

CSE-322

Report on NS2

Submitted By-

Waqar Hassan Khan (1505107)

Ahmed Nafis Fuad (1505113)



Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology
(BUET)

Dhaka 1000

January 12, 2019

Contents

1	Network Topologies Under Simulation	3
2	Parameters Under Variation	3
2.1	wireless 802.11 - mobile	3
2.2	wireless 802.15.4 - static	3
3	Modifications Made in The Simulator	4
4	Graphs	6
4.1	802.11	6
4.2	802.15.4	22
5	Summary Findings	38

1 Network Topologies Under Simulation

- wireless 802.11 - mobile
- wireless 802.15.4 - static

2 Parameters Under Variation

2.1 wireless 802.11 - mobile

- number of nodes
- number of flow
- number of packets sent per second
- speed of node

2.2 wireless 802.15.4 - static

- number of nodes
- number of flow
- number of packets sent per second
- coverage area

3 Modifications Made in The Simulator

- change in tcp.cc : the RTO(re-transmission timeout) calculation implemented in ns2.35 \rightarrow tcp \rightarrow tcp.cc follows Jacobson Algorithm. The calculation has been changed.

In Jacobson algorithm, in order to calculate the current RTO, TCP sender need to maintain two variables, SRTT and RTTVAR. After completing the first RTT,

$$SRTT_1 = RTT_1$$

$$RTTVAR_1 = RTT_1/2$$

for the next updates,

$$SRTT_{n+1} = (1 - \alpha) * SRTT_n + \alpha * RTT_{n+1}$$

$$RTTVAR_{n+1} = (1 - \beta) * RTTVAR_n + \beta * |SRTT_{n+1} - RTT_{n+1}|$$

$$RTO_{n+1} = SRTT_{n+1} + 4 * RTTVAR_{n+1}$$

The calculation has been changed according to a paper, a variable K - RTT's rate of change is required by the following formula $K_{n+1} = |RTT_{n+1} - RTT_n|/RTT_n$

if $K > 1$ then $K = 1$.

unlike Jacobson algo α and β would be changed

$$\alpha_{n+1} = \alpha_0 * (1 + K_{n+1})$$

$$\beta_{n+1} = \beta_0 * (1 - K_{n+1})$$

the values of $\alpha_0 = 1/8$ and $\beta_0 = 1/4$

$$SRTT_{n+1} = (1 - \alpha_{n+1}) * SRTT_n + \alpha_{n+1} * RTT_{n+1}$$

$$RTTVAR_{n+1} = (1 - \beta_{n+1}) * RTTVAR_n + \beta_{n+1} * |SRTT_{n+1} - RTT_{n+1}|$$

$$RTO_{n+1} = SRTT_{n+1} + 4 * RTTVAR_{n+1}$$

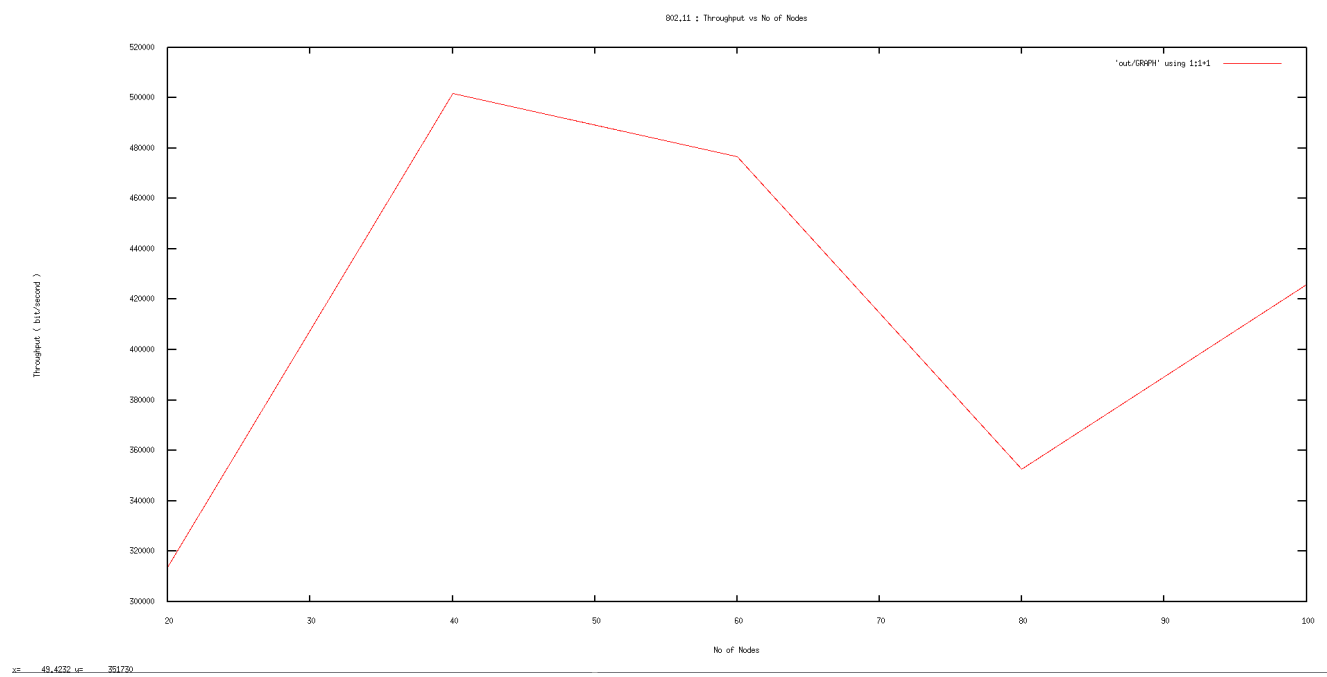
the change improves the retransmission of packet, it decreases.

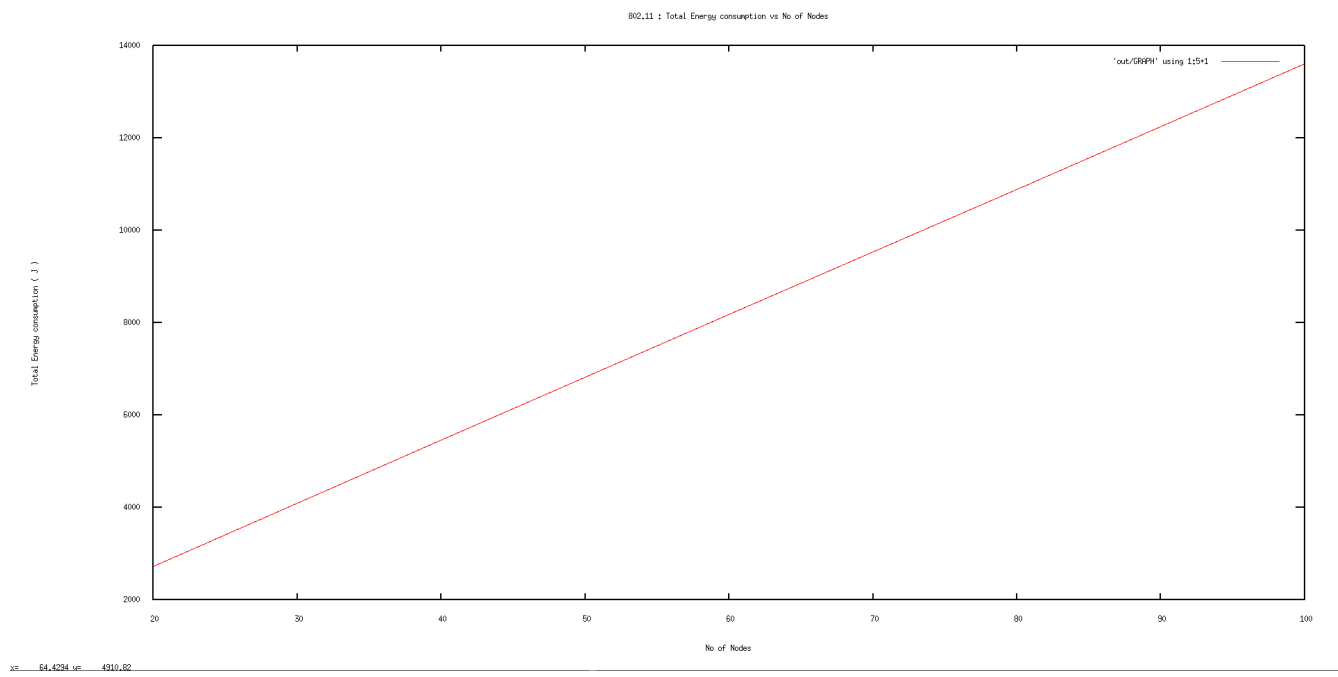
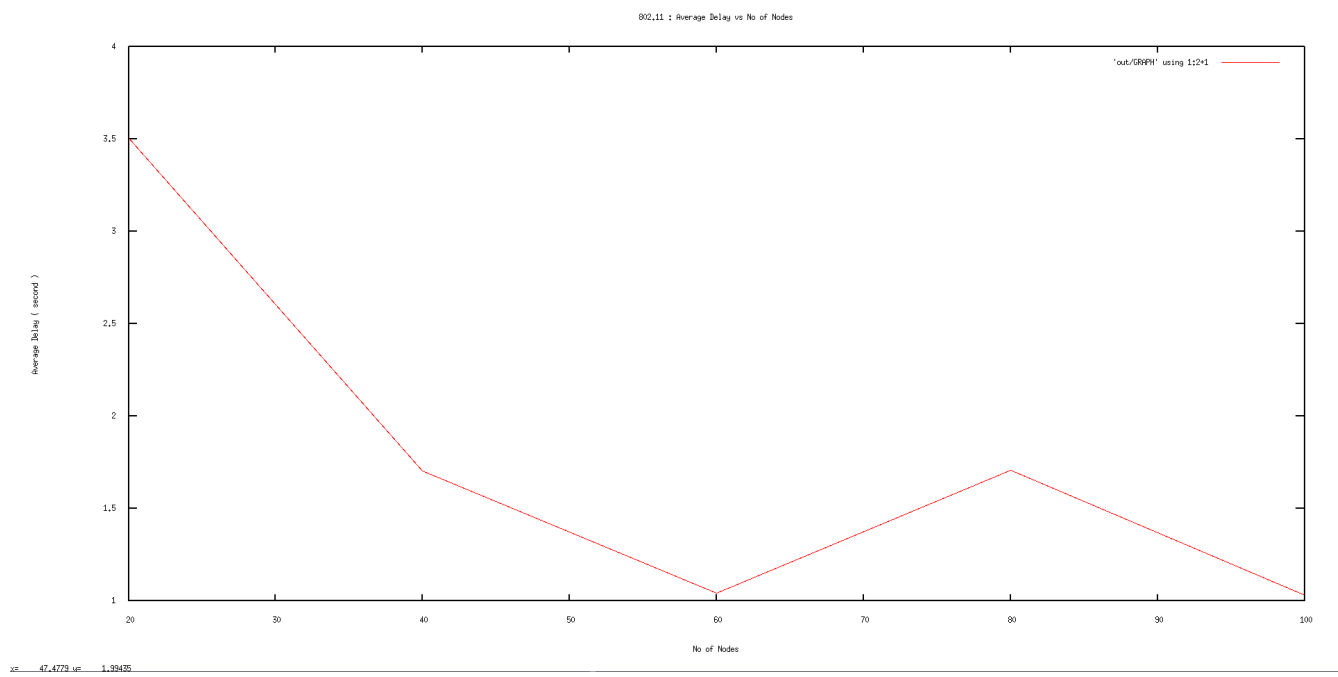
- DSDV protocol has been modified. The DVR(distance vector routing) algorithm had a problem of "count to infinity", in order to fix it DSDV uses a sequence number. Here DSDV has been converted to DVR by omitting the codes associated with sequence number -j, as a result throughput has decreased, possibly there are some entries got missing.

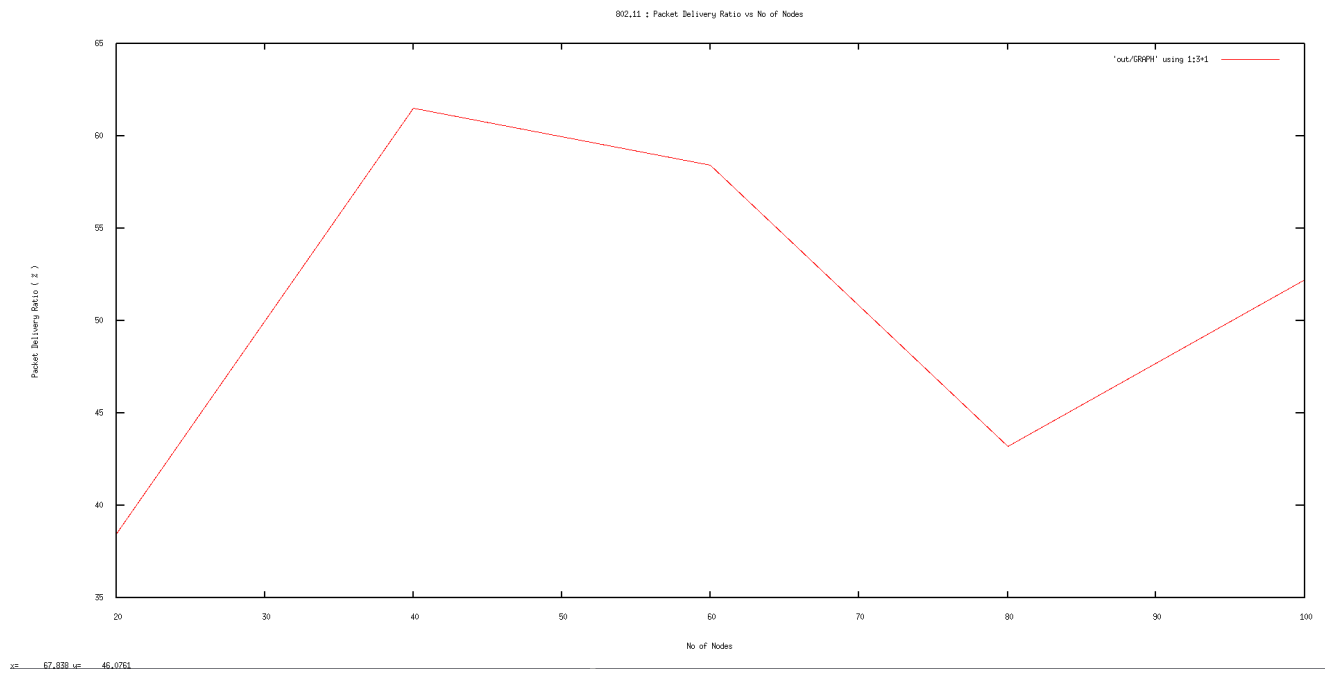
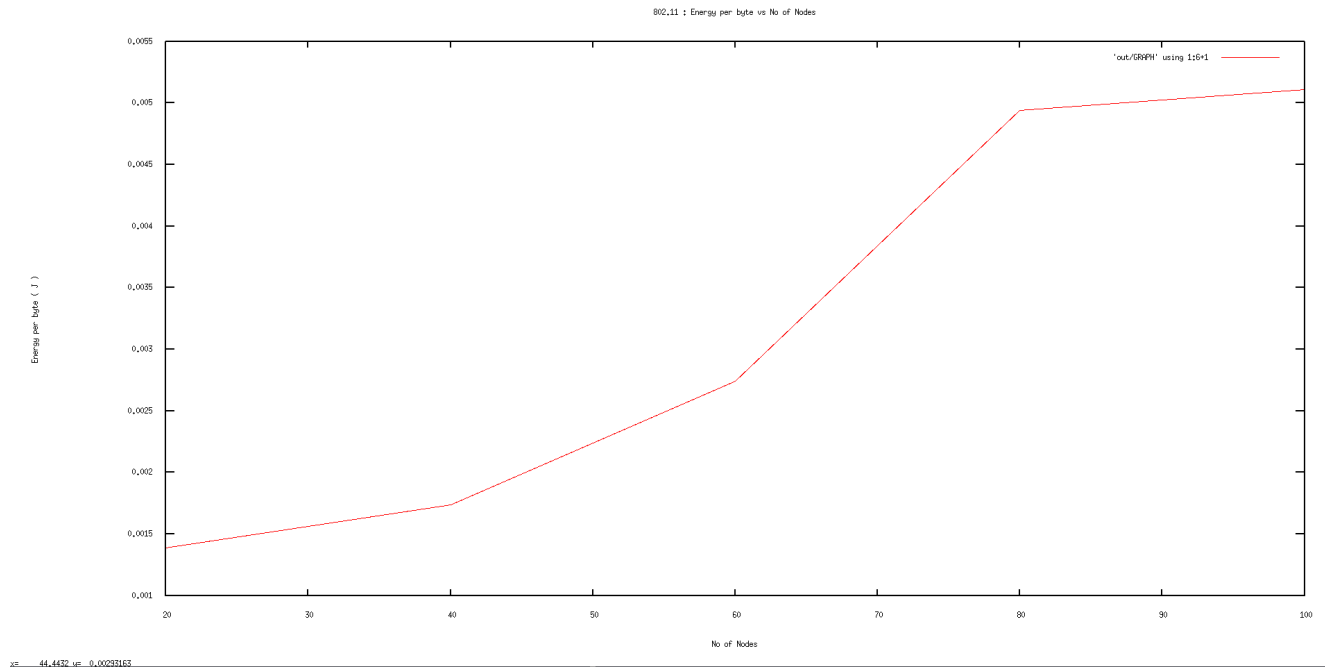
4 Graphs

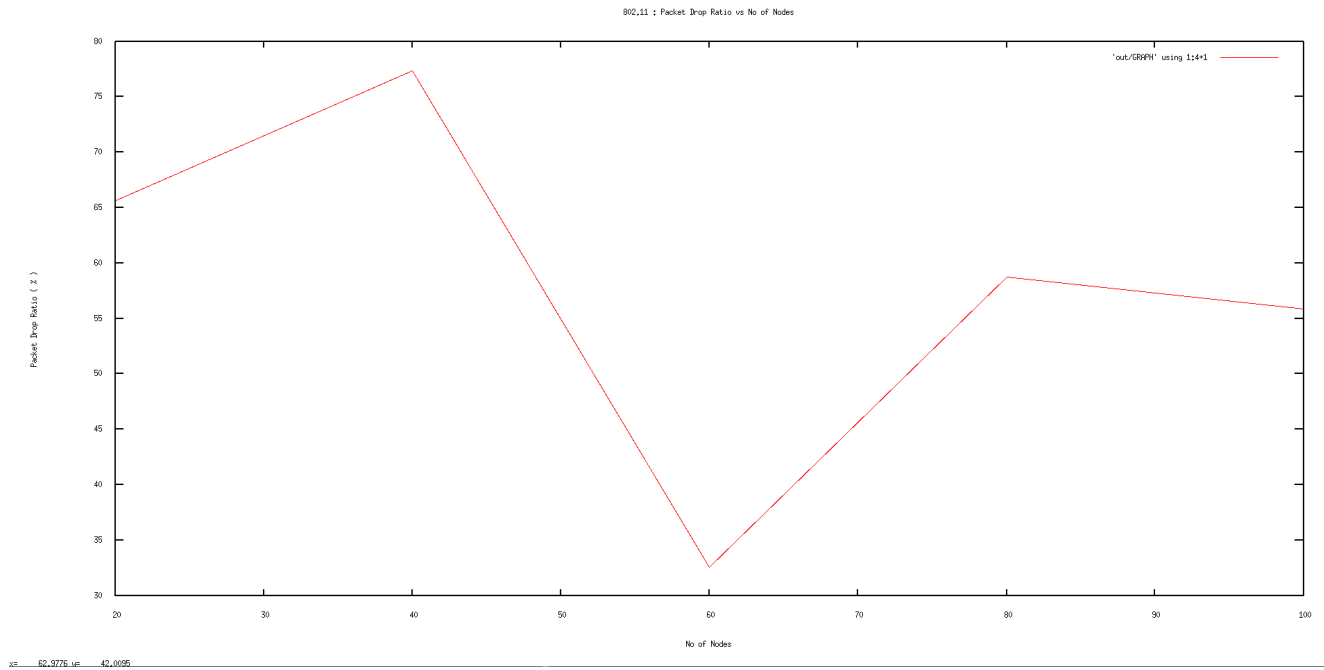
4.1 802.11

Variation in Number of Nodes

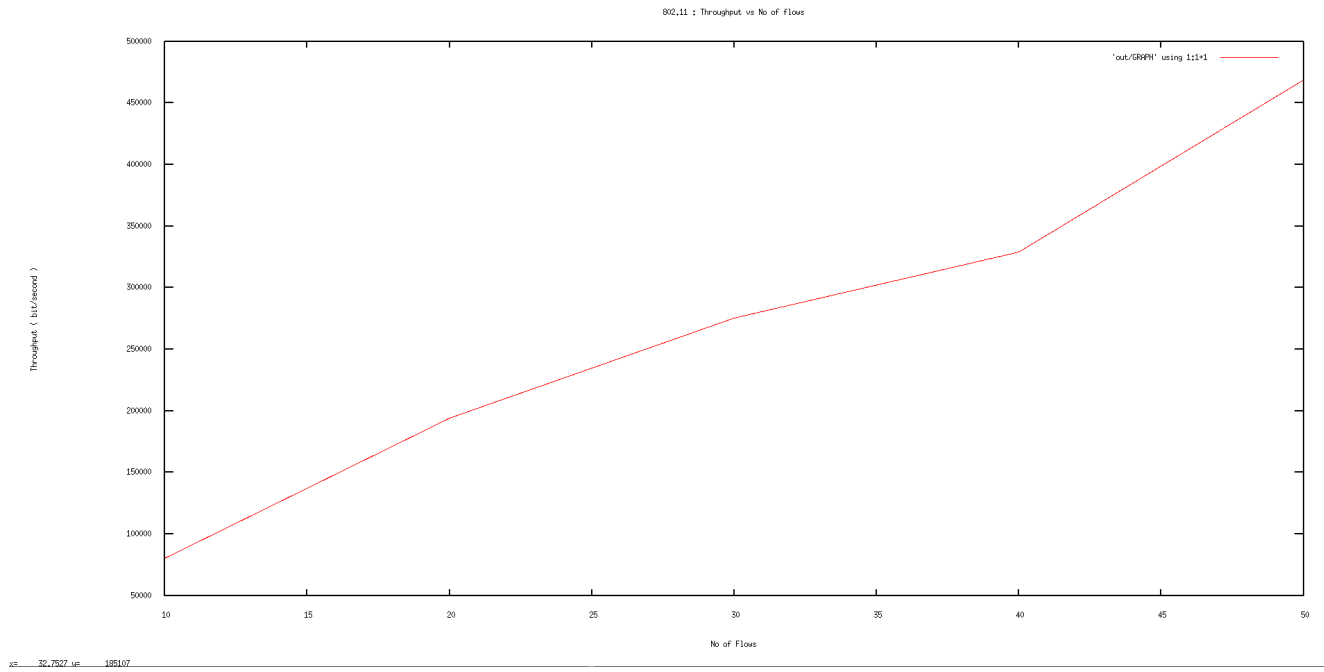


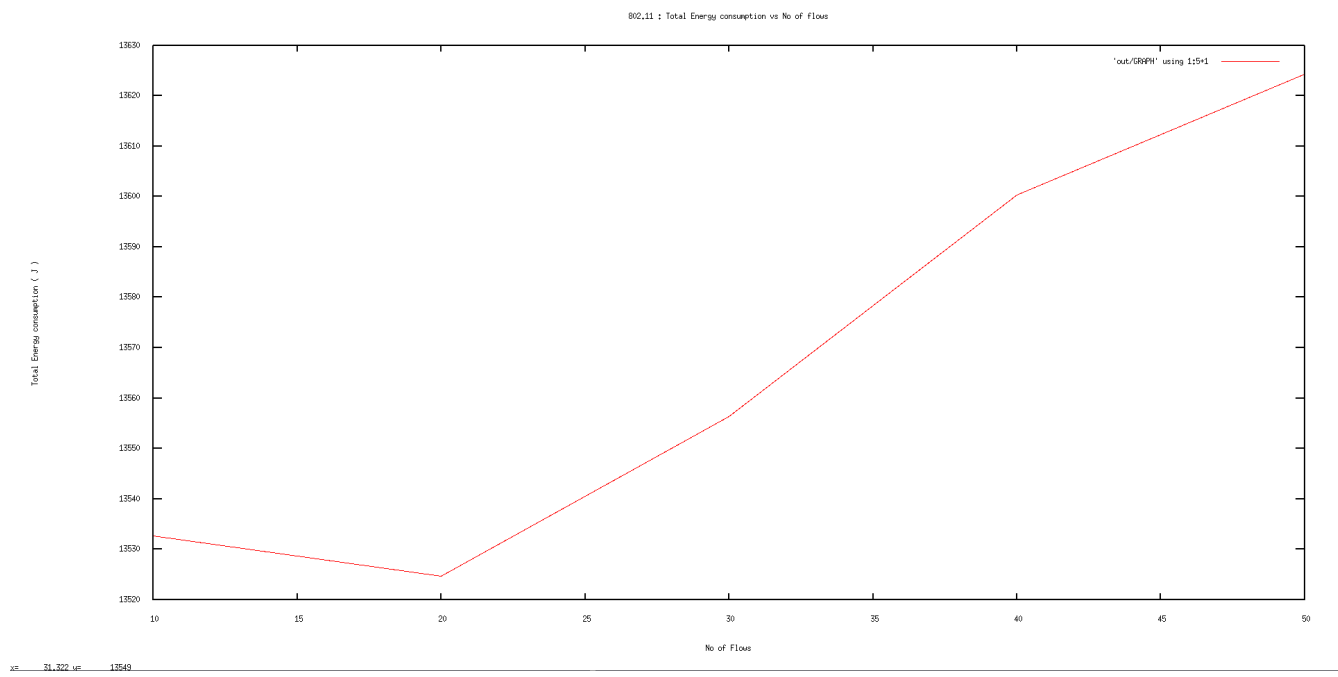
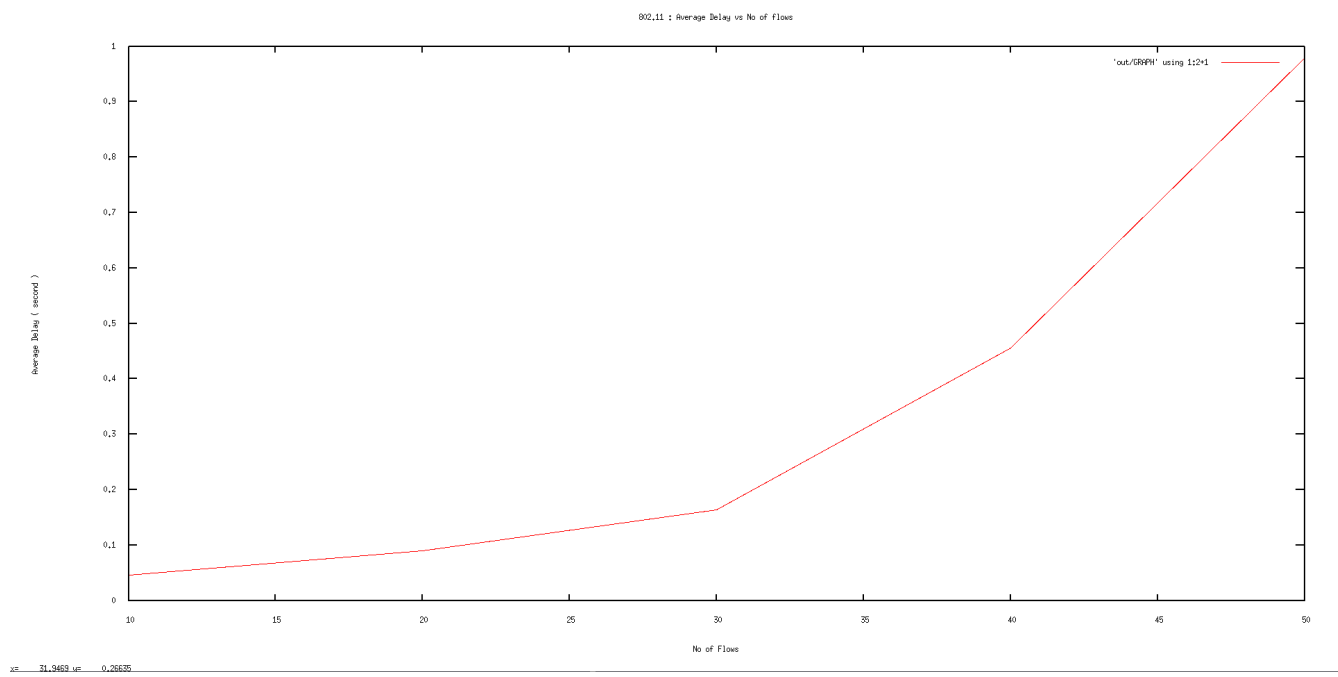


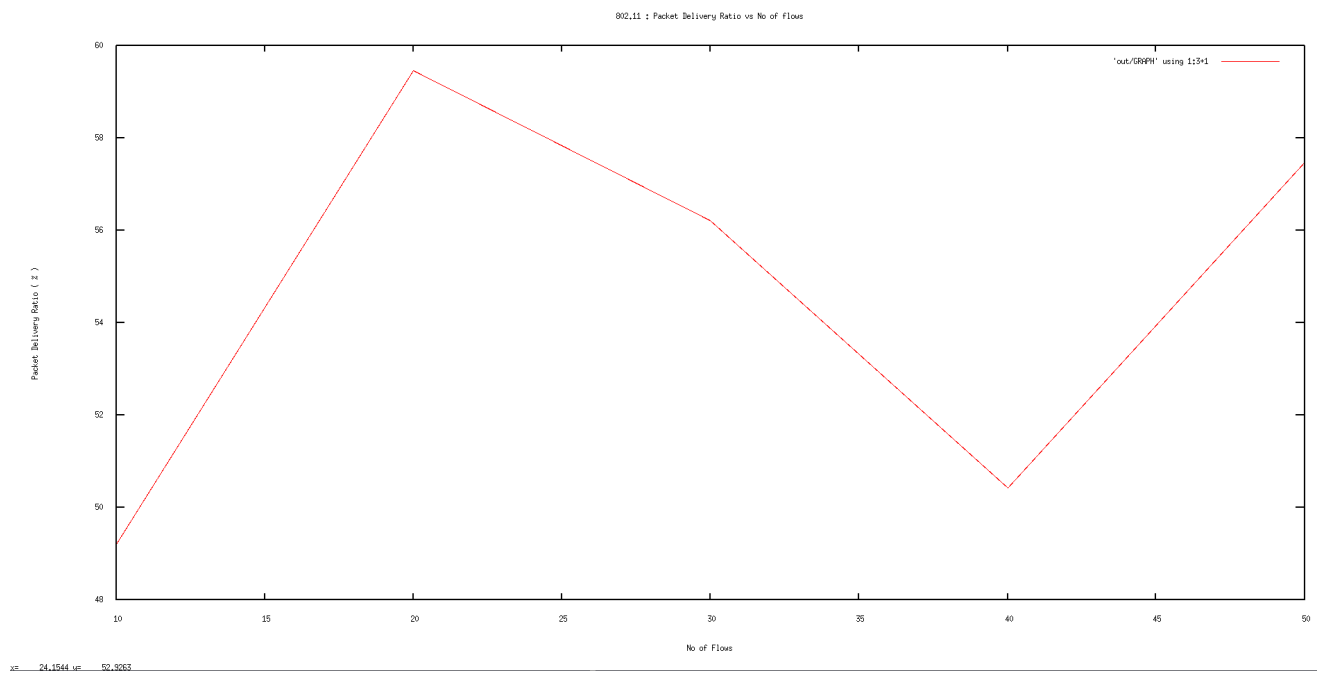
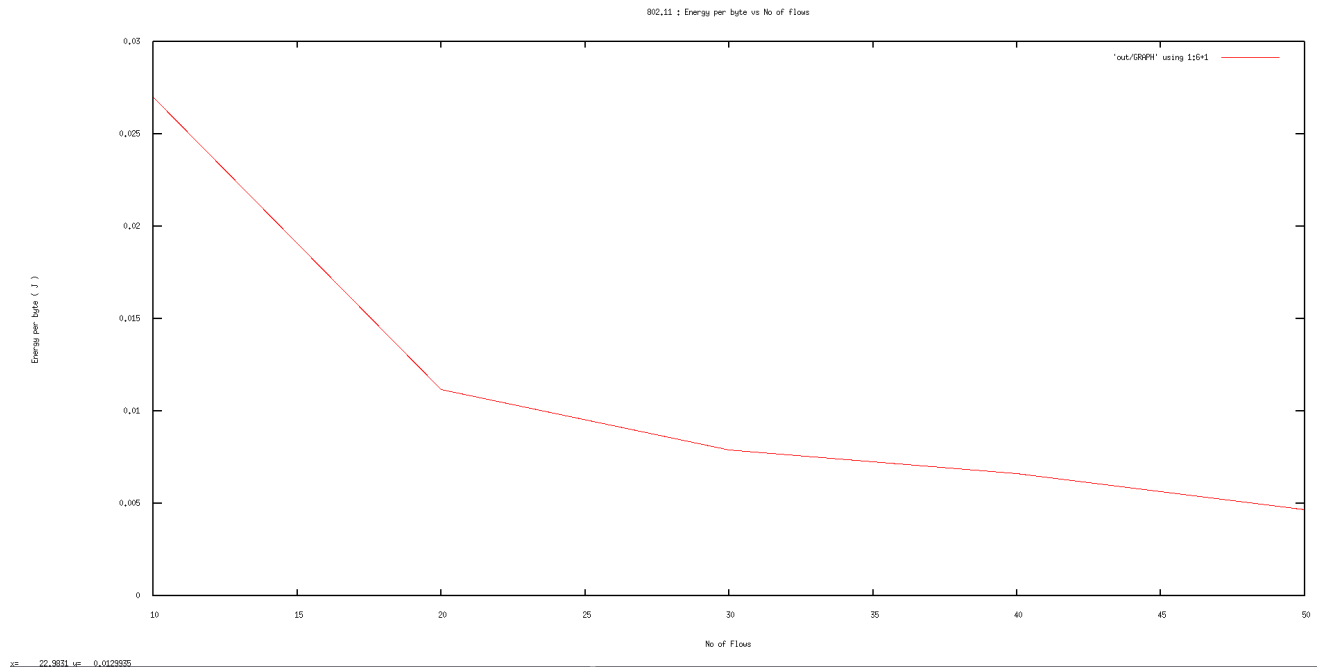


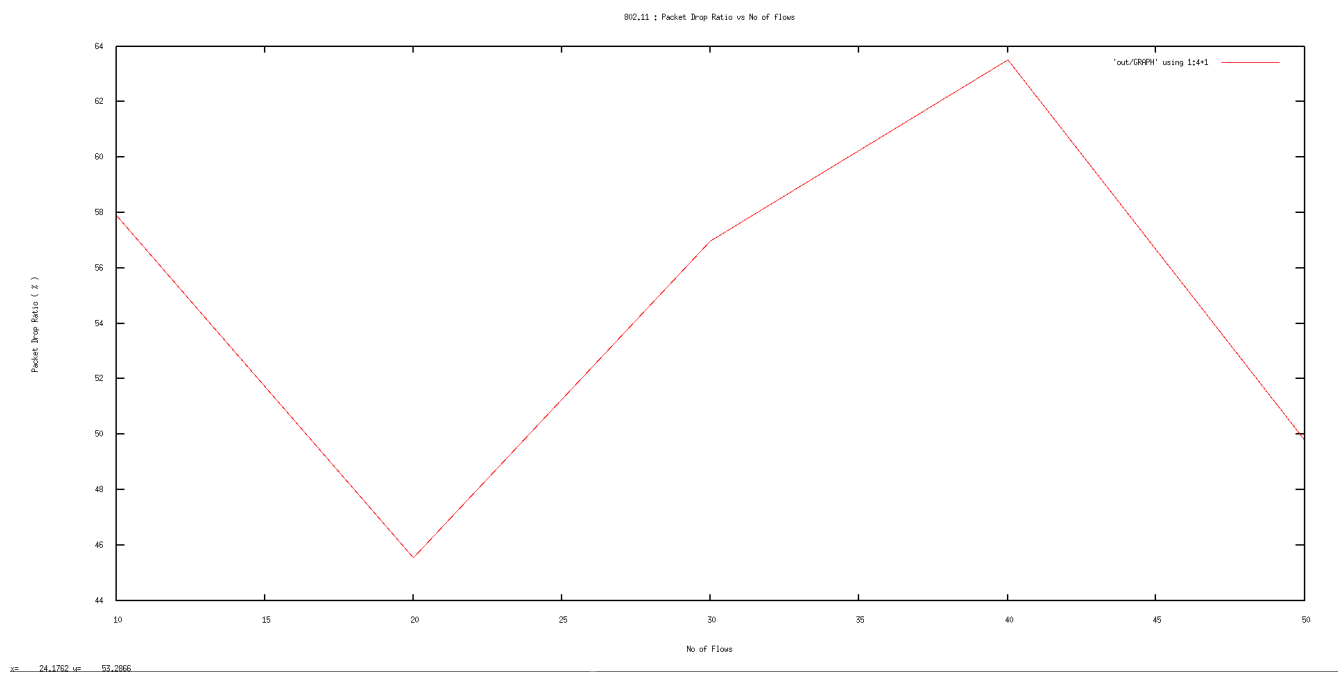


Variation in Number of Flow

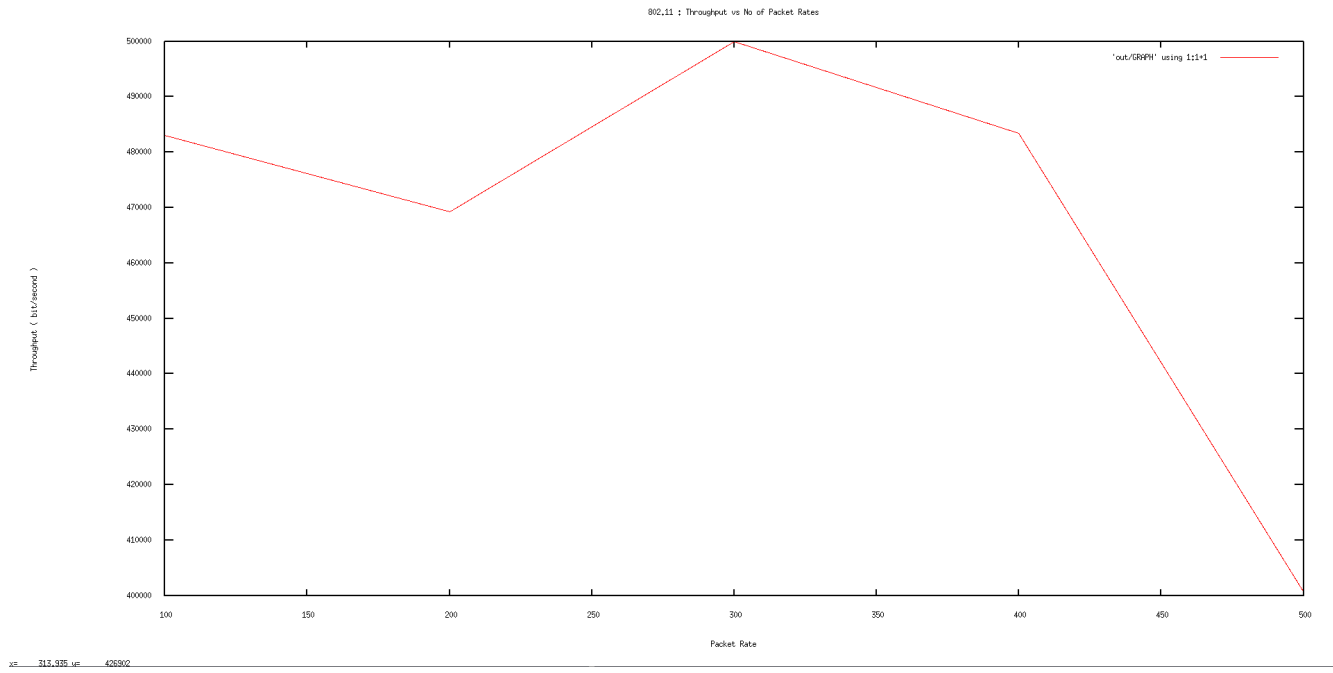


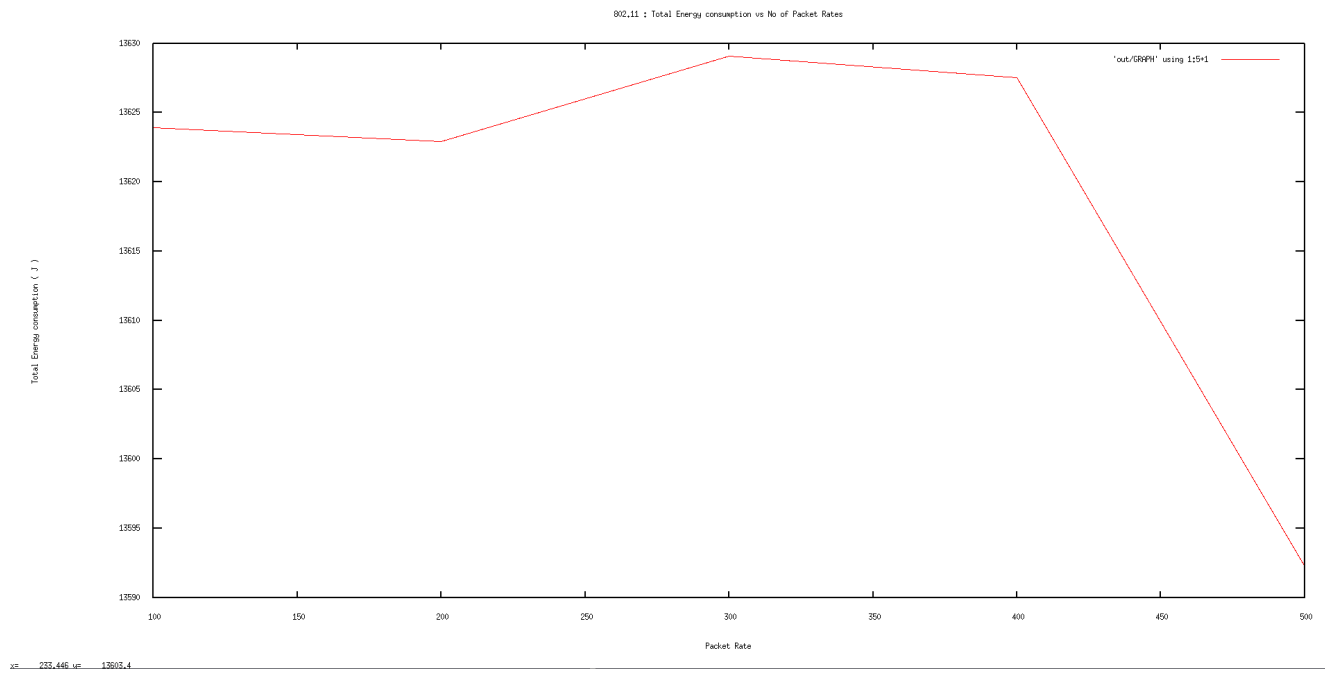
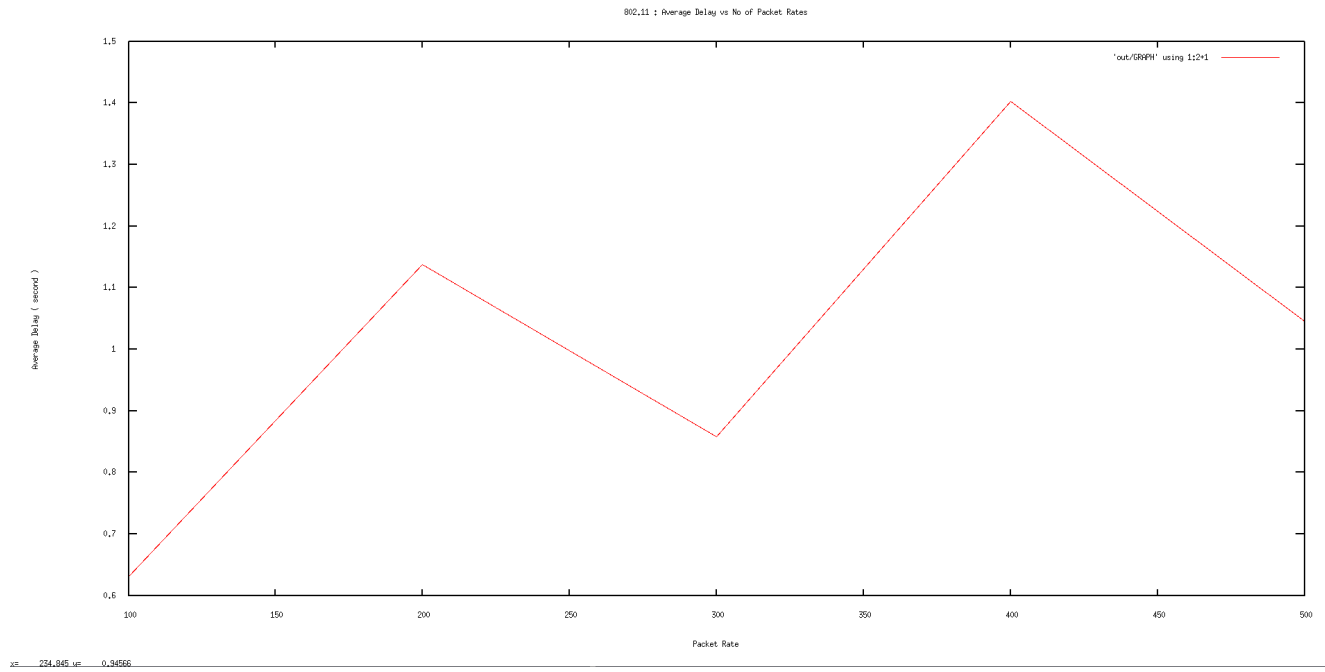


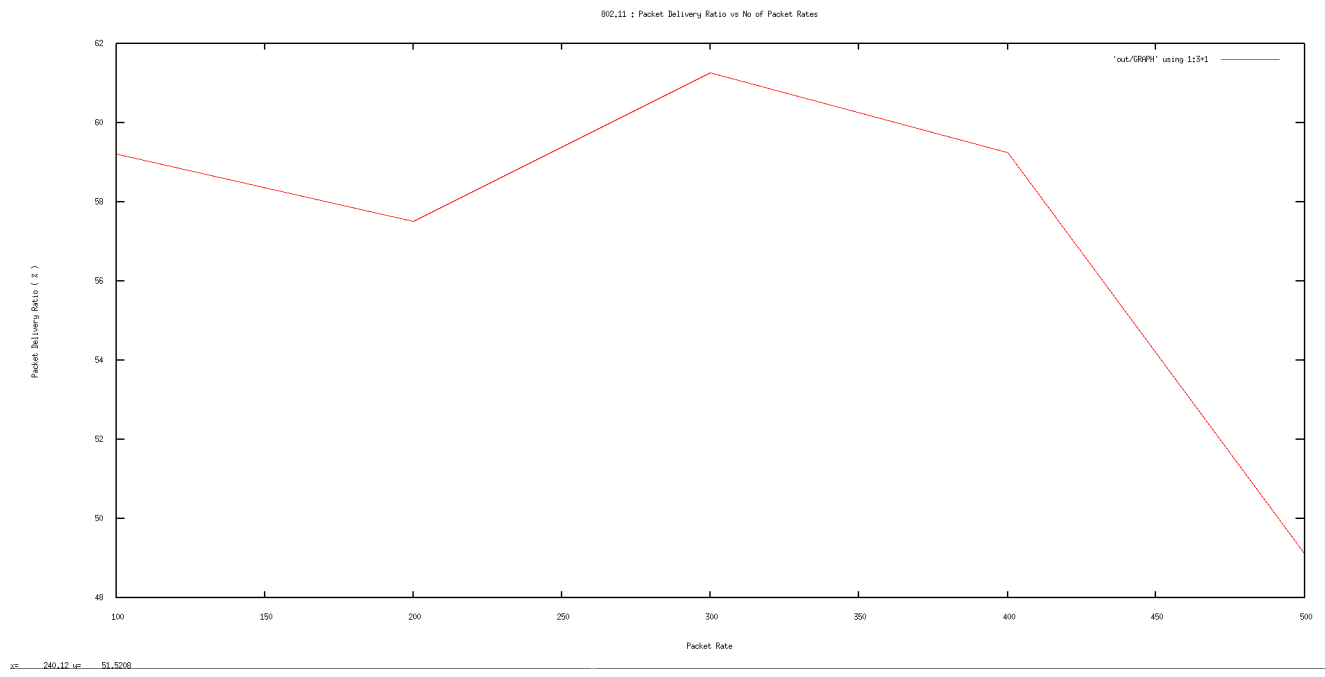
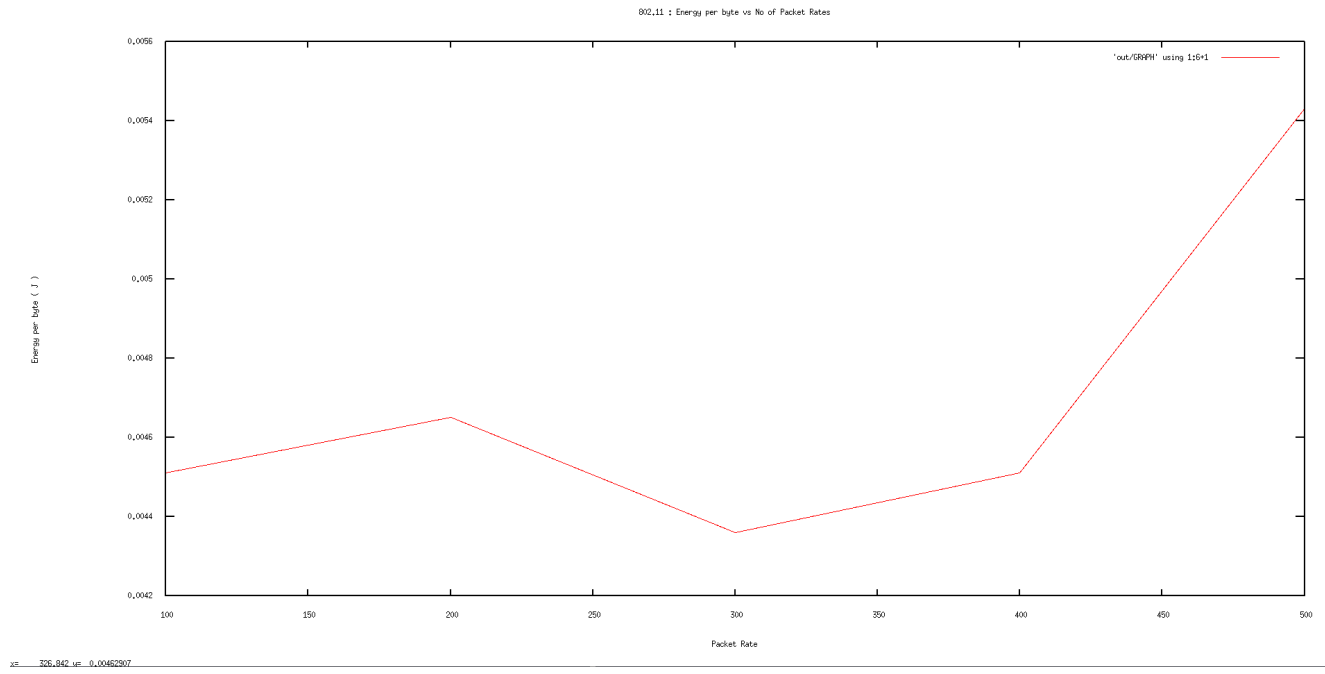


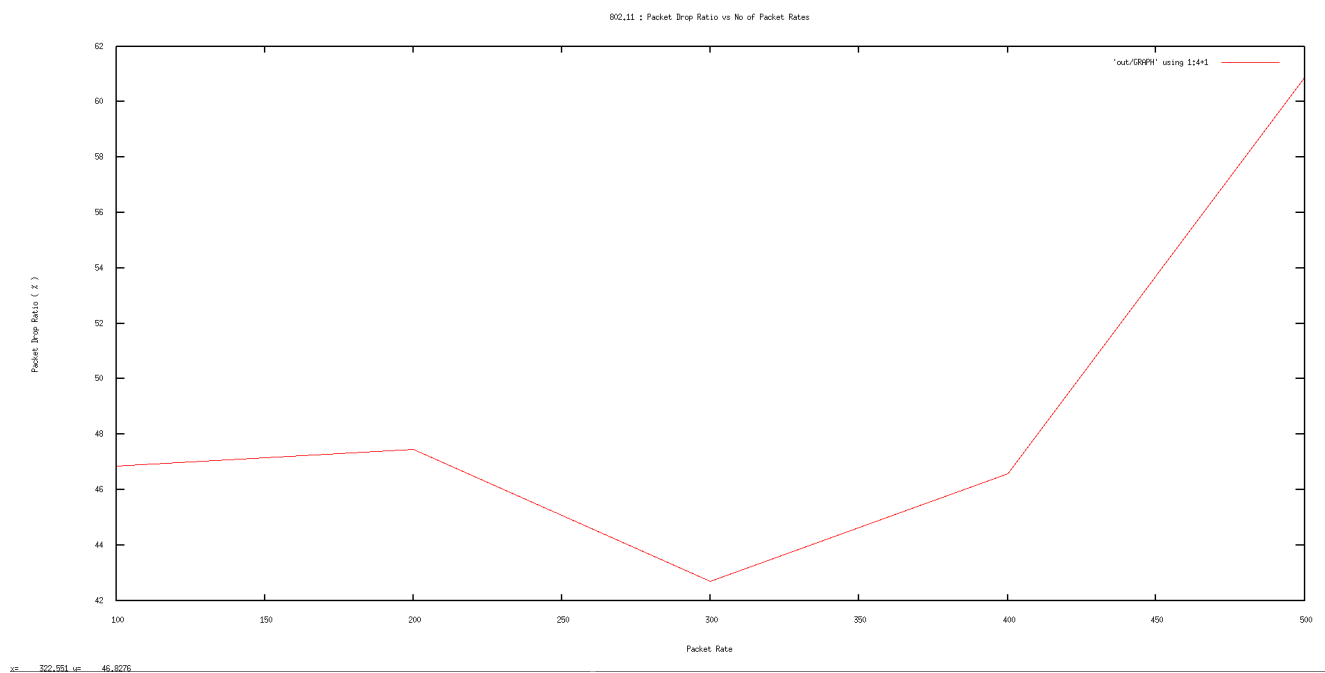


Variation in Packets Sent per Second

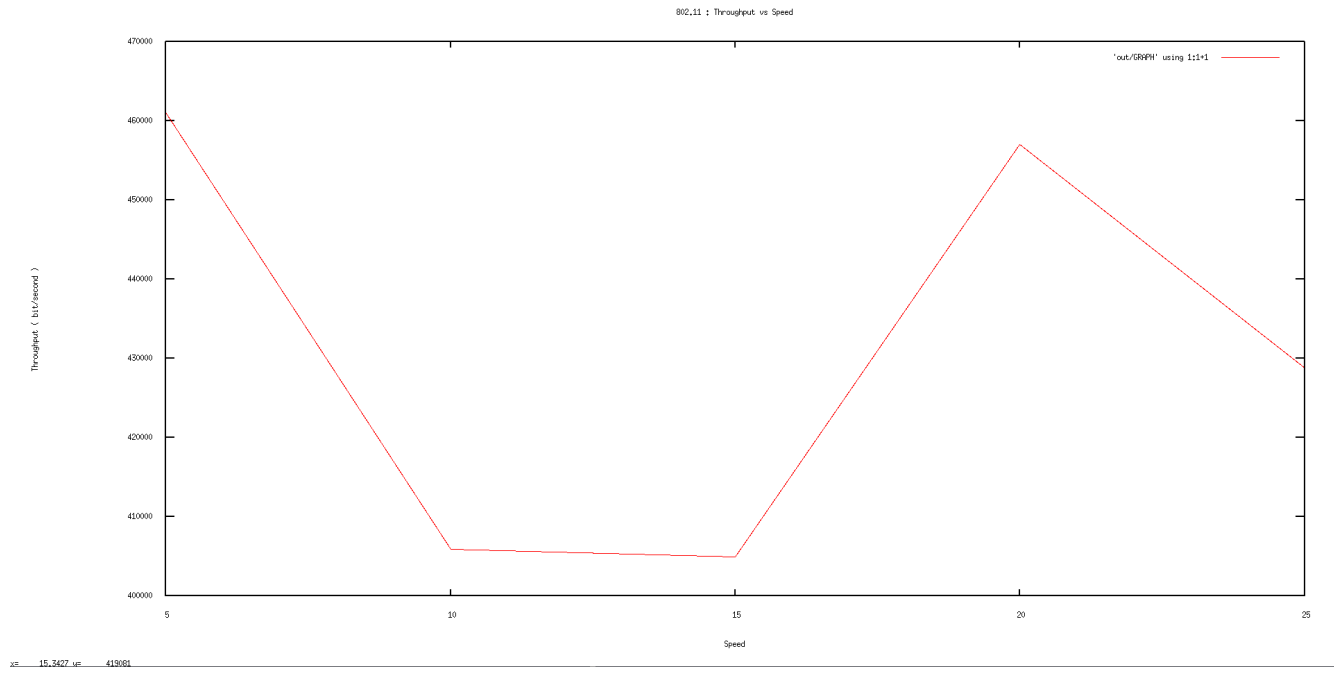


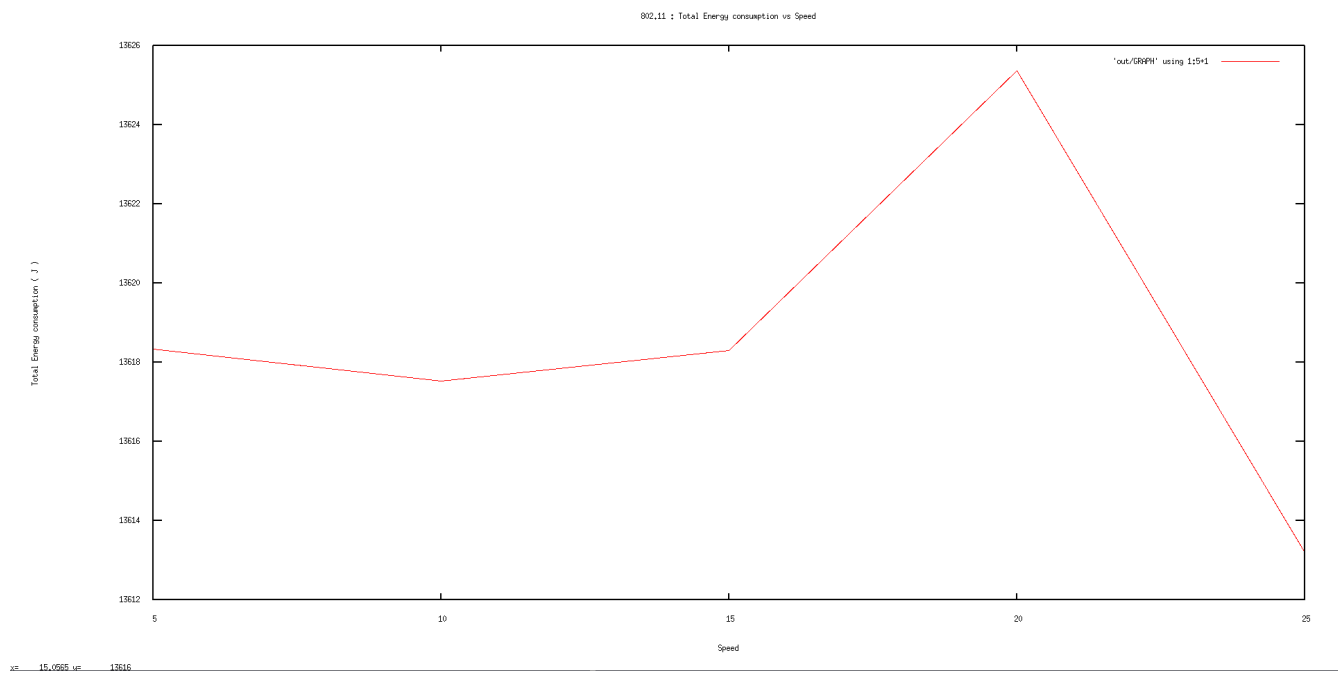
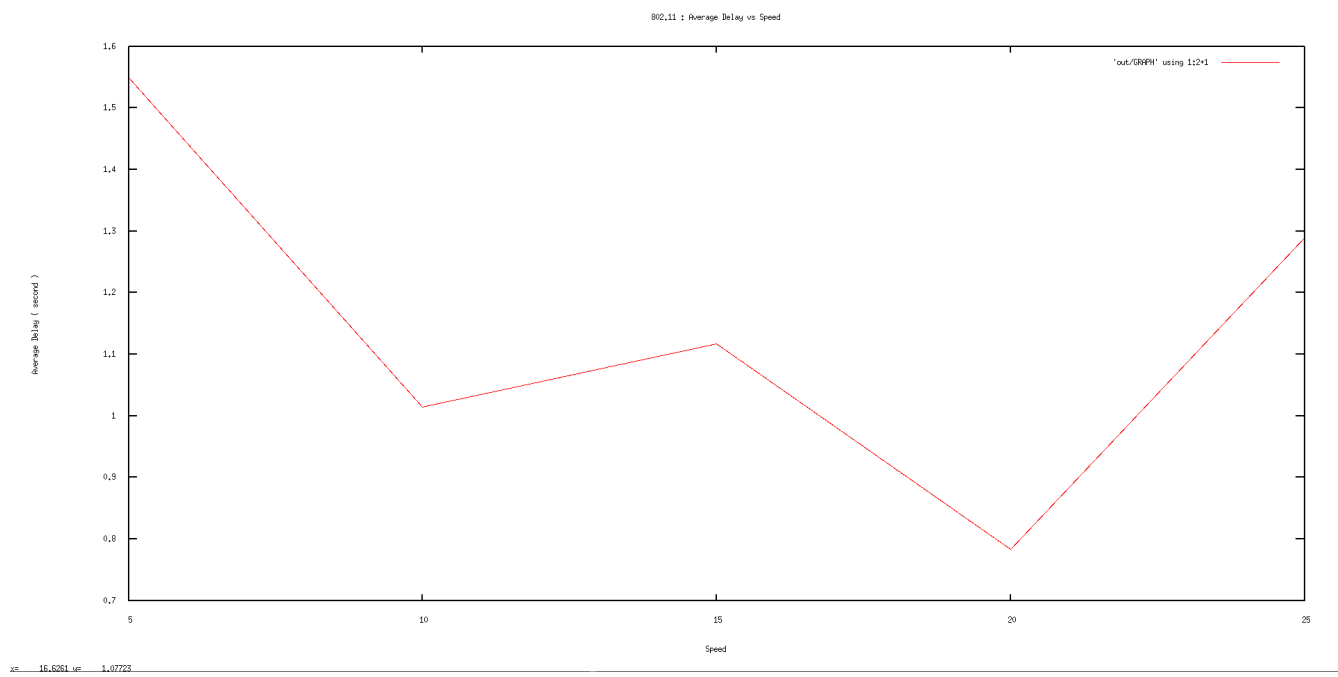


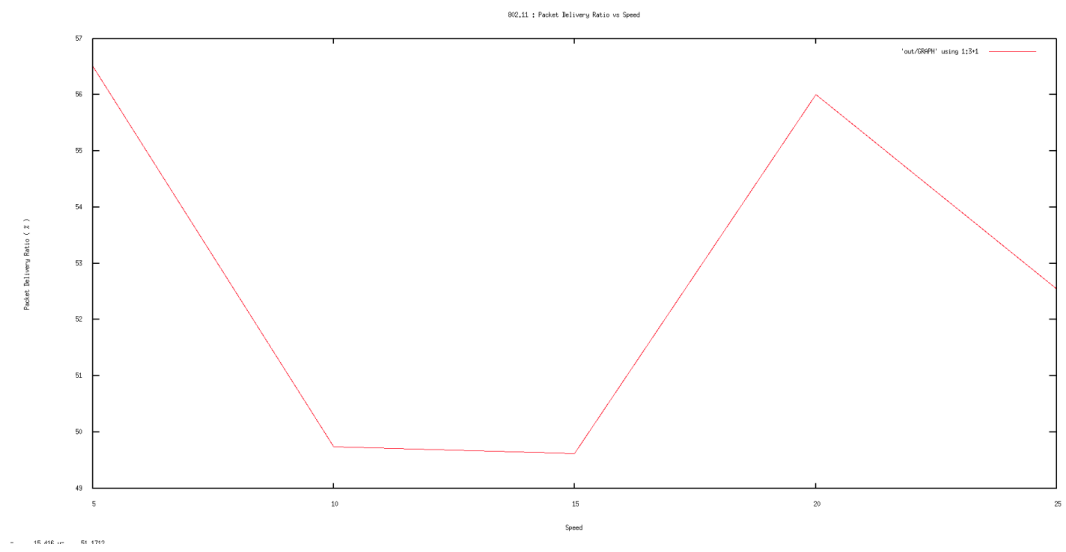
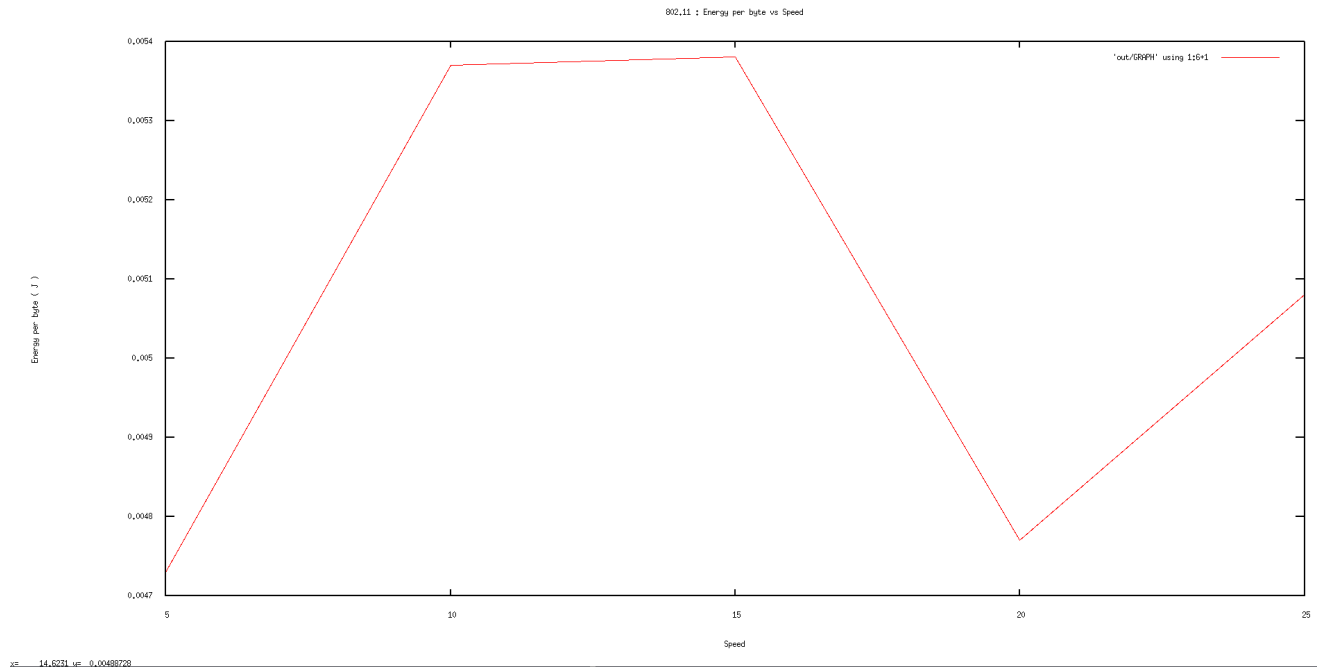


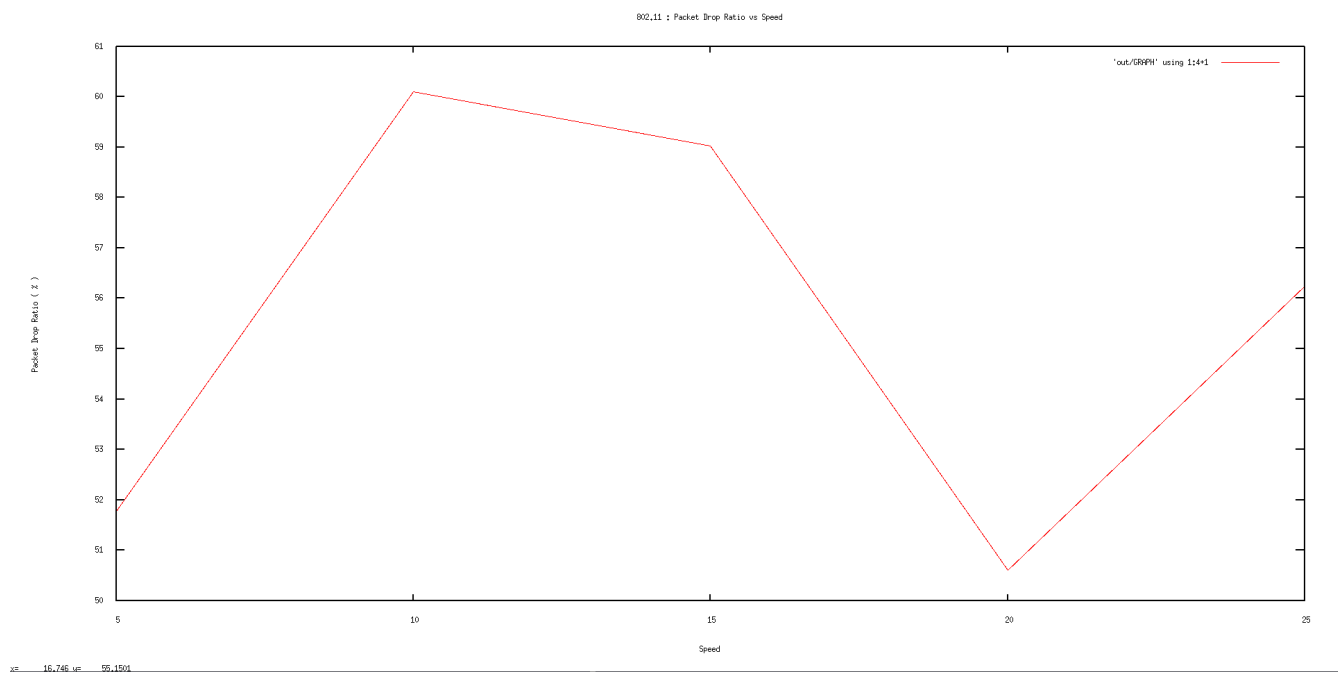


Variation in Speed



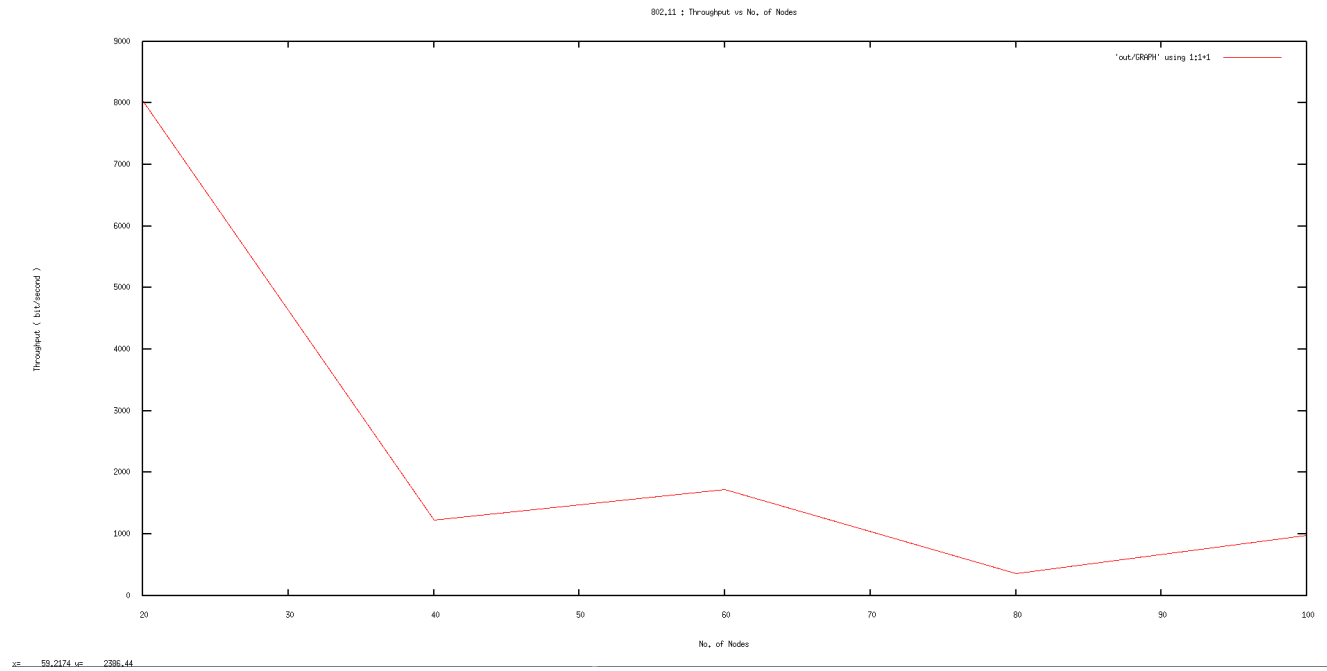


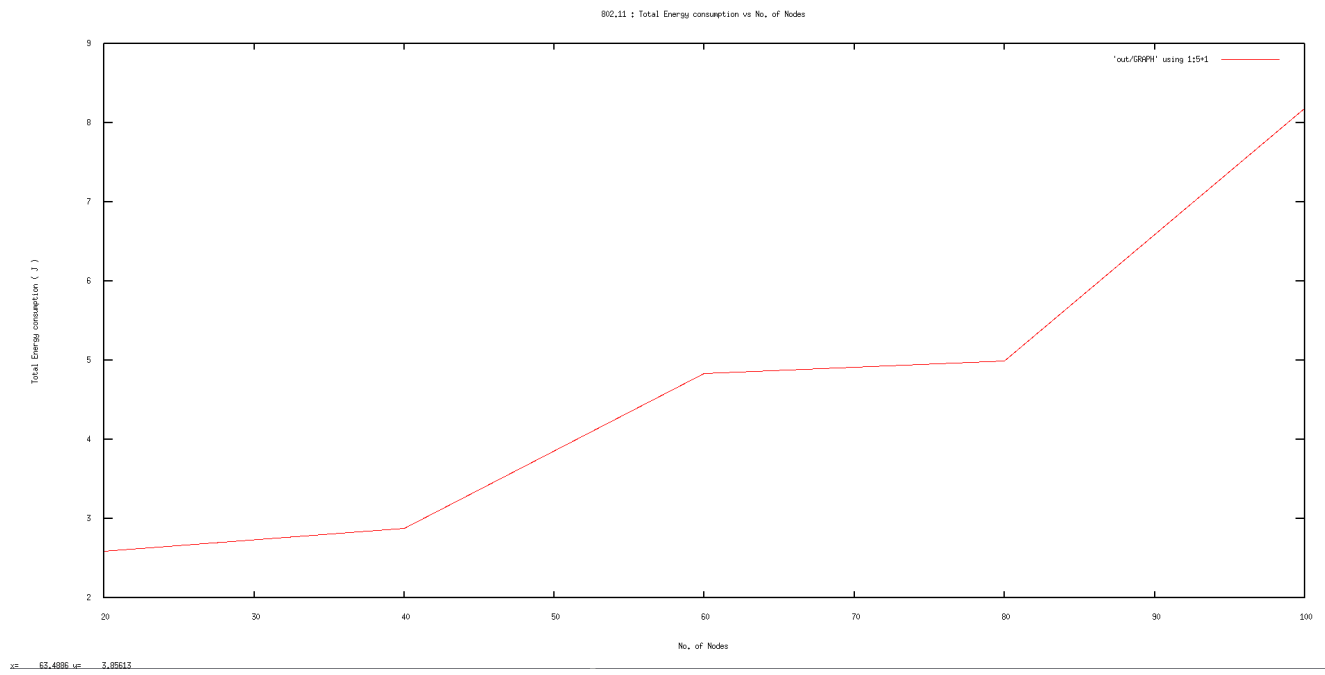
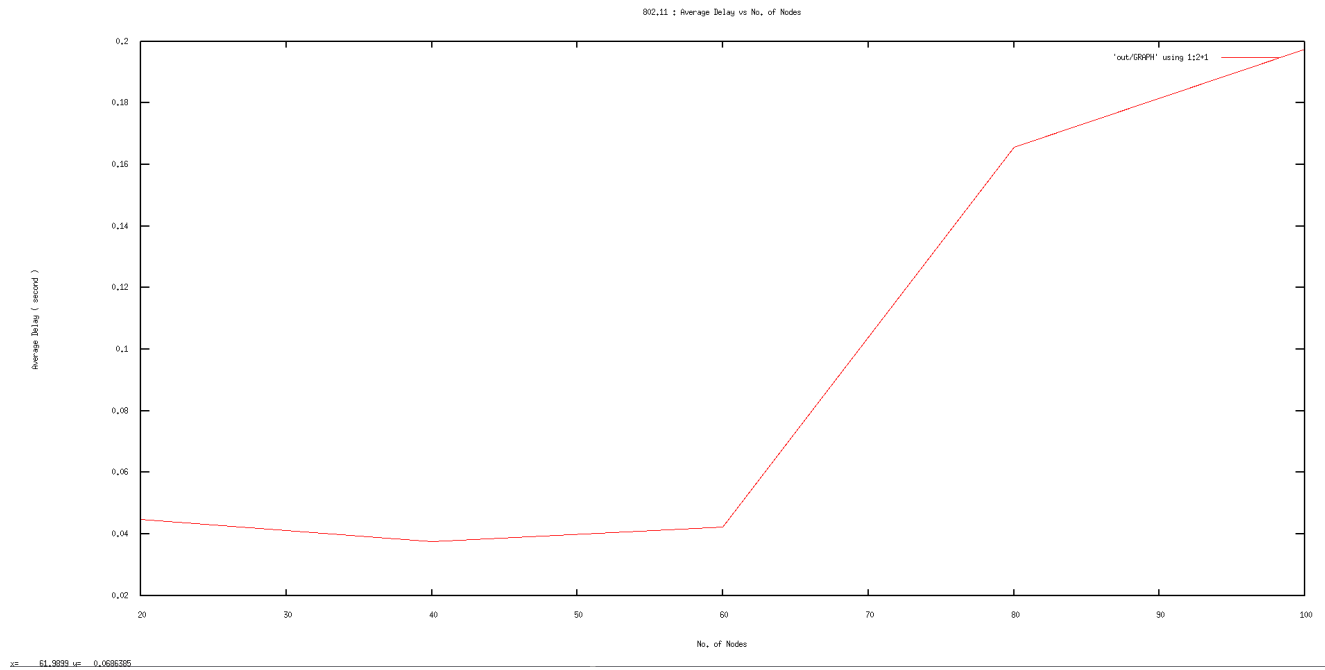


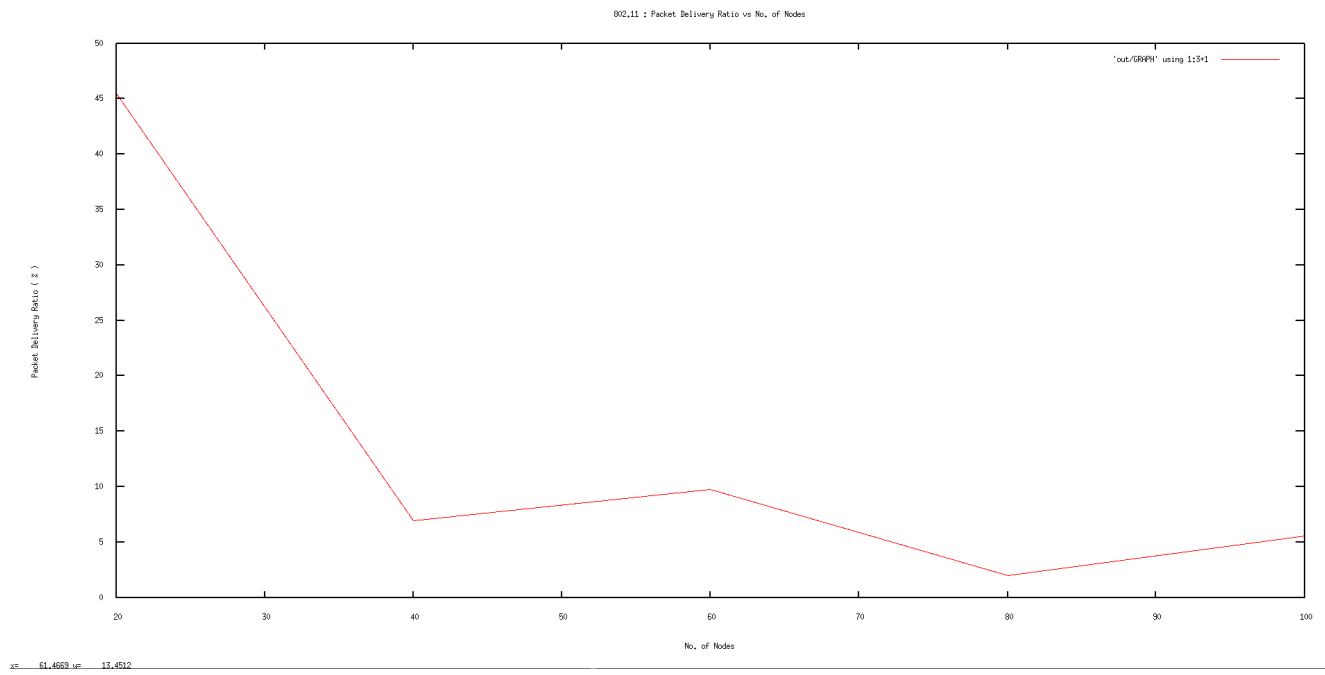
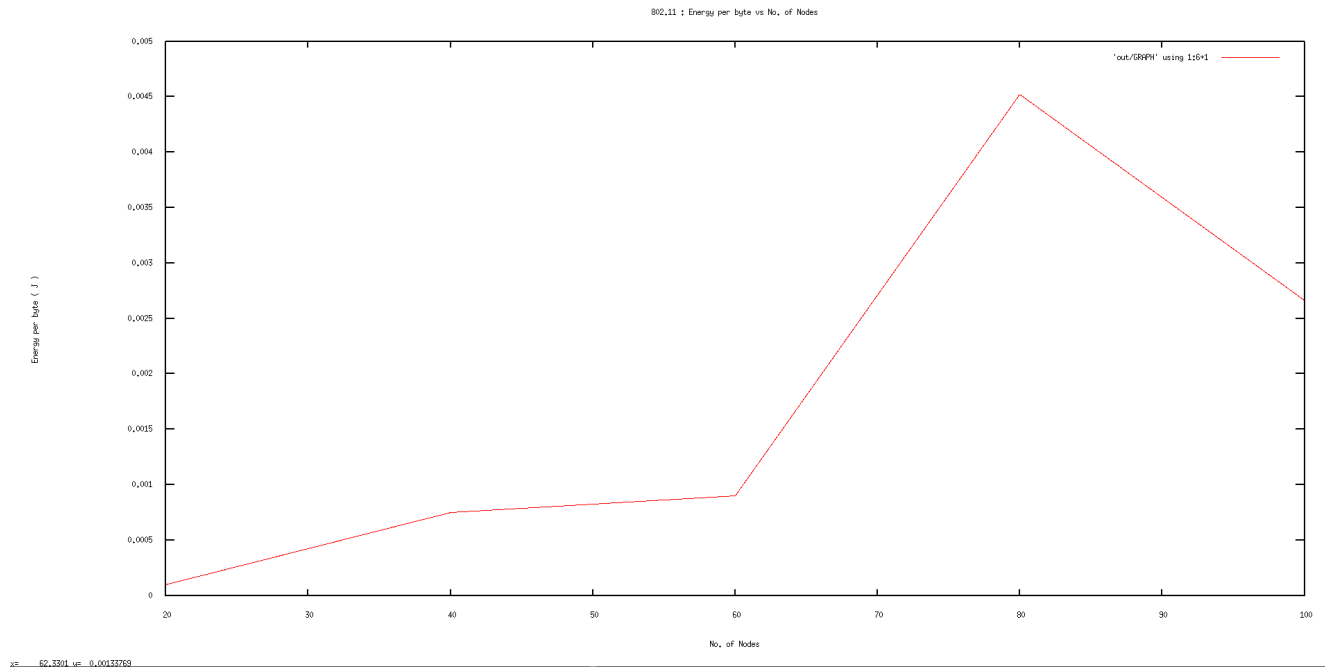


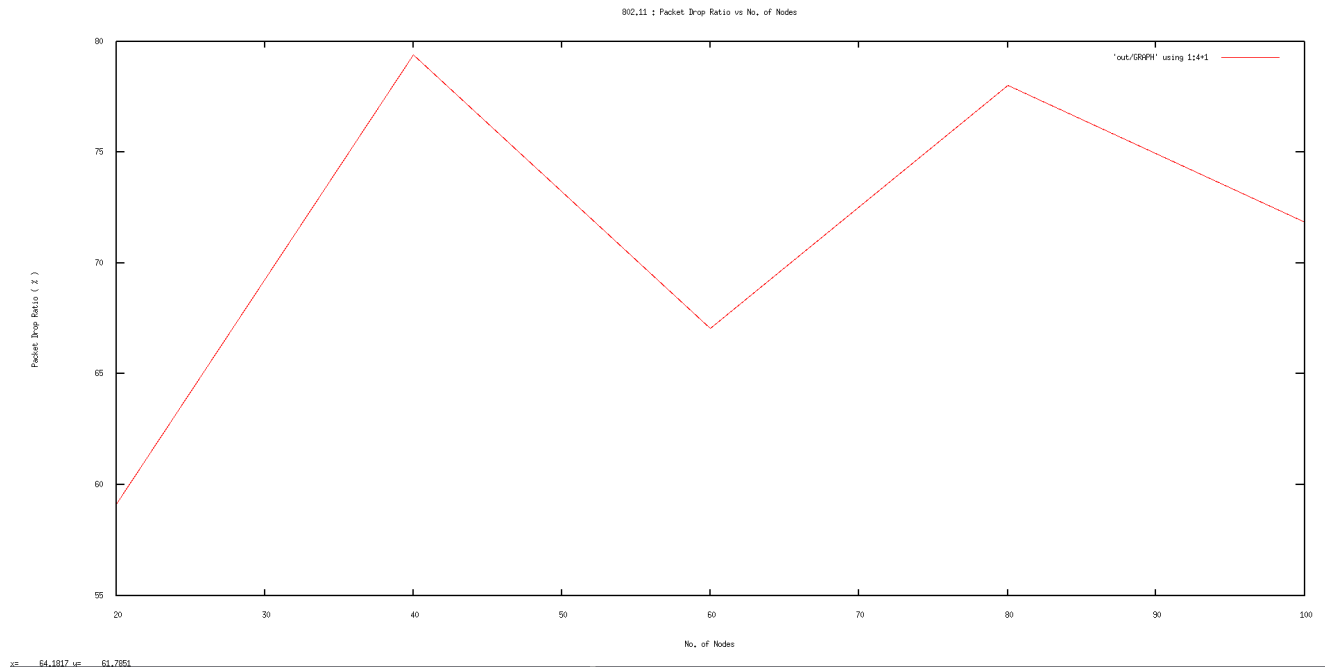
4.2 802.15.4

Variation in Number of Nodes

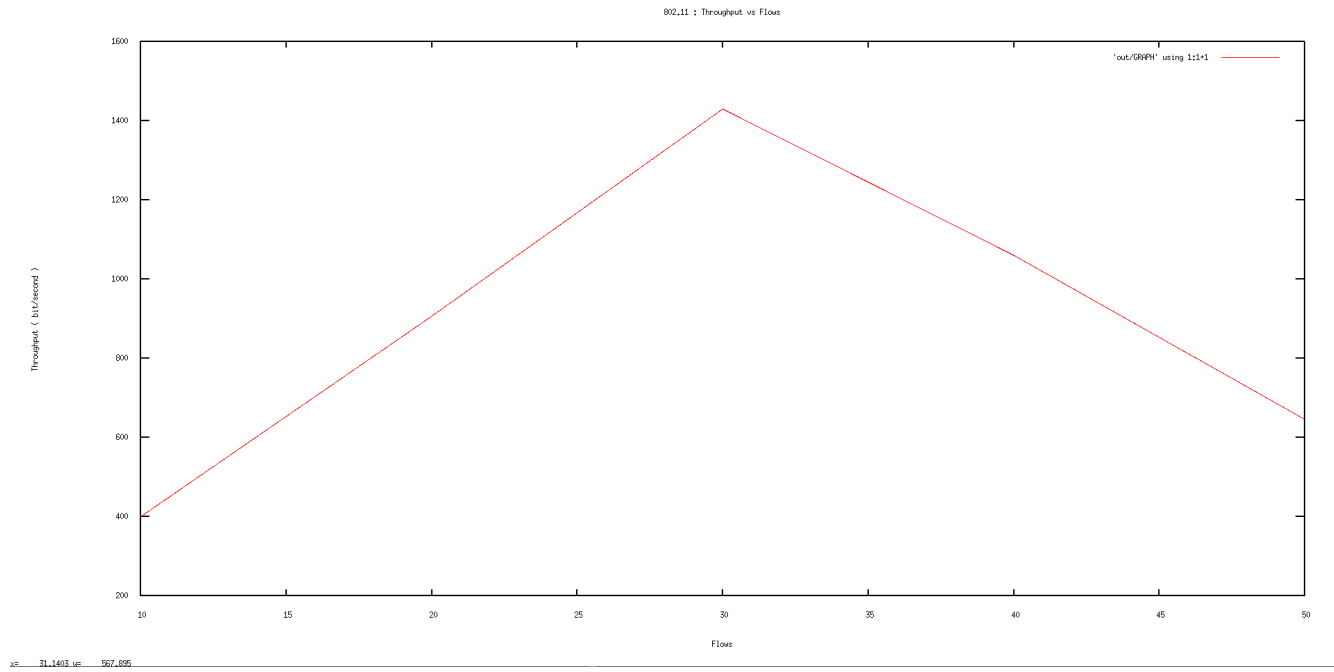


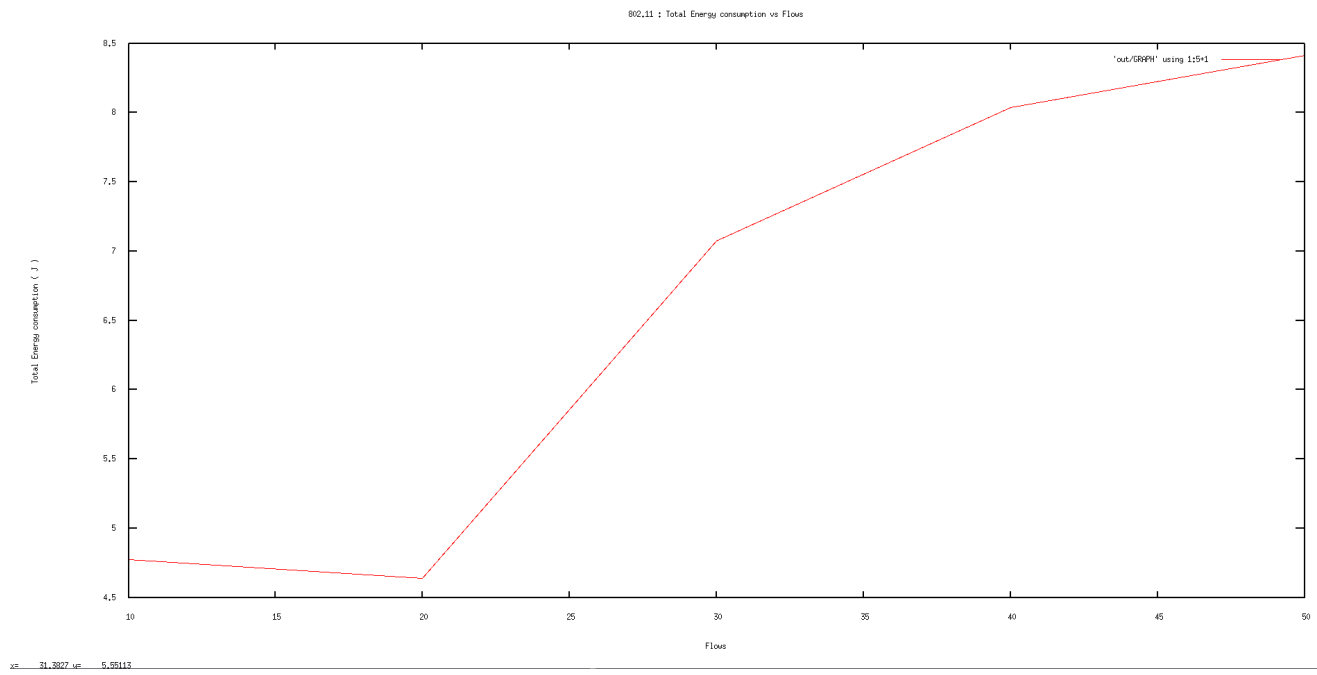
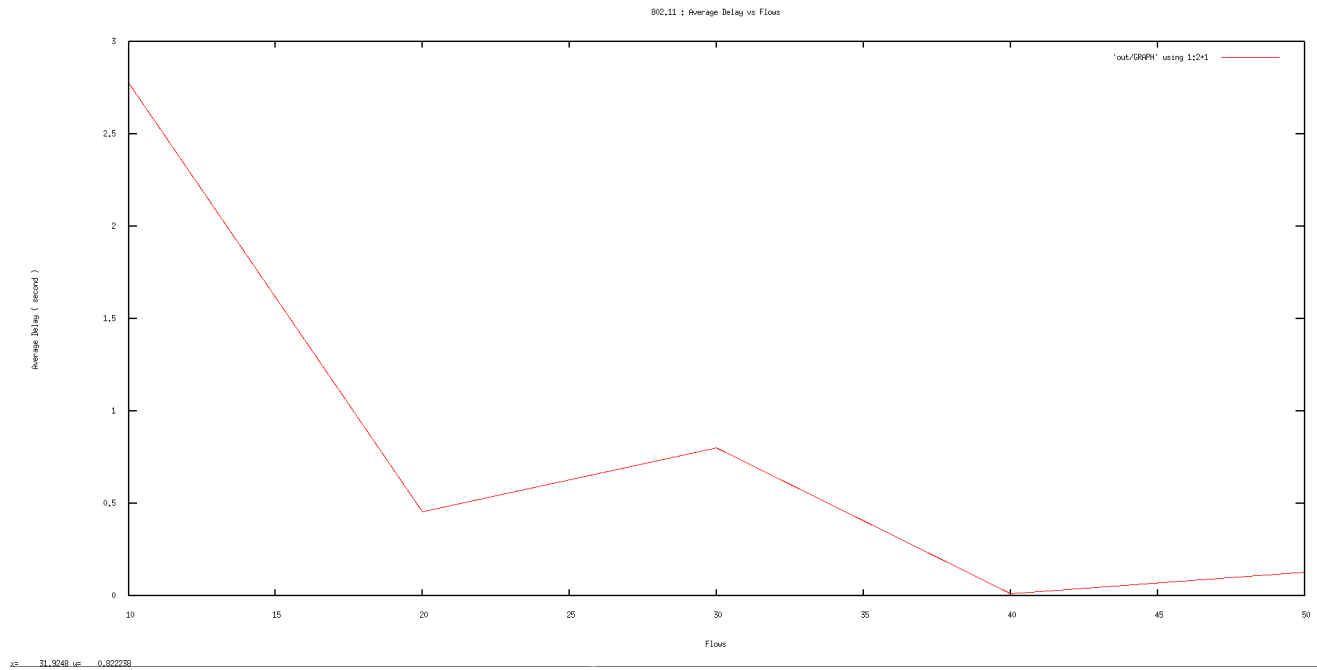


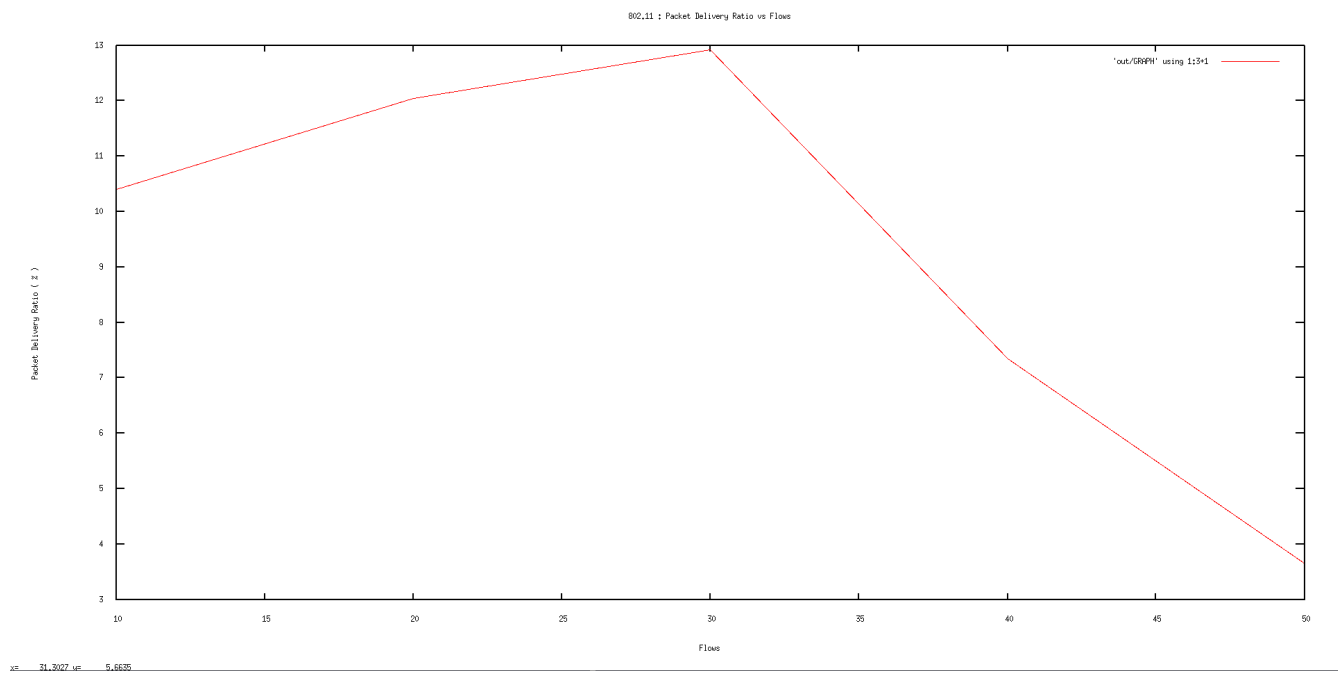
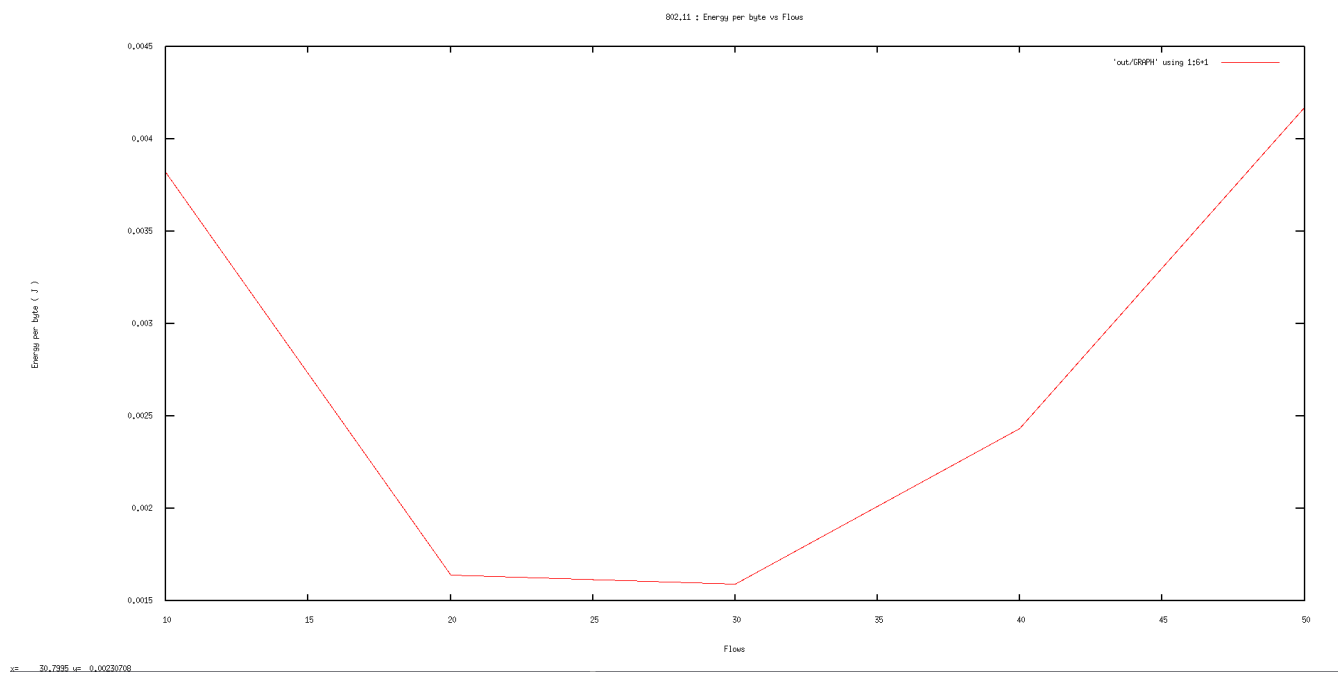


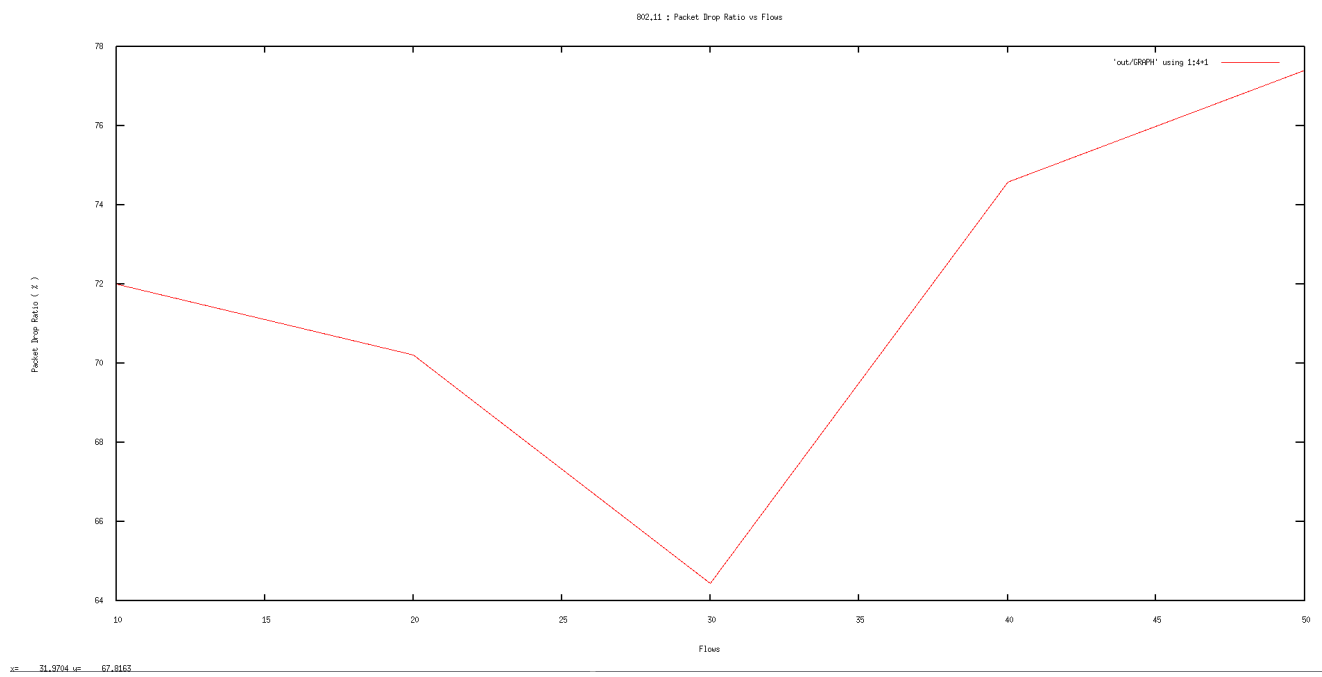


Variation in Number of Flow

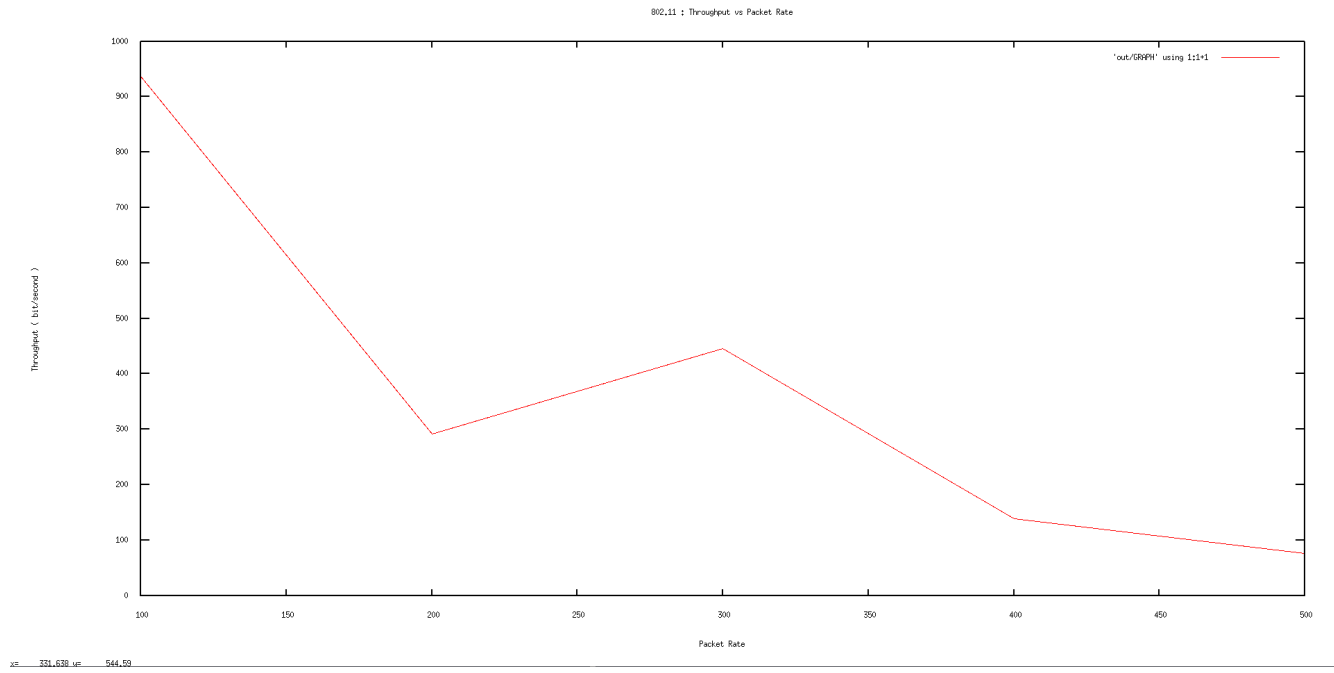


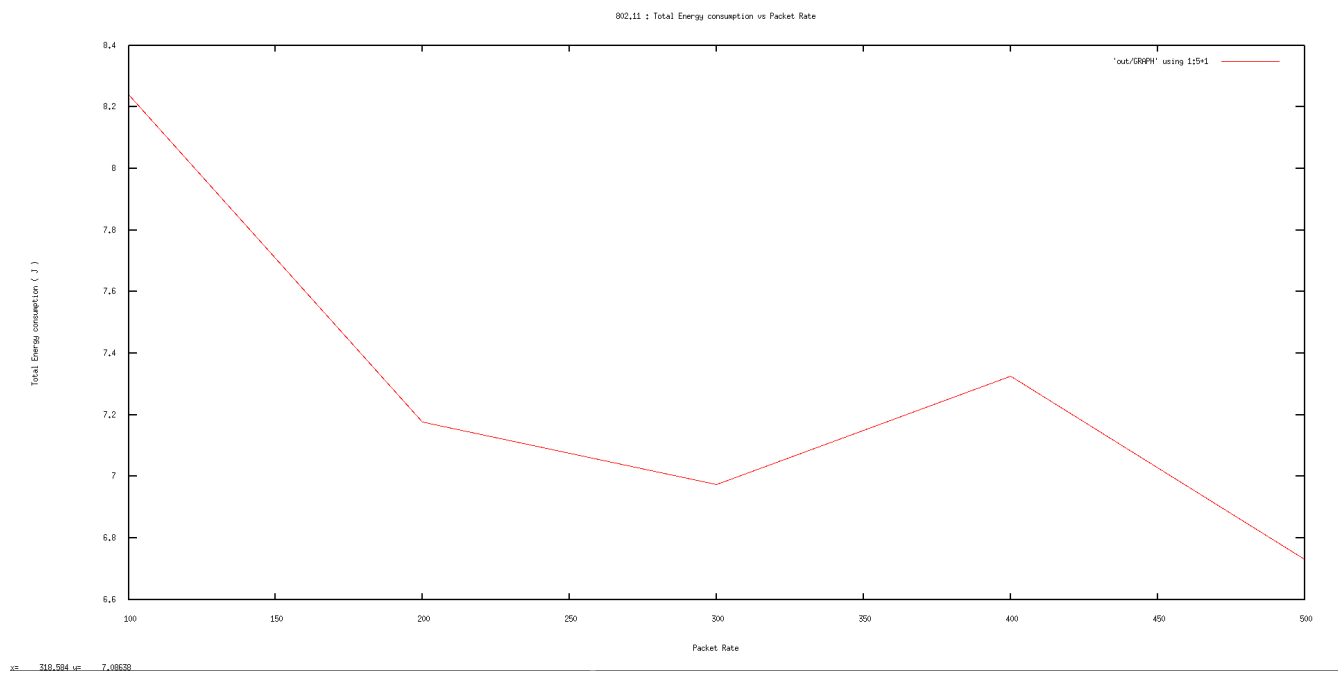
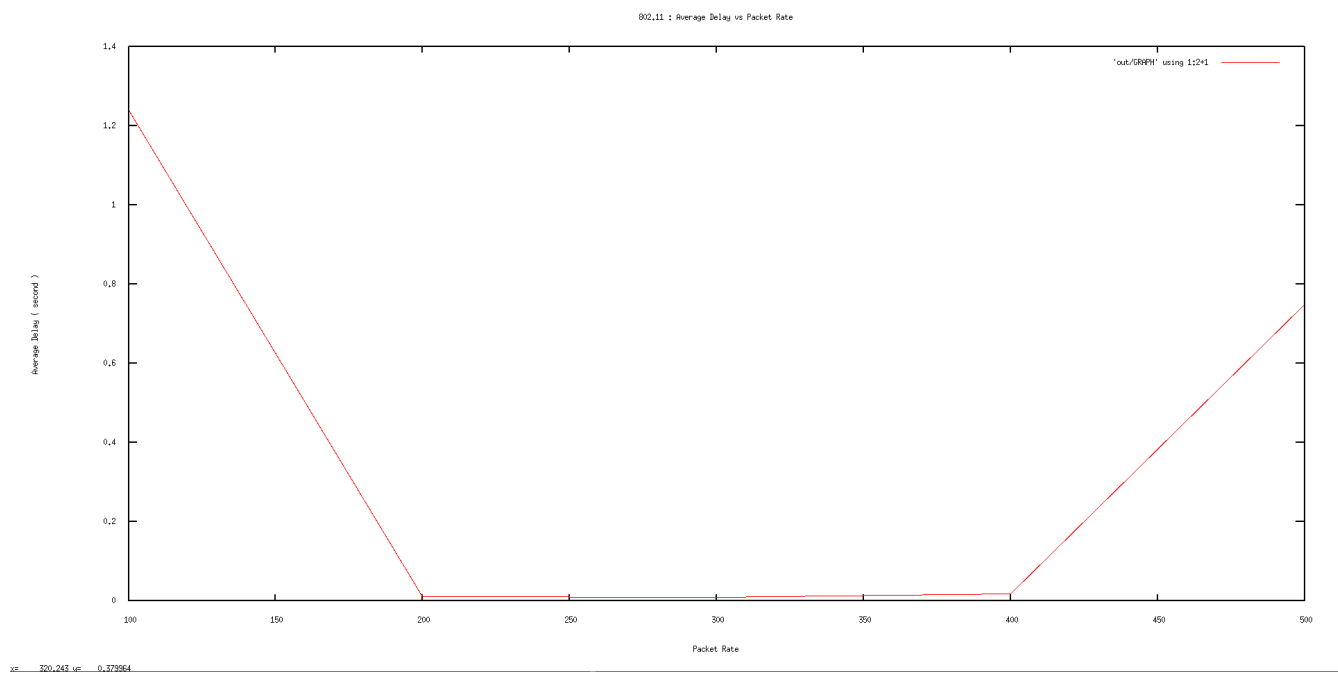


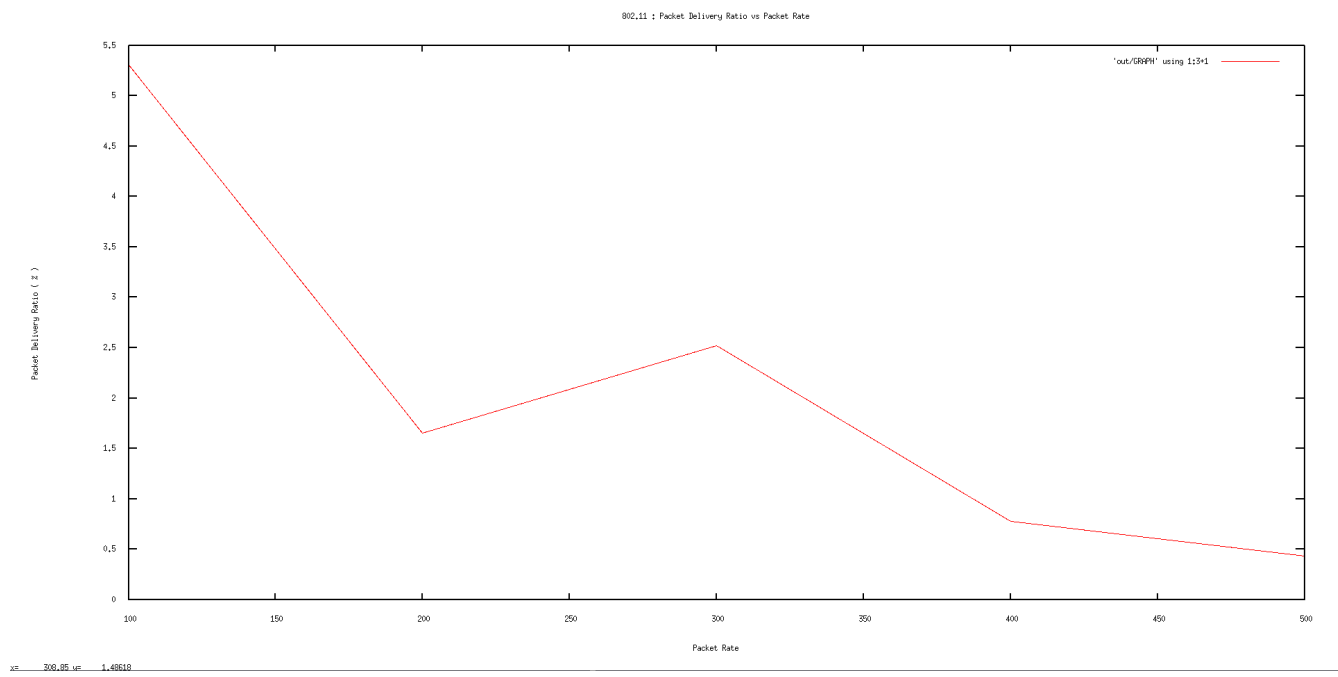
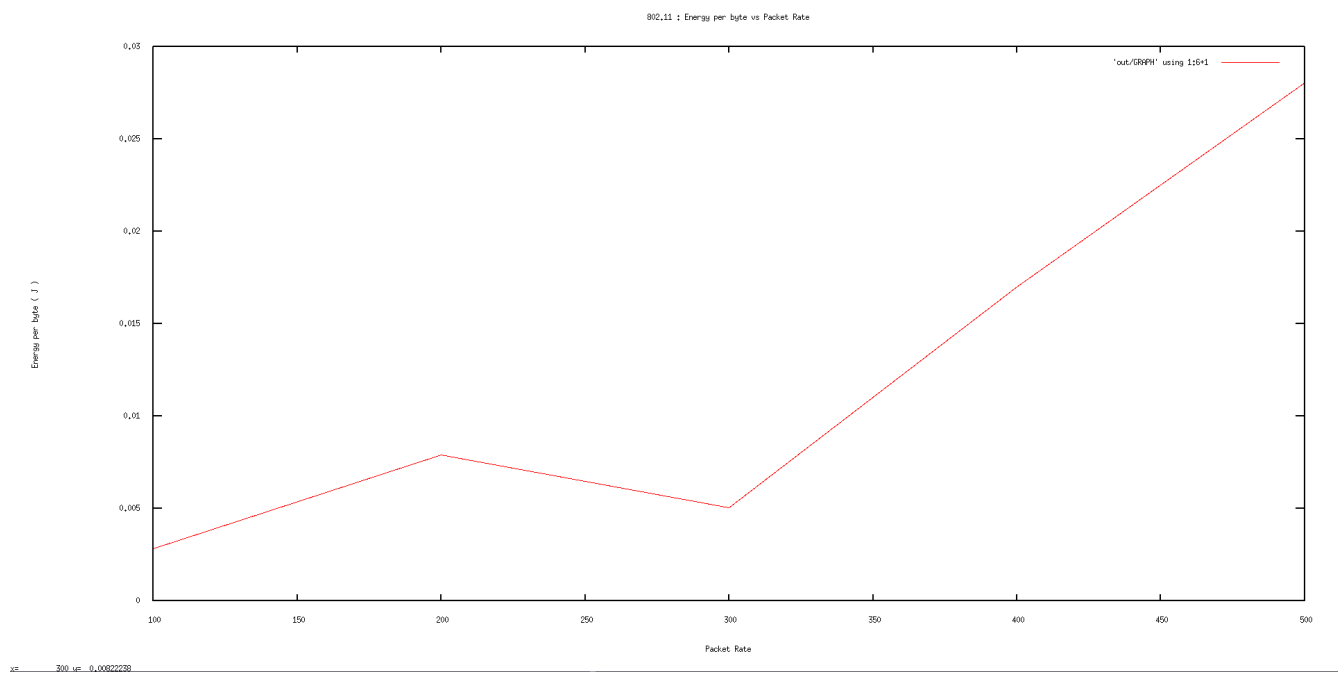


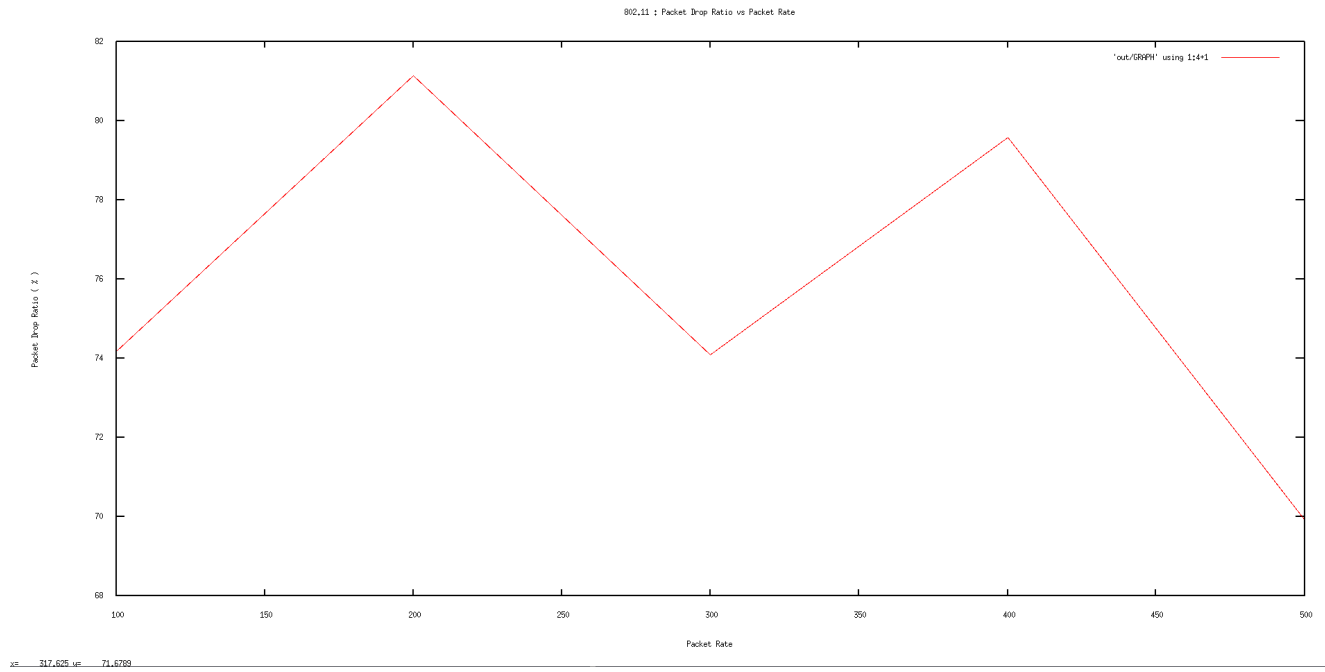


Variation in Packets Sent per Second

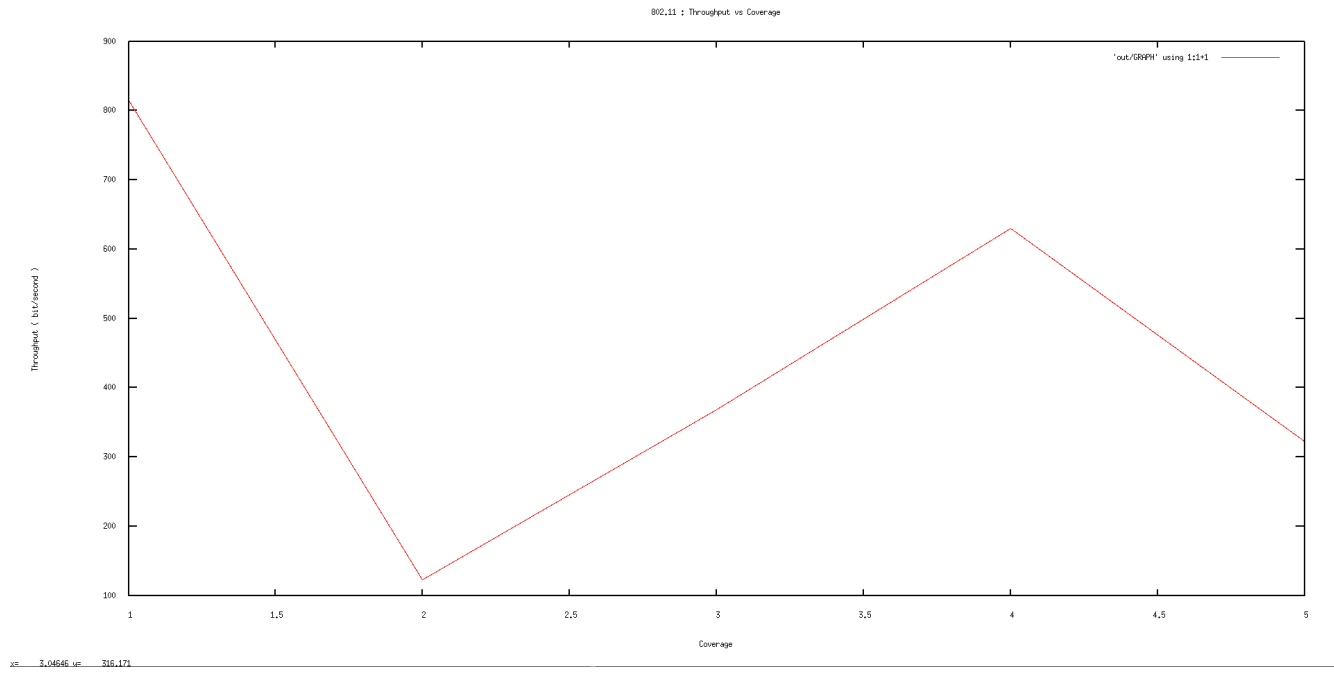


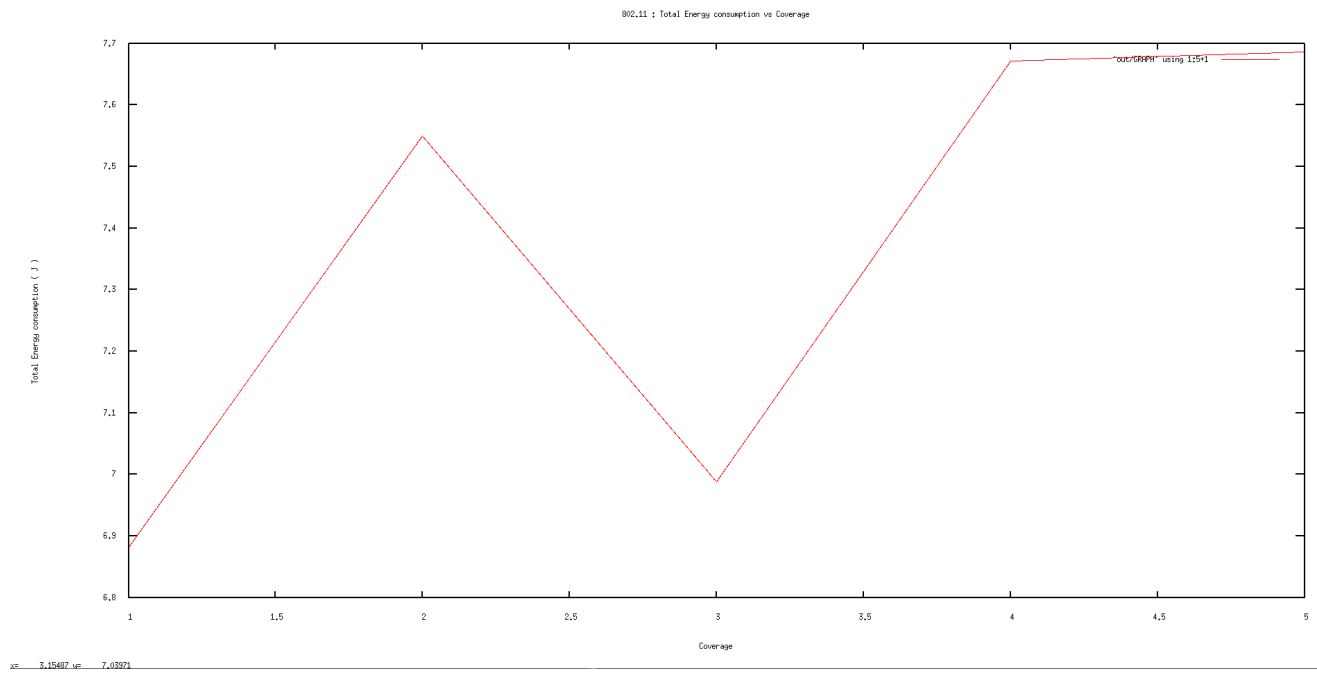
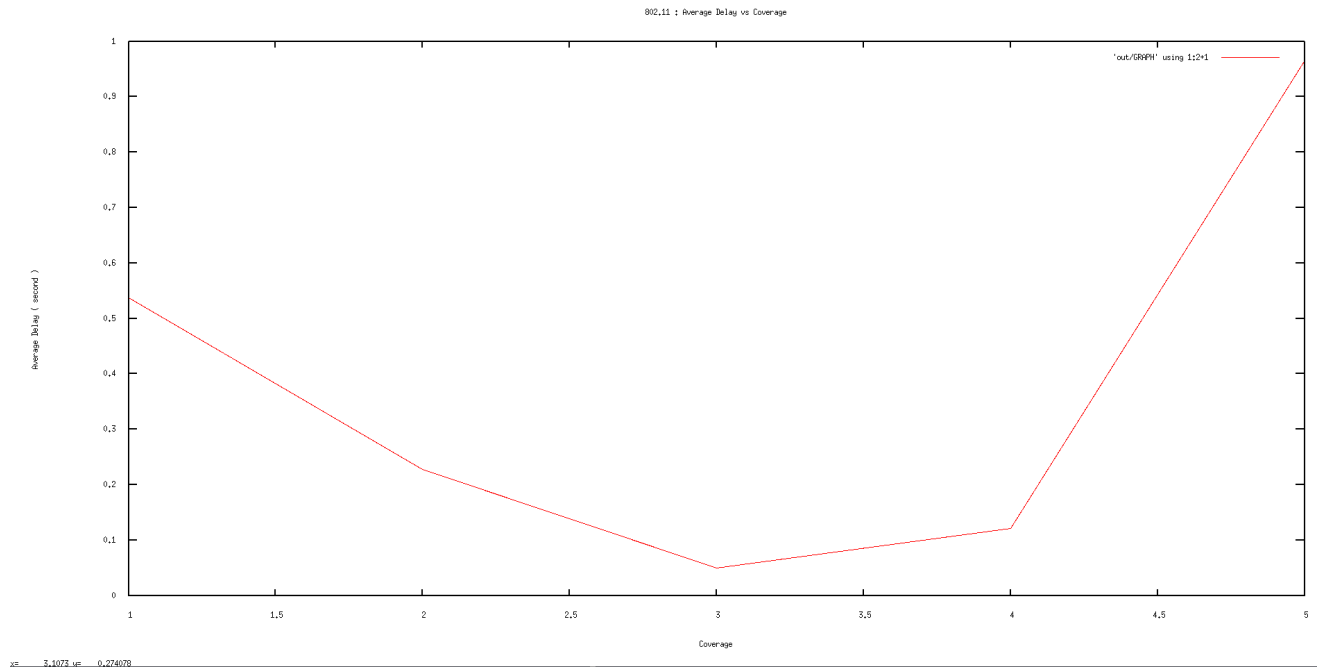


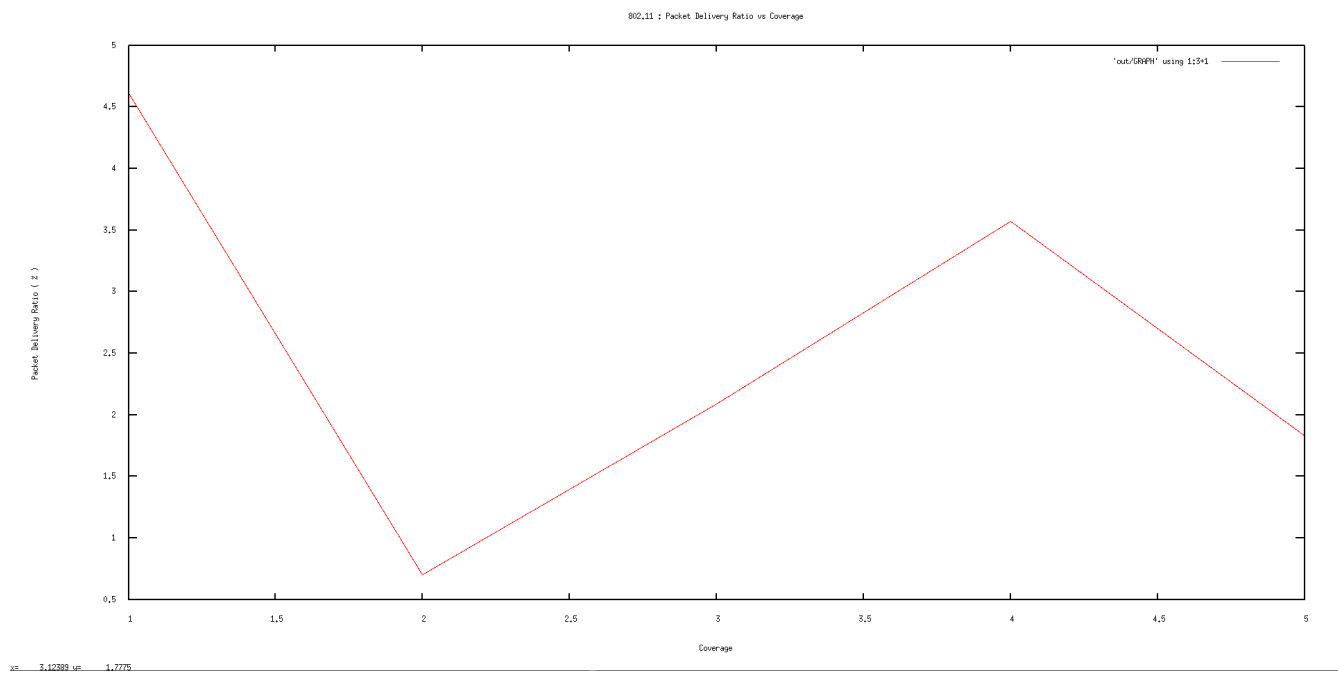
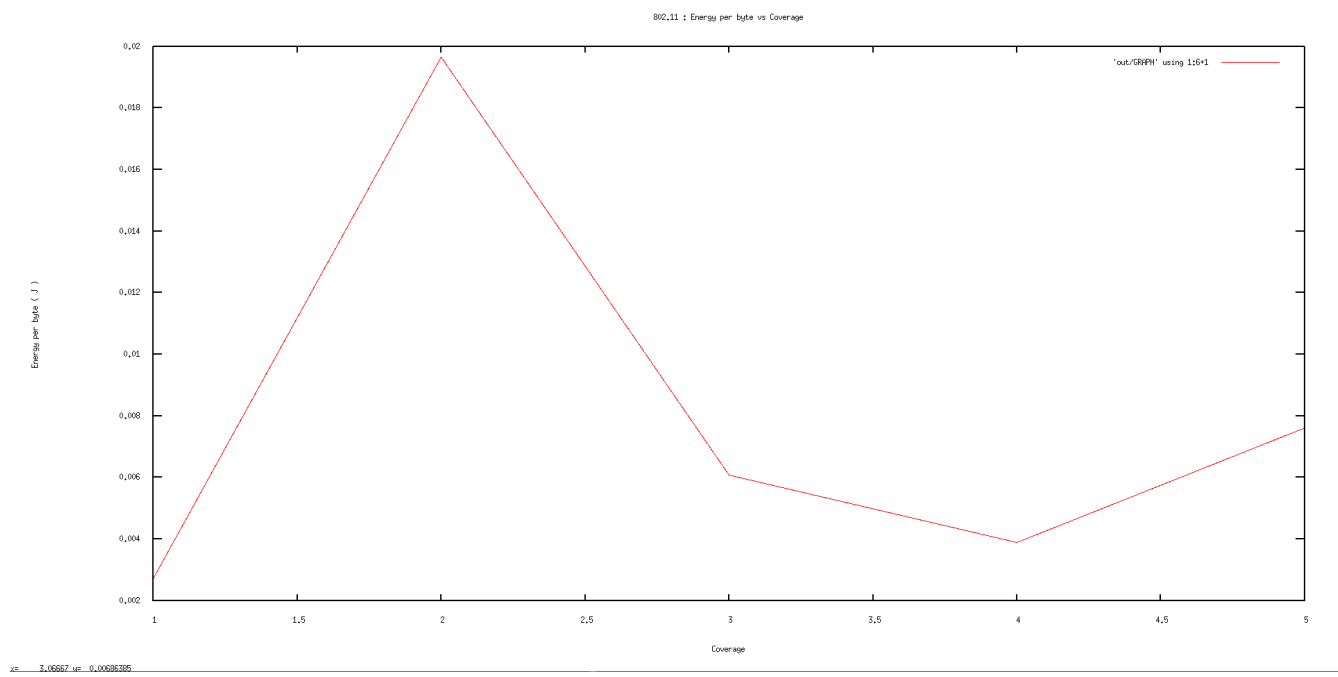


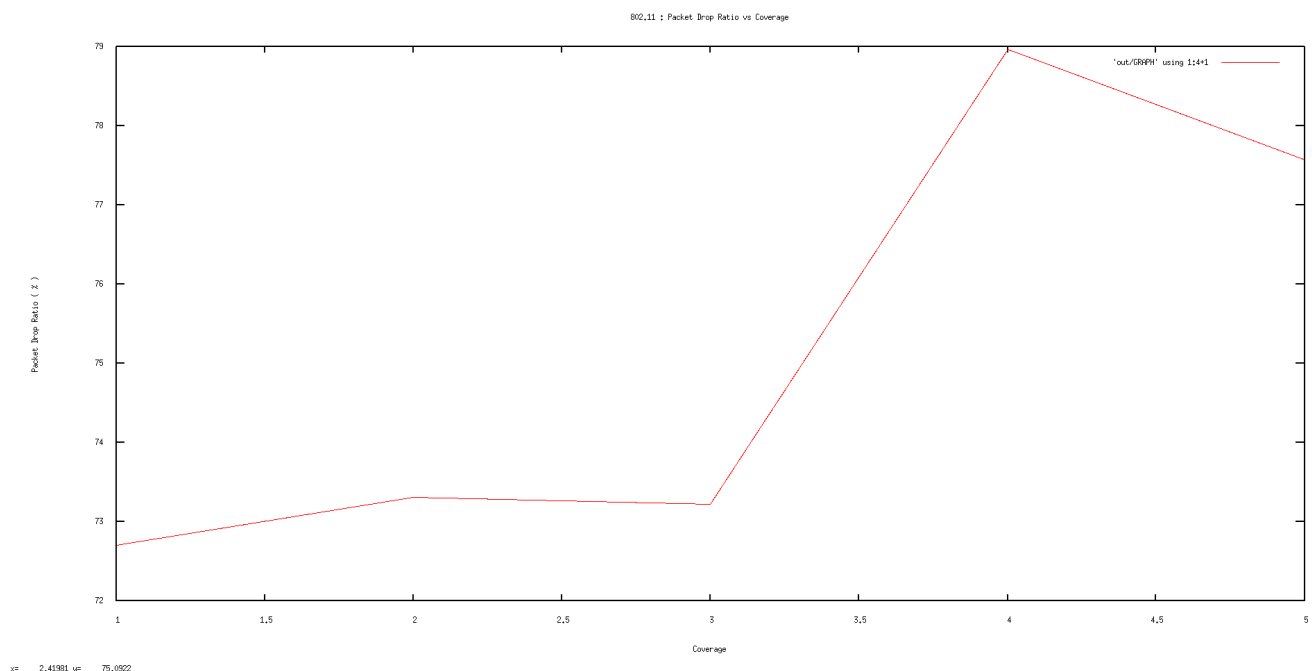


Variation in Coverage









5 Summary Findings

1. Packet delivery ratio is much higher in 802.11 than 802.15.4. Not only that throughput is also greater.
2. 802.15.4 works better when AODV is used
3. The maximum application packet size in 802.15.4 is 100 bytes. DSR adds larger overhead than AODV and hence makes the overall packet size larger than AODV does, so AODV works better
4. The modification in DSDV that converted in to DVR proves that DSDV works better than DVR
5. The modification in tcp.cc improves performance, retransmission of packet decreases