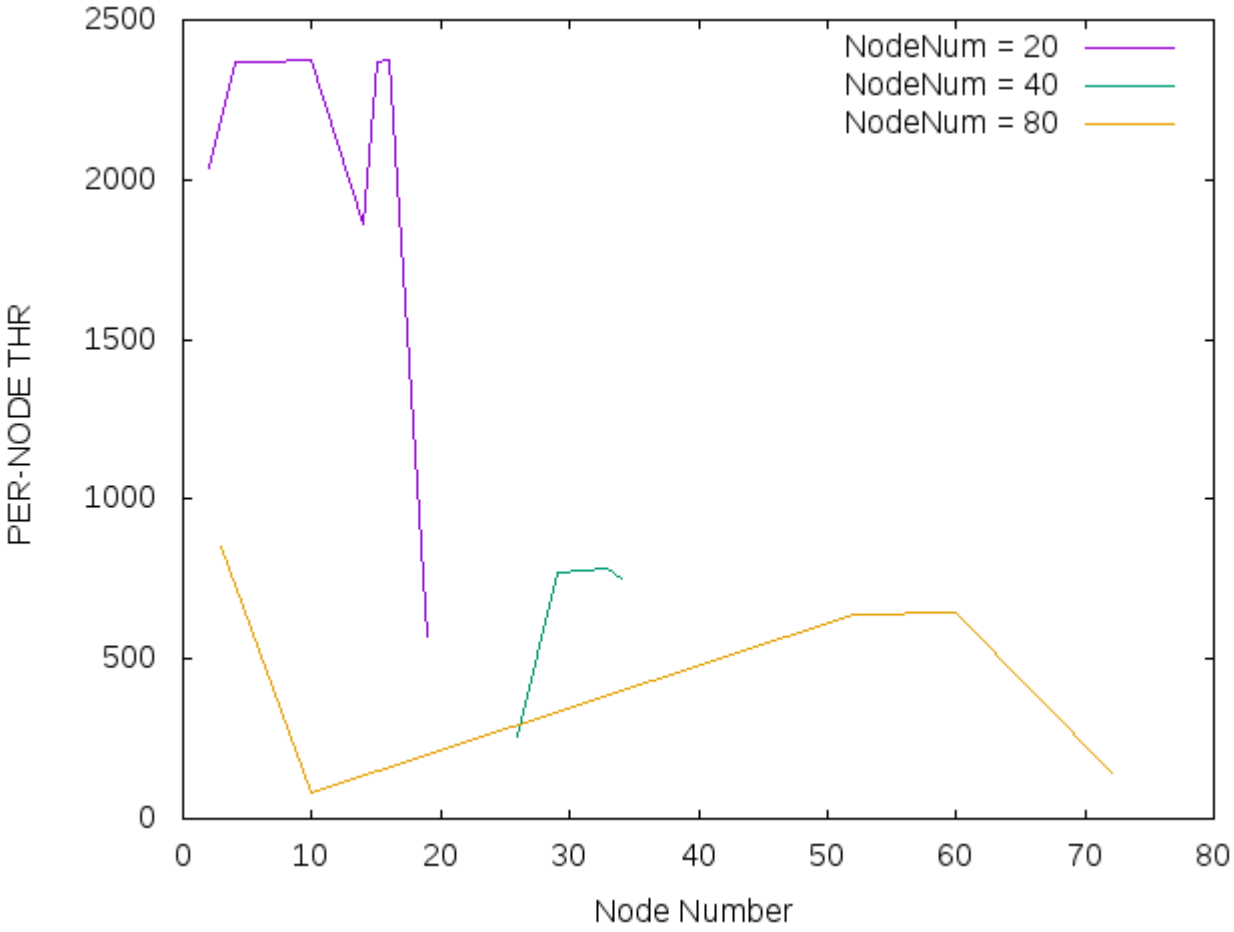


VS NodeQueueMOD.png

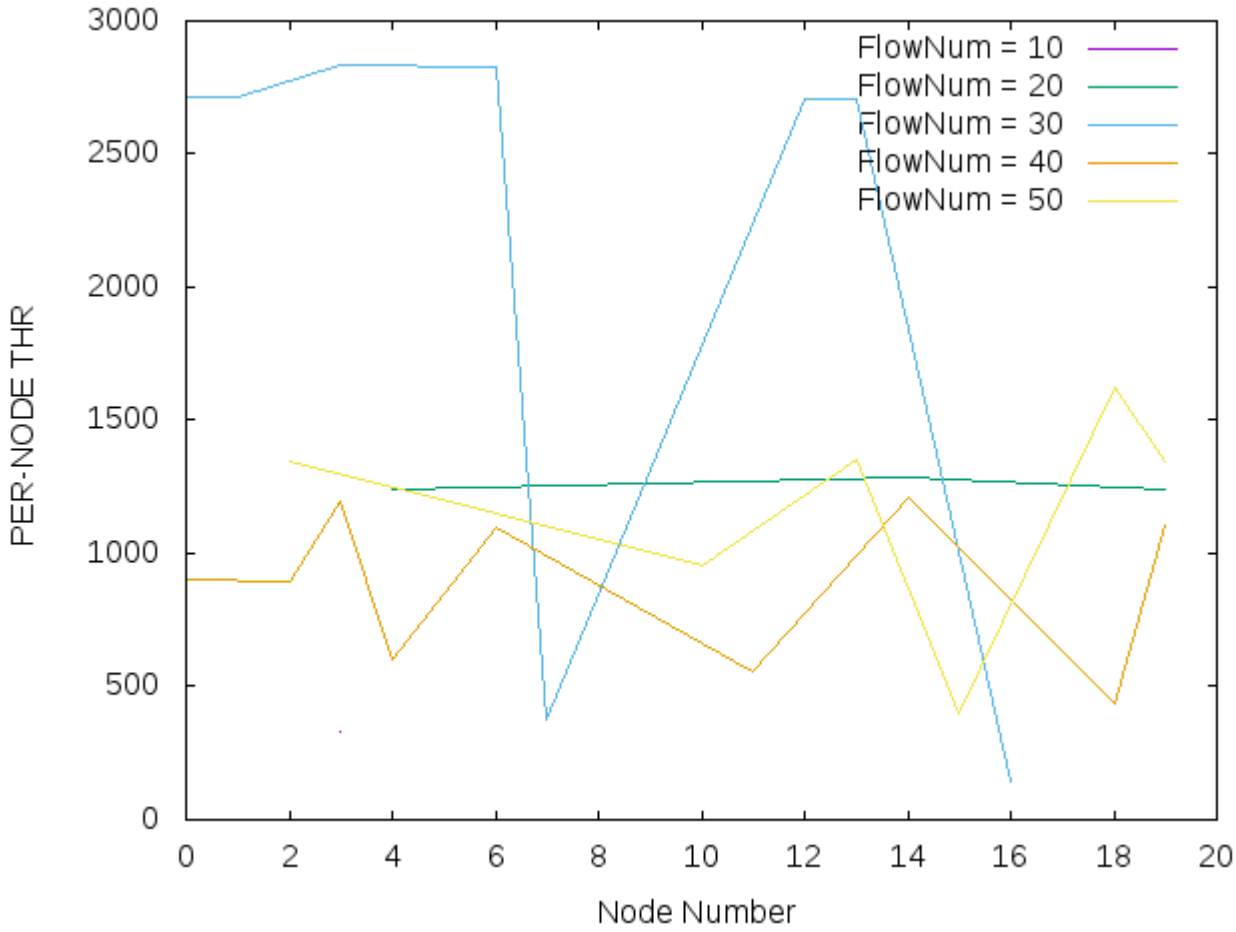

DelRatio VS NodeQueue
DelRatio
NodeQueue
96
95
94
93
92
91
90
20
40
60
80
100
120
140

7.4 Wireless Static Modified

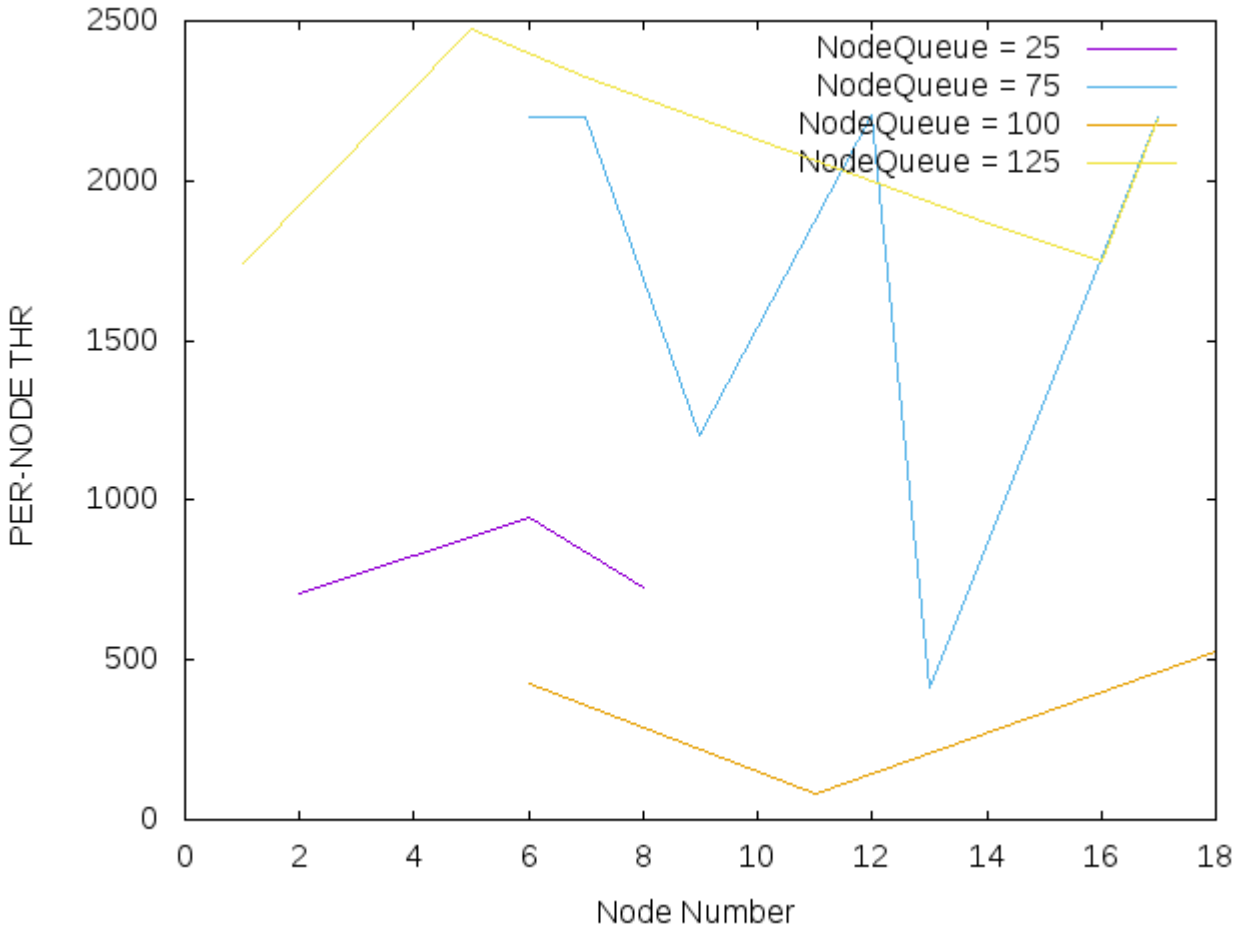
THR VS NodeNumMODS.png



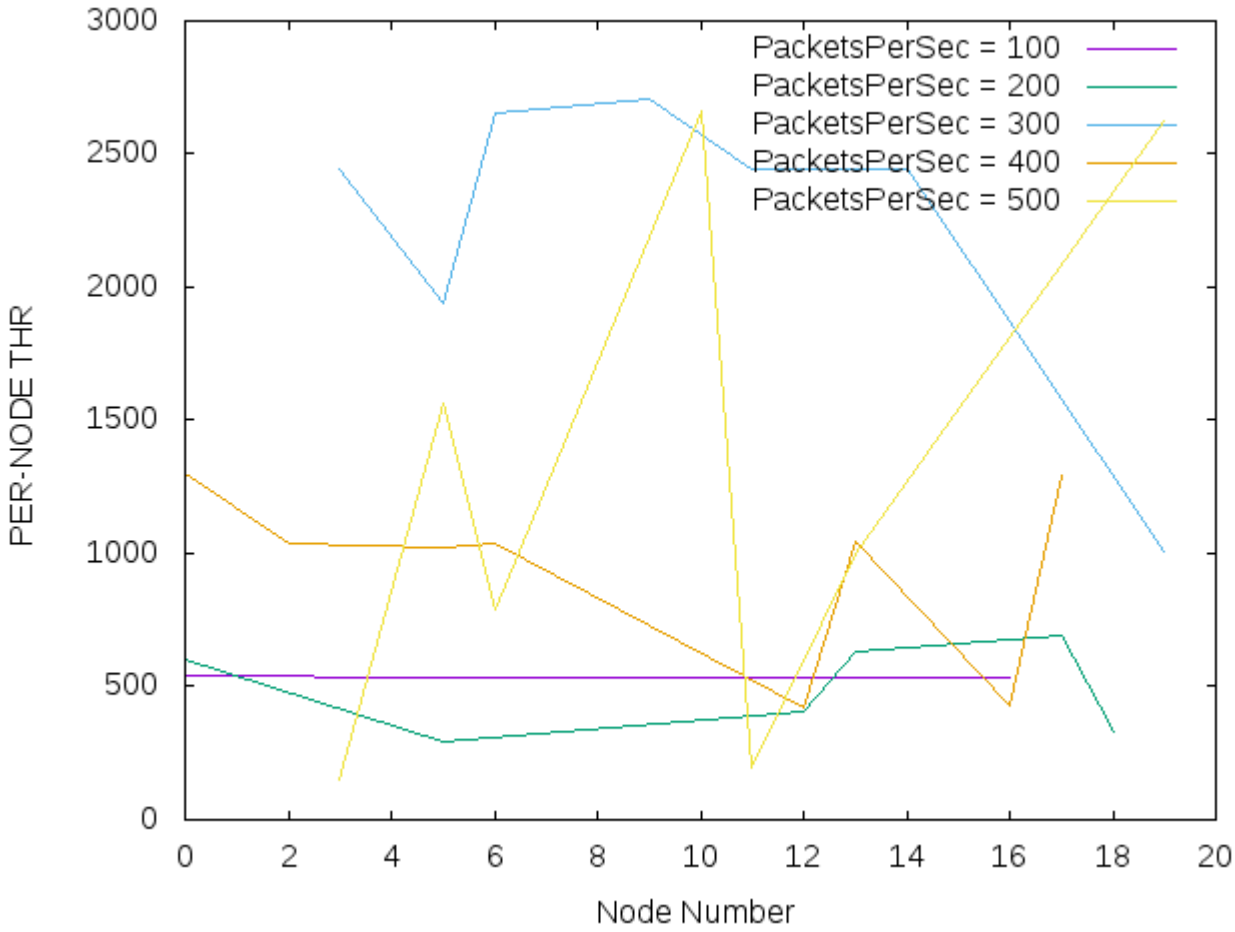
THR VS FlowNumMODS.png



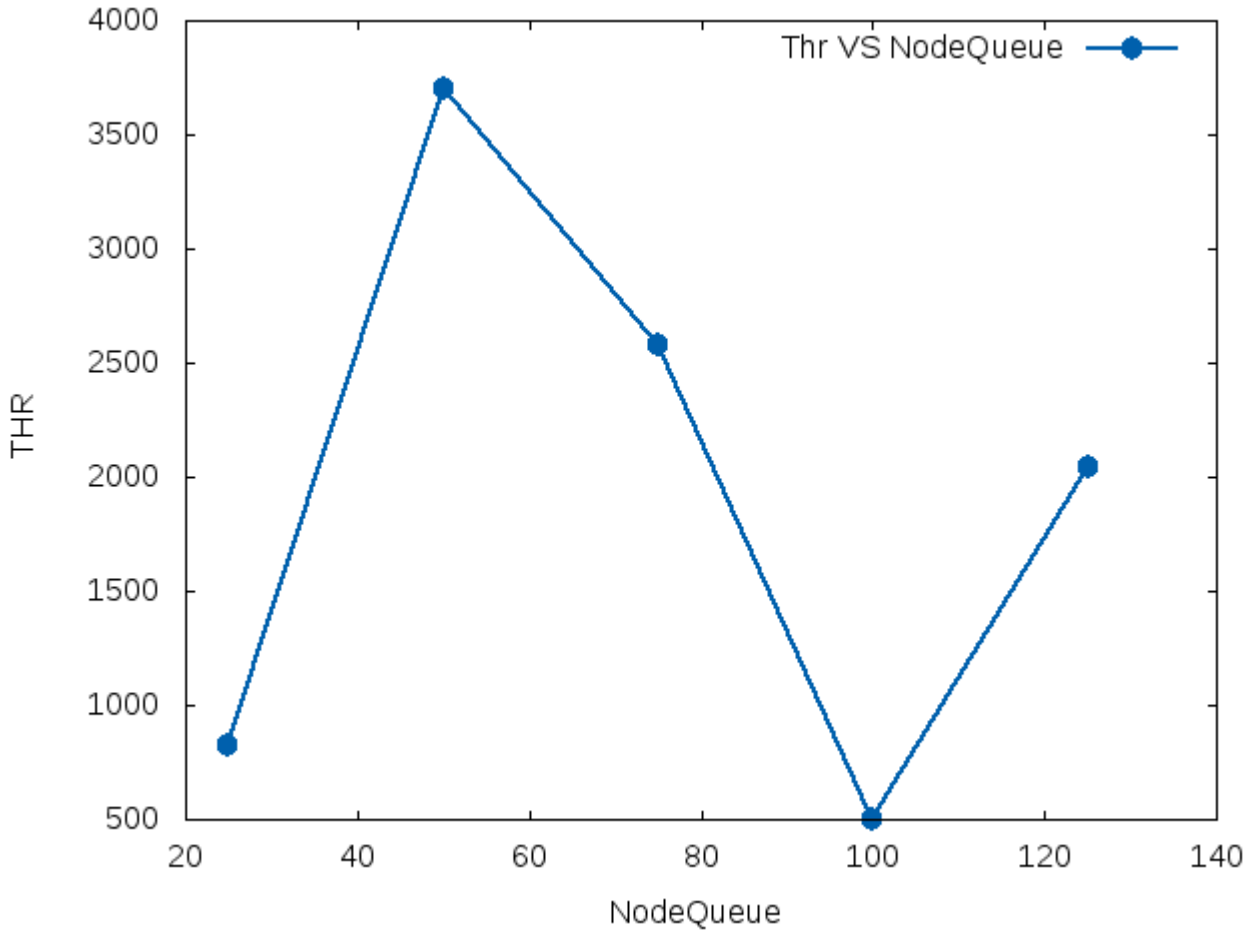
THR VS NodeQueueMODS.png



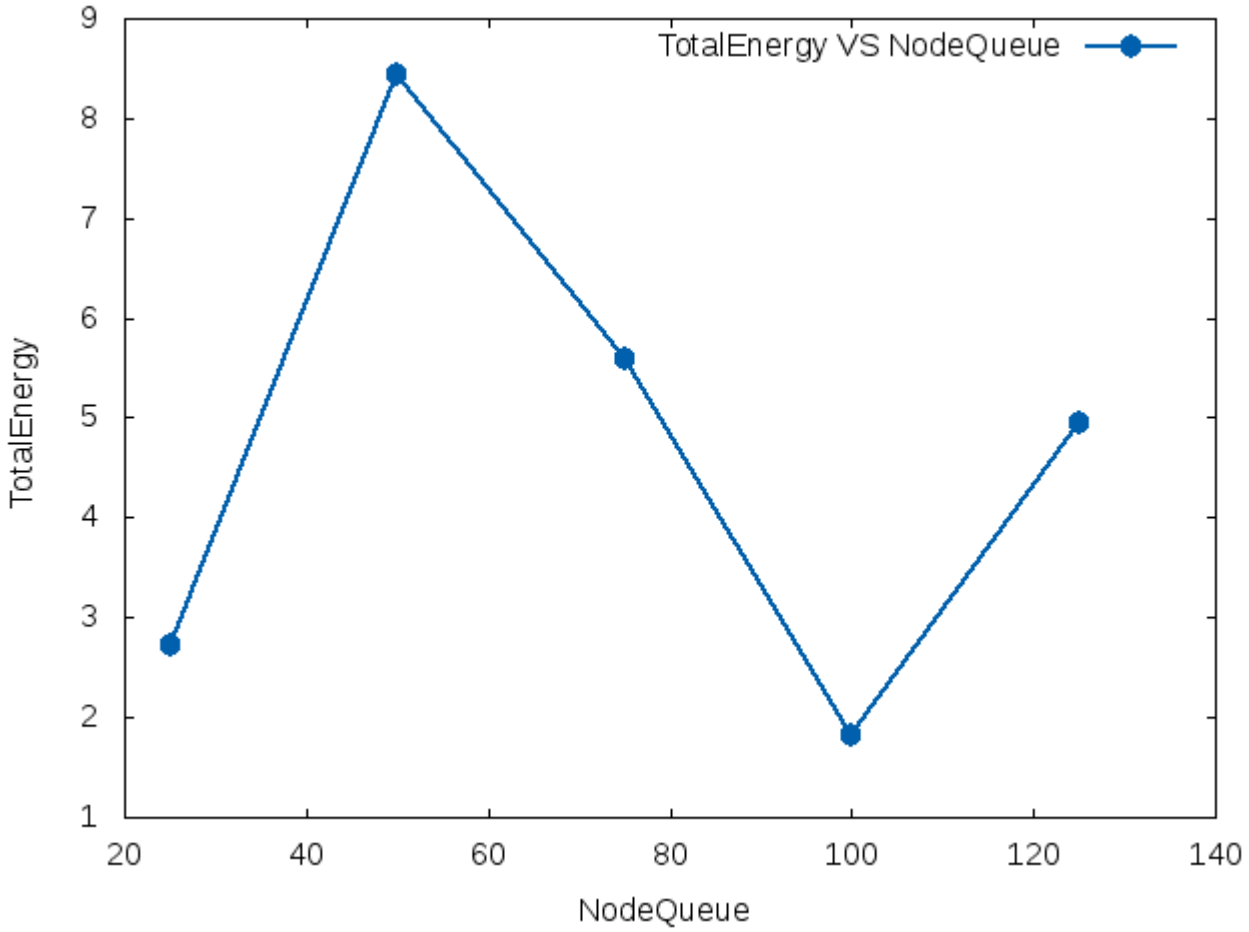
THR VS PacketsPerSecMODS.png



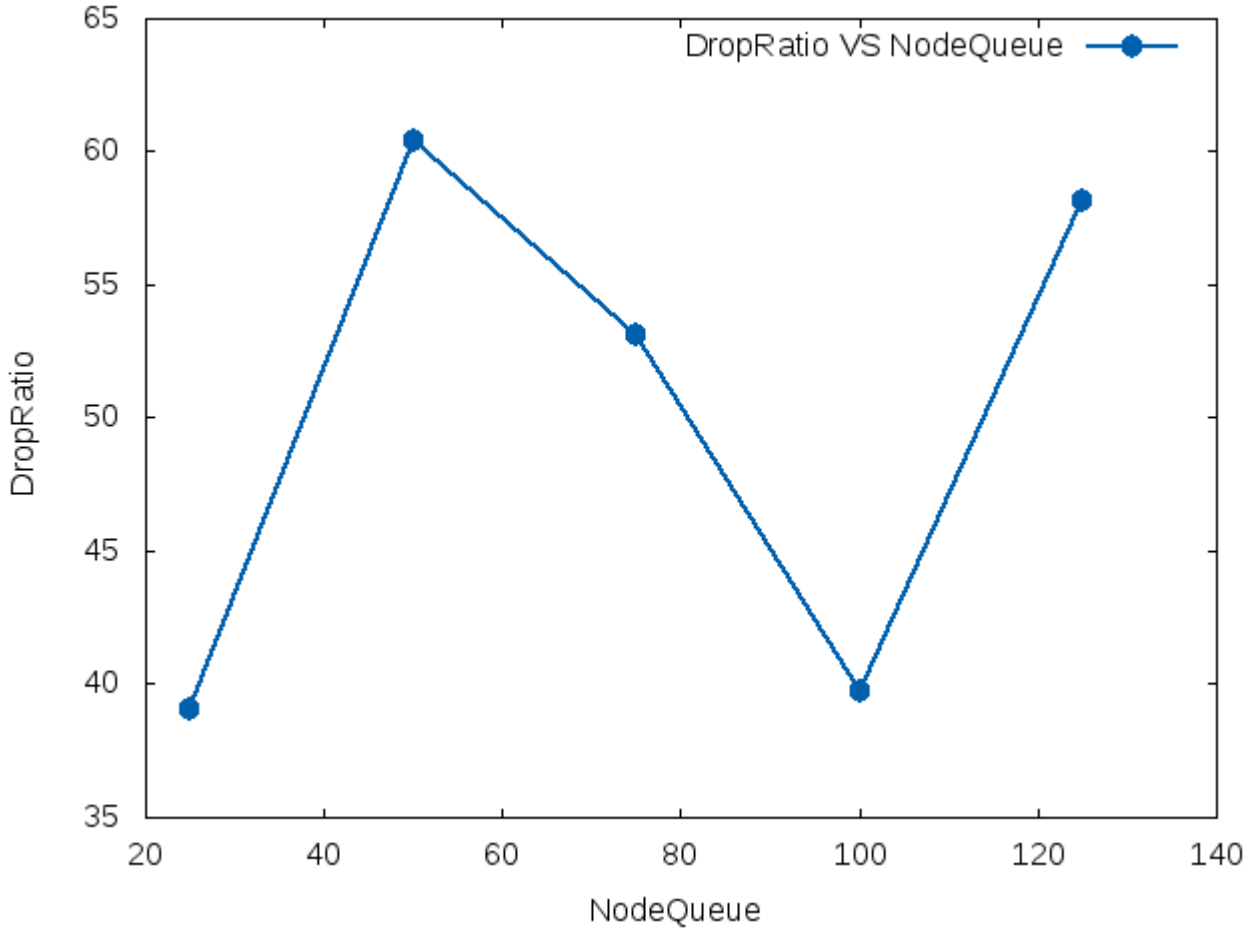
VS NodeQueueMODS.png



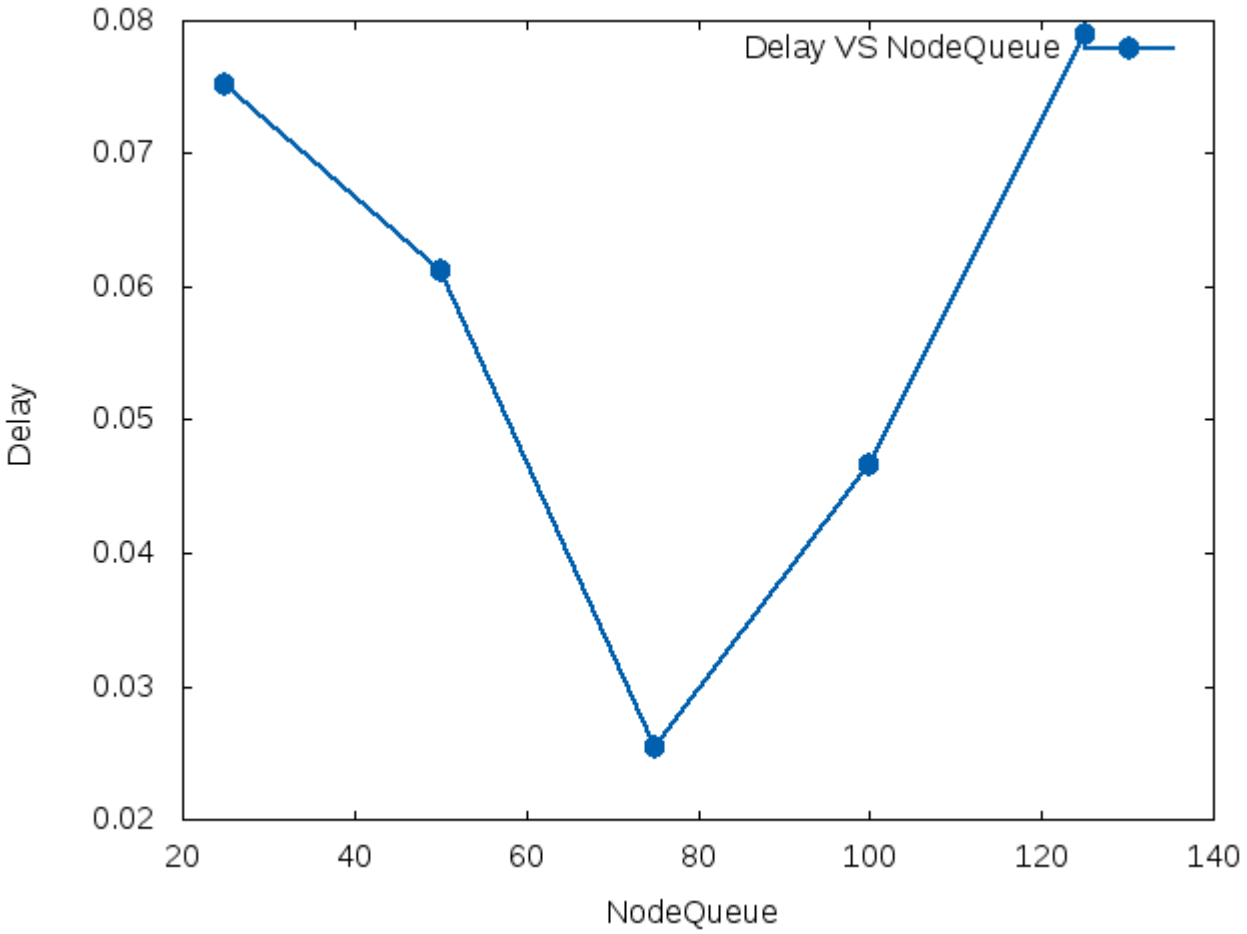
VS NodeQueueMODS.png



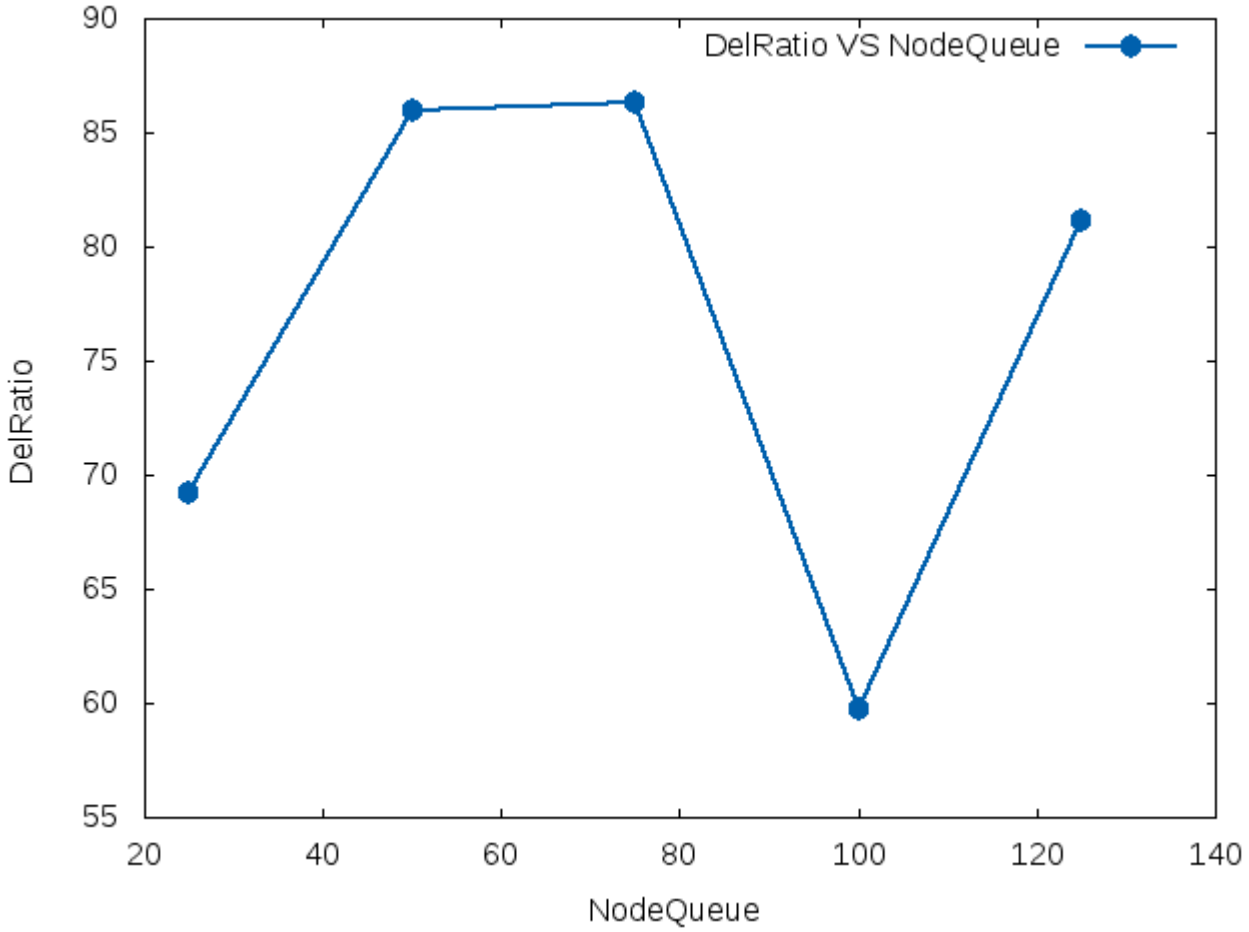
VS NodeQueueMODS.png



VS NodeQueueMODS.png



VS NodeQueueMODS.png


DelRatio VS NodeQueue
DelRatio
NodeQueue
90
85
80
75
70
65
60
55
20
40
60
80
100
120
140

8 Crossed transmission of packets

For Cross transmission of packets, i.e., transmission of packets from one type of node to that of another type (example: simulating a flow n1-¿n2-¿...-¿nx, where the first few nodes are wired and the rest nodes are wireless), we have 1 base node, 1 wired sink node, 1 wireless (802.11) sink node , 2 wired source node and 1 wireless (802.11) source node.Here are the output of the wired wireless simulaton .

8.1 Table of metric

Metric Name	value
Throughput	1737468.00
Average Delay	7.50454
Sent packets	9690.00
Receive packets	8375.00
Drop packets	16.00
Delivery ratio	86.43
Drop ratio	0.17
Time	40.09973
Total transmit	4945
Throughput 1	870182.38863
Throughput 2	867285.61112

9 WiMax 802.16

Here are 2 red nodes that are routers , 1 green node that is sink and 2 black nodes that are base stations.Here are the output of wimax simulation.

9.1 Table of metric

Metric Name	value
Throughput	8.123496807
Average Delay	0.00600
Sent packets	100.00
Receive packets	100.00
Drop packets	0.00
Delivery ratio	100.00
Drop ratio	0.00
Time	49.23988
Throughput 1	7.99184
Throughput 2	8.11623

10 Discussion

- For using Jacov Kernel's algorithm, delay is reduced at minimal rate.
- For slow start algorithm increasing cwnd_ size rapidly (greater than given $cwnd\_+ = cwnd\_;$) doesn't always lead to good performance.
- For slight modification in AODV protocol, drop ratio decreased.
- AODV ensures better performance than DSDV.
- While running modified NS2, an error was faced, titled "floating point exception(core dumped)". This happened because of introducing new congestion avoidance algorithm, where there was a line $cwnd\_+ = 1/cwnd\_$ and $cwnd\_$ was initialized as 0. Adding simple if else block, the problem was solved.