

# CSE-322 (Networking Sessional)

## Project Report

“TCP CERL: congestion control enhancement over wireless networks”

Name: Mohammad Shamim Ahsan

Student ID: 1705097

Department: CSE

Section: B

Supervisor:

Md. Tareq Mahmood

Lecturer,

CSE, BUET.

# Contents

<b>Section A:</b> Assigned Networks	3
<b>Section B:</b> Network topologies under simulation	3
<b>Section C:</b> Parameter under variation	4
<b>Section D:</b> Overview of the proposed algorithm	4
<b>Section E:</b> Modifications made in the simulator	6
<b>Section F:</b> Results with graphs (For Task A)	
i)    Wireless high-rate (e.g., 802.11) (mobile) --- Wi-Fi	7
ii)   Wireless low-rate (e.g., 802.15.4) (mobile) --- LR-WPAN	21
<b>Section G:</b> Explanation of results of Task A	34
<b>Section H:</b> Results & Comparison with graphs (Task B)	
i)    Wireless high-rate (e.g., 802.11) (mobile) --- Wi-Fi	35
ii)   Wireless low-rate (e.g., 802.15.4) (mobile) --- LR-WPAN	45
<b>Section I:</b> Explanation of results of Task B and Comparison with TCP-NewReno	55
<b>Section J:</b> Comparing results with Paper	56

## Section A: Assigned Networks

(ID%6=1)

- 1) Wireless high-rate (e.g., 802.11) (mobile) --- Wi-Fi
- 2) Wireless low-rate (e.g., 802.15.4) (mobile) --- LR-WPAN

## Section B: Network topologies under simulation

In this project, “Ad-hoc” topology is used. It is a wireless topology where devices are connected without using any additional network infrastructure devices like a wireless access point.

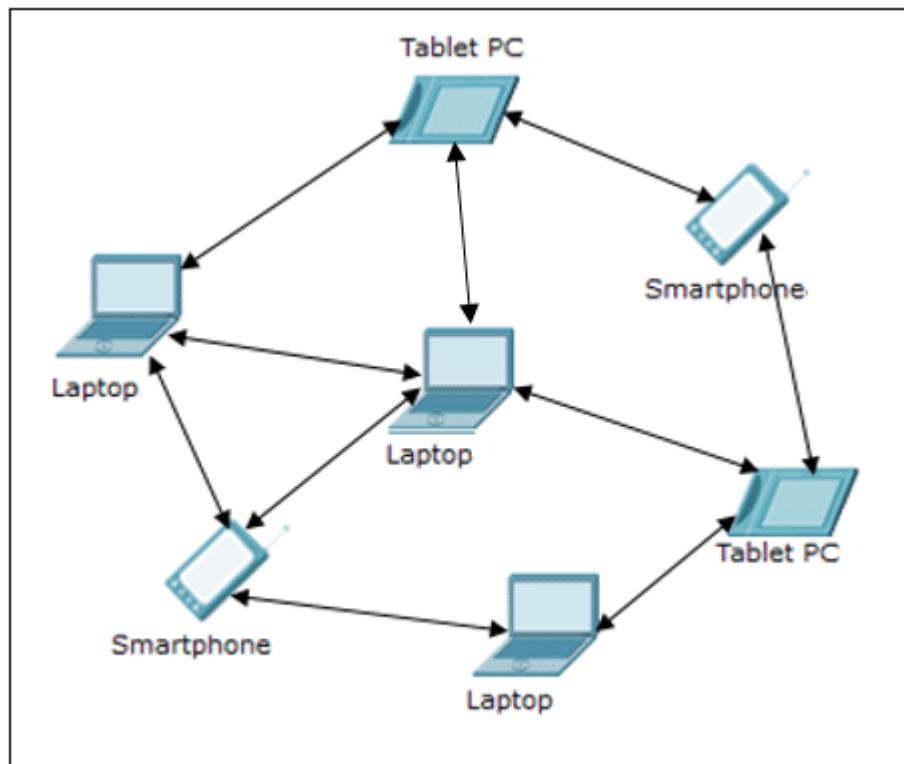


Figure: Mobile Ad-hoc network topology.

## Section C: Parameter under variation

- 1) Number of nodes (e.g., 30, 40, 60, 80, 100)
- 2) Number of packets per second (e.g., 100, 200, 300, 400, 500)
- 3) Speed of node (as both network needs mobility) (e.g., 5 m/s, 10 m/s, 15 m/s, 20 m/s, 25 m/s)
- 4) Number of flows (e.g., 6, 10, 12, 14, 16)

## Section D: Overview of the proposed algorithm

TCP CERL is a sender-side modification of TCP Reno. CERL and Veno are similar in concept. Specifically, they both attempt to distinguish between random loss and congestive loss to allow for the independent treatment of each case. However, the mechanisms utilized by CERL differ greatly from those used in Veno, as should become clear in the remaining parts of this section.

### Algorithm:

Step-1: Calculating the bottleneck queue length(L)

$$\begin{aligned} L &= (\text{RTT} - T) * \text{Bandwidth} \\ &= (\text{RTT} - T) * (\text{cwnd} / \text{RTT}), \text{ where } T \text{ is the smallest RTT} \end{aligned}$$

Step-2: Calculating the dynamic queue length threshold(N)

$$N = A * L_{\max}, \text{ where } A \text{ is constant}$$

Step-3: Each time a new RTT is received, values of T, L, L<sub>max</sub>, N are updated.

Step-4: Slow start and congestion avoidance is same as Reno.

Step-4: When 3 duplicate acknowledgements are received,

- i) If L < N, CERL will assume this segment loss is “random loss” rather than congestion loss. The lost segment will

- be retransmitted, but the congestion window and slow start threshold will not be reduced.
- ii) Otherwise, CERL will assume the loss is caused by congestion, and the congestion window and slow start threshold will be reduced as in Reno. However, multiple segment losses in one window of data will only reduce the congestion window once.

```

IF (l ≥ N&highestAck-1>lastDecMaxSentSeqno)
THEN
    ssthresh=min(cwnd,awnd)/2
    IF (ssthresh<2*segsize)
        THEN
            ssthresh=2*segsize
        END IF
    cwnd==ssthresh+3*segsize
    lastDecMaxSentSeqno=MaxSentSeqno
ELSE
    oldcwnd=cwnd
    cwnd=cwnd+3*segsize
END IF

```

Step-5: Small modification in fast recovery algorithm code that handles window deflation.

```

IF (oldcwnd > 0)
THEN
    cwnd=oldcwnd
    oldcwnd=-1.0
ELSE
    cwnd=ssthresh
END IF

```

## **Section E: Modifications made in the simulator**

As ns-3.35 uses tcp-newreno as root of all tcp congestion algorithm, tcp-cerl has been developed using tcp-newreno as basement.

### **Modification:**

First of all, two new files – “tcp-cerl.h” and “tcp-cerl.cc” are added in the “/ns-3.35/src/internet/model” folder.

Secondly, four new files – “wifi-tcp-reno.cc”, “wifi-tcp-cerl.cc”, “Irwpn-reno.cc” and “Irwpn-cerl.cc” are added in the “/ns-3.35/scratch” folder.

Thirdly, two new variables are added in the ‘TcpSocketState’ class in “/ns-3.35/src/internet/model/tcp-socket-state.h” file.

Fourthly, a small modification has been done in the ‘TcpSocketBase::EnterRecovery()’ function in “/ns-3.35/src/internet/model/tcp-socket-base.cc” file.

Last of all, but most importantly, to build CERL 'model/tcp-cerl.cc' is added in the “def build(bid)” method in “/ns-3.35/src/internet/wscript” file and “tcp-cerl.h” is added in “/ns-3.35/build/ns3” folder. A variable is made public in “/ns-3.35/src/internet/model/tcp-socket-base.h” file.

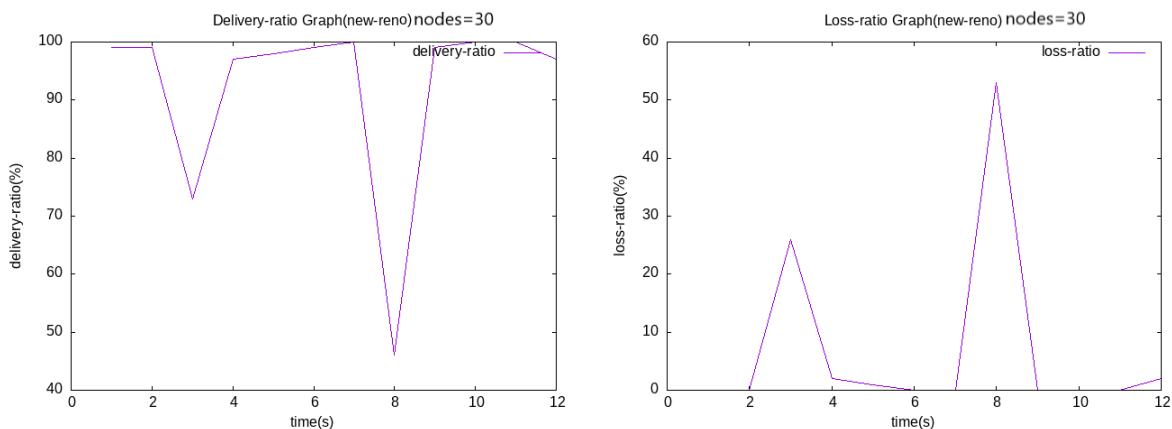
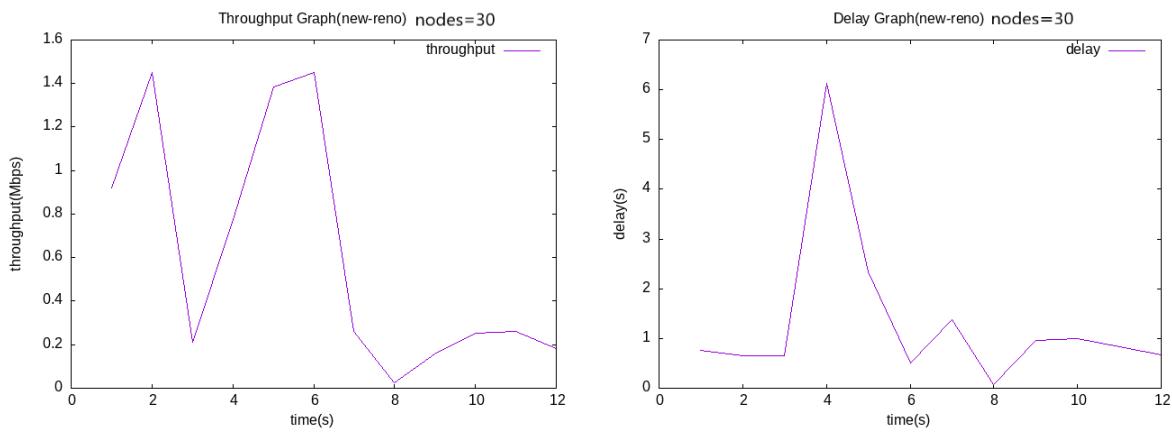
## Section F: Results with graphs (For Task A)

(Task A has been conducted with TCP New-Reno)

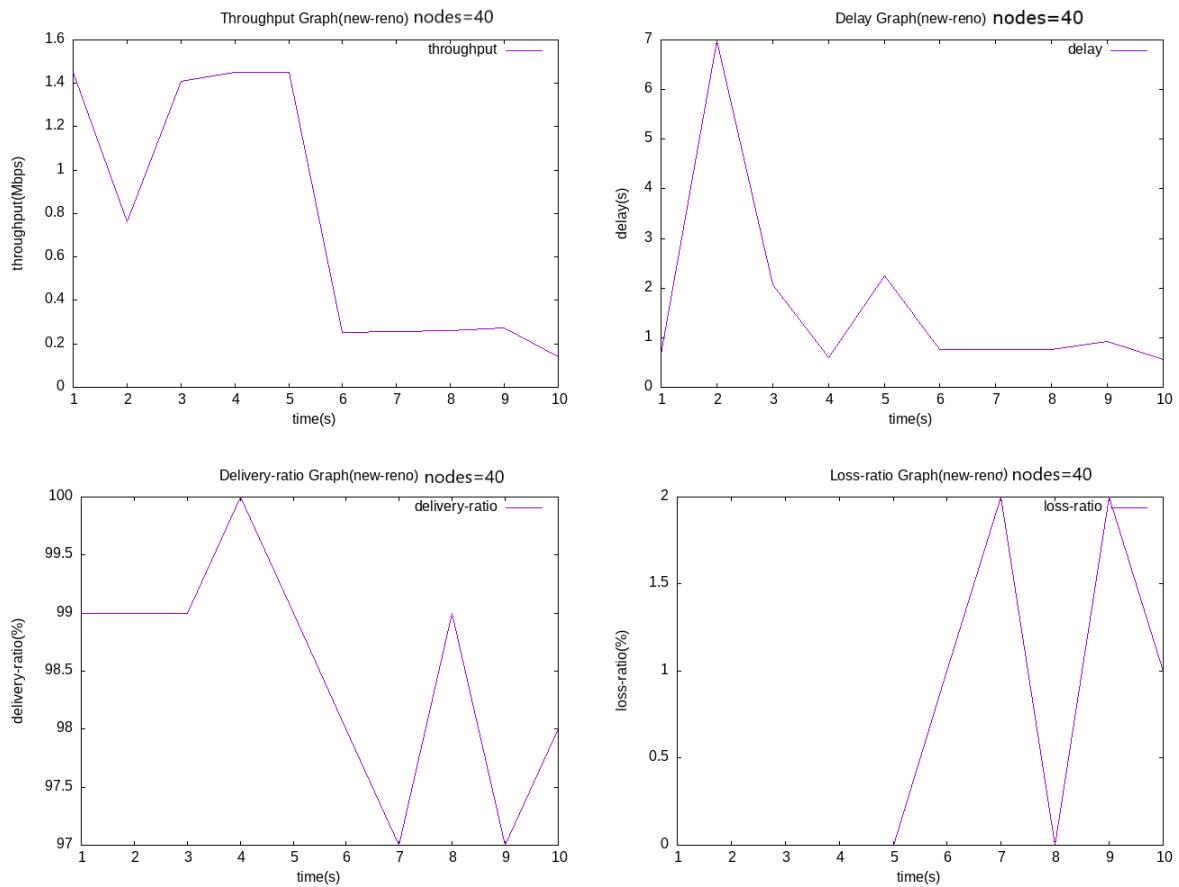
Wireless high-rate (e.g., 802.11) (mobile) --- Wi-Fi

**1)Varying number of nodes:**

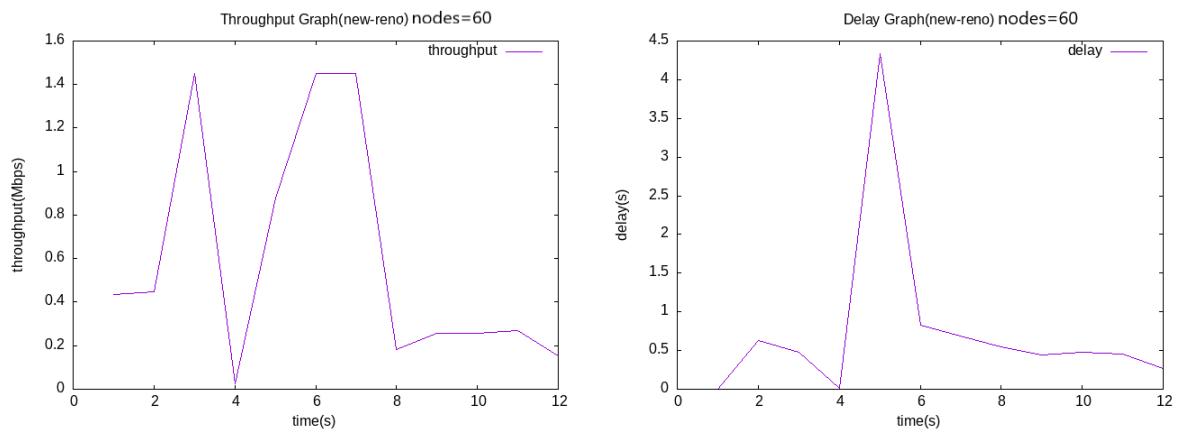
i) Number of nodes=30

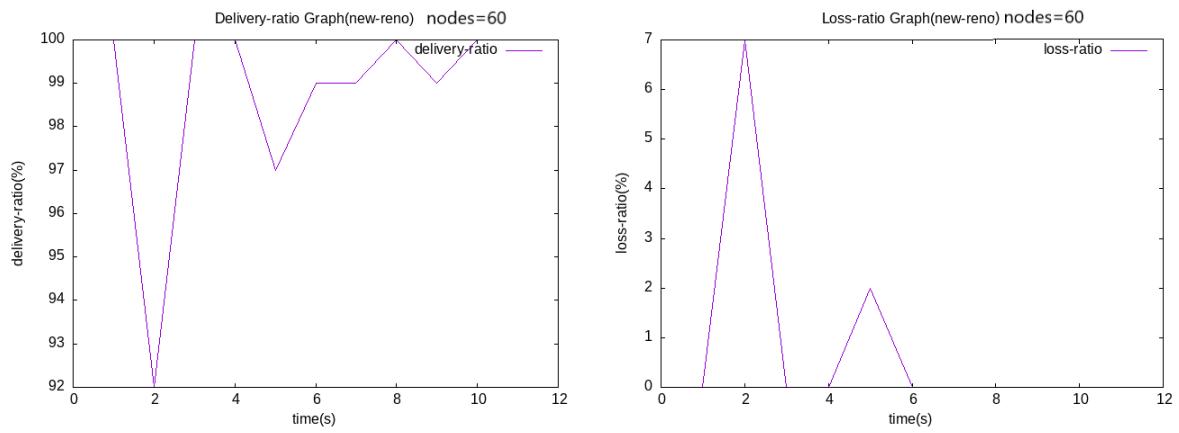


ii) Number of nodes=40

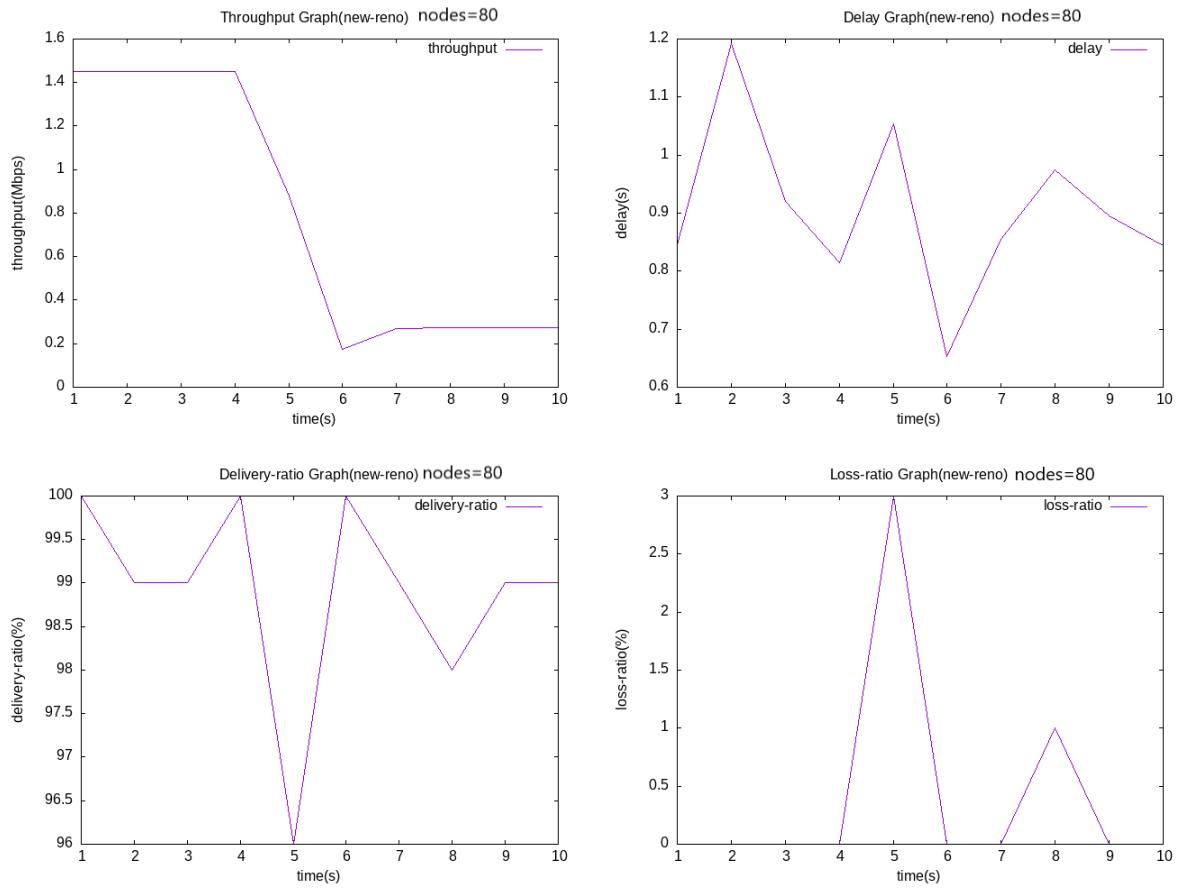


### iii) Number of nodes=60

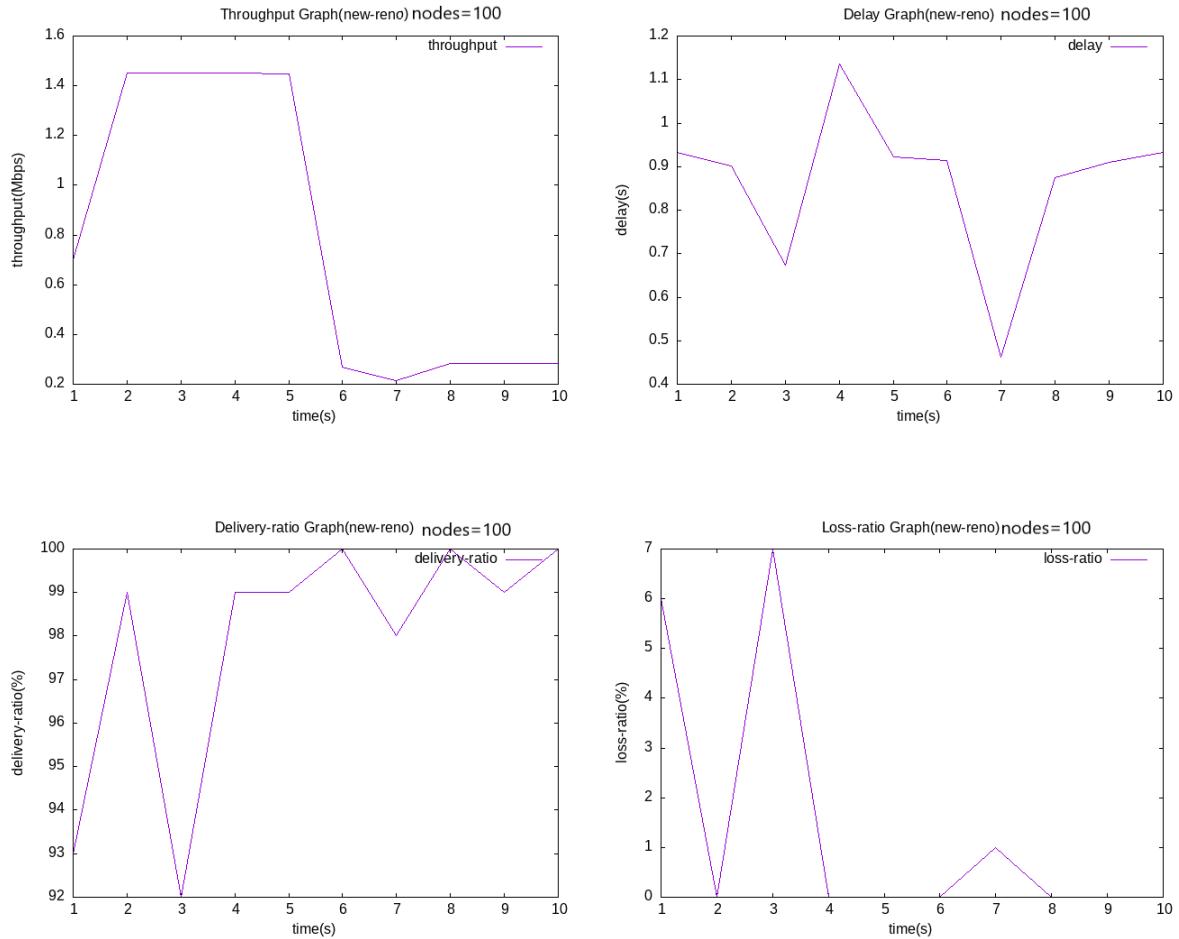




#### iv) Number of nodes=80

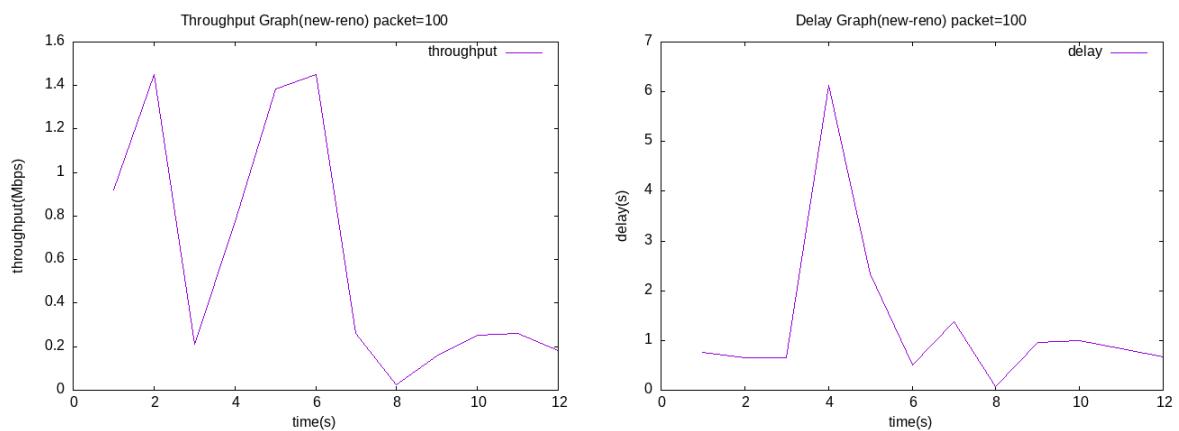


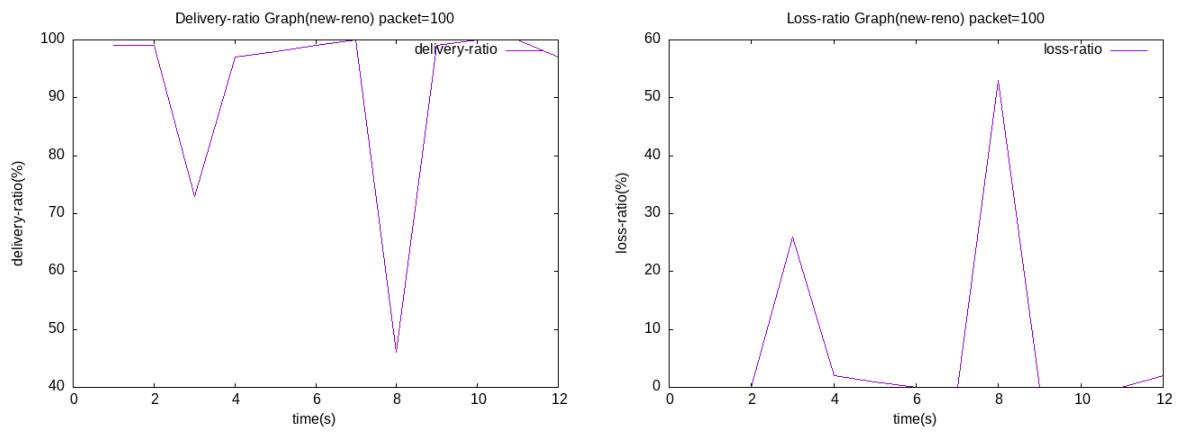
## v) Number of nodes=100



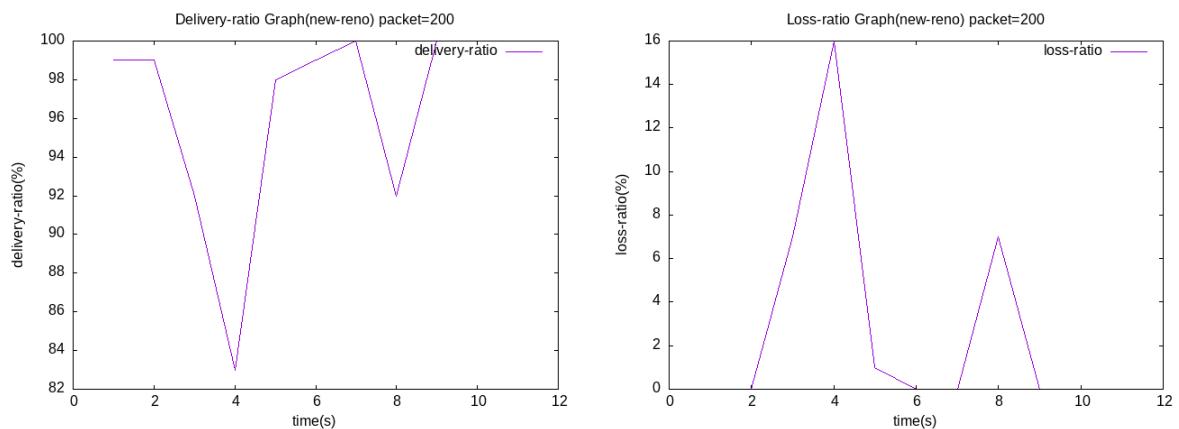
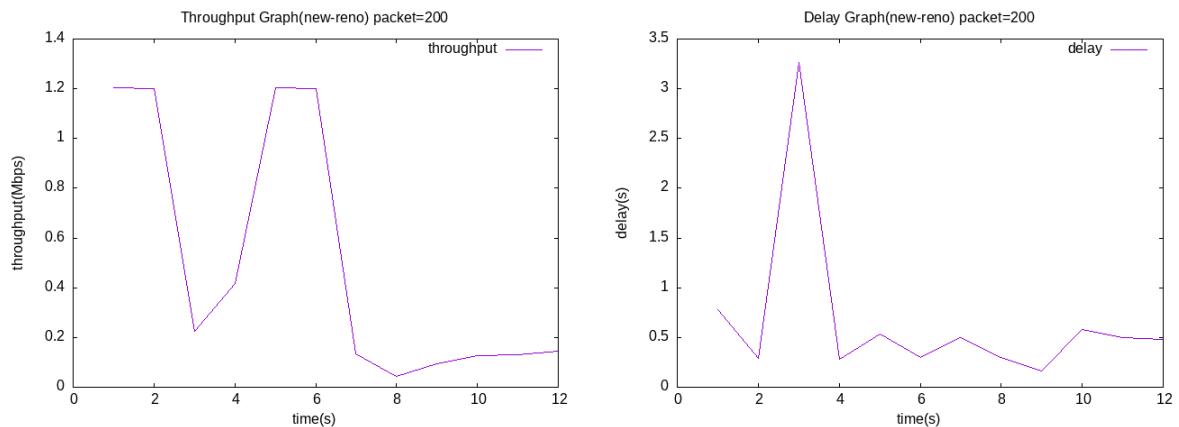
## 2) Varying number of packets per second:

### i) Number of packets=100

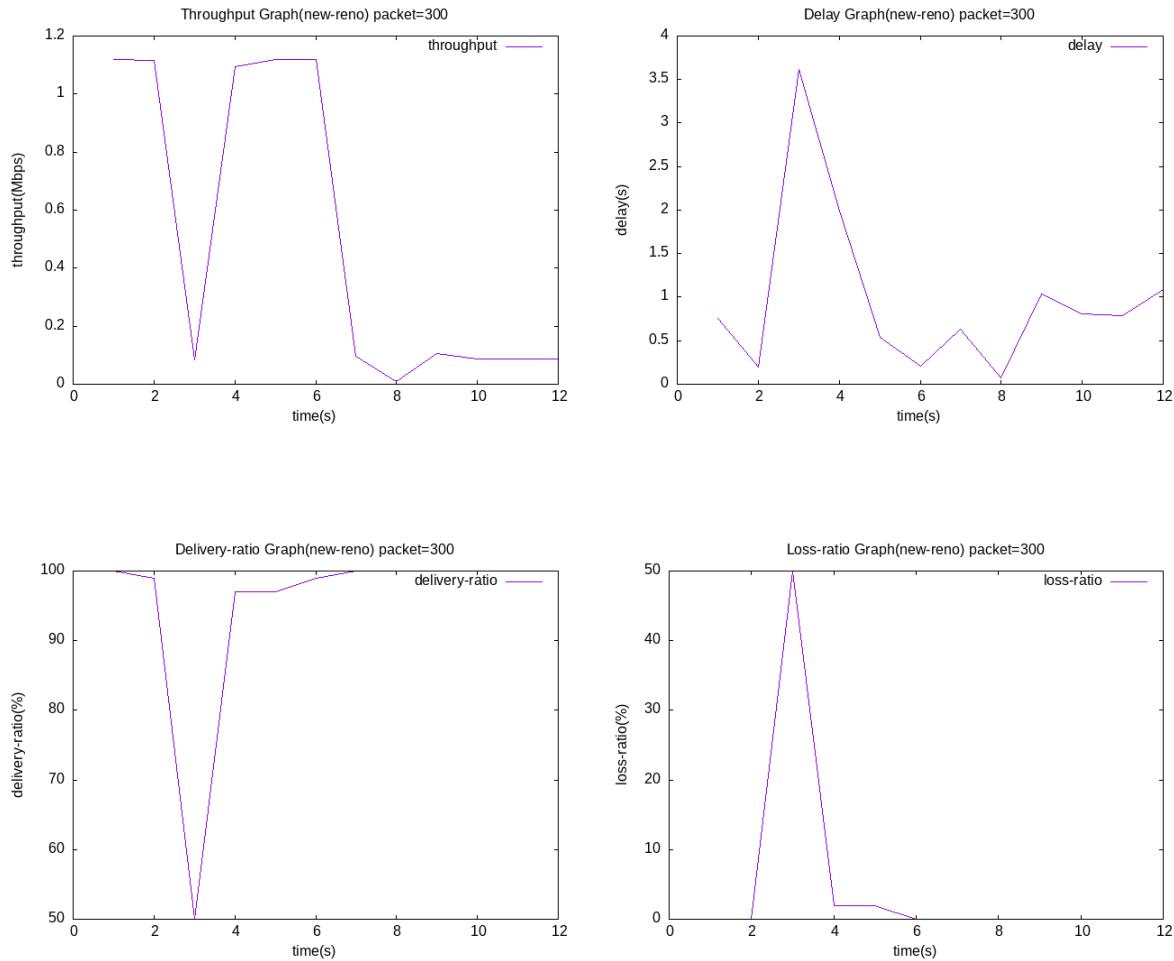




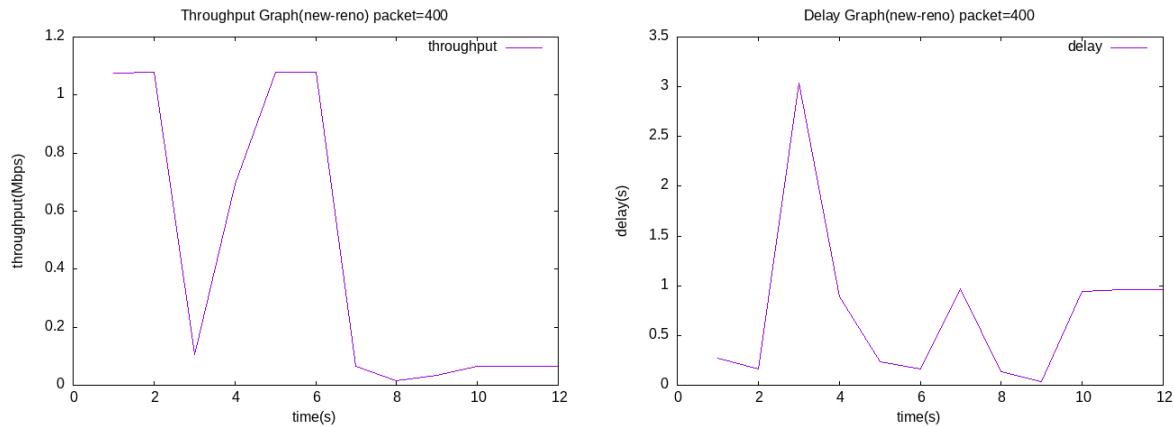
## ii) Number of packets=200

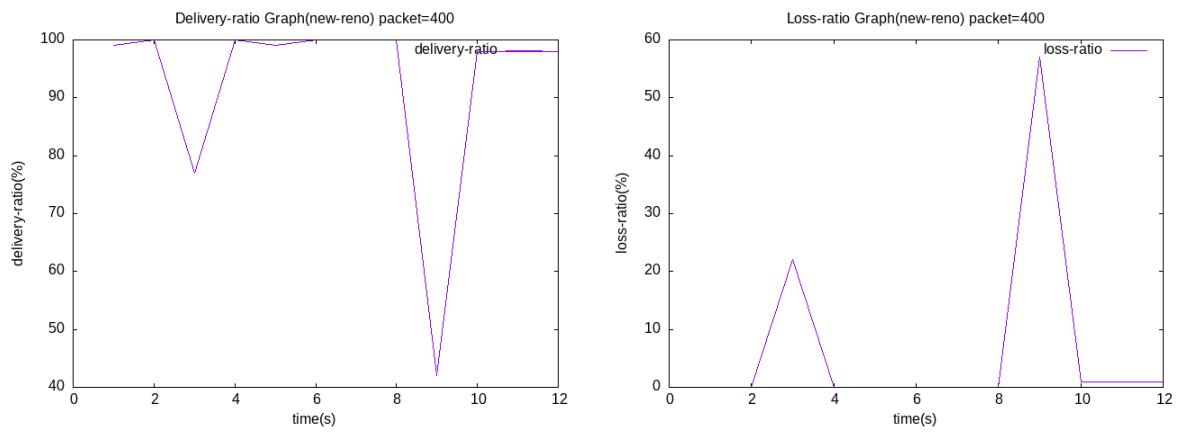


### iii) Number of packets=300

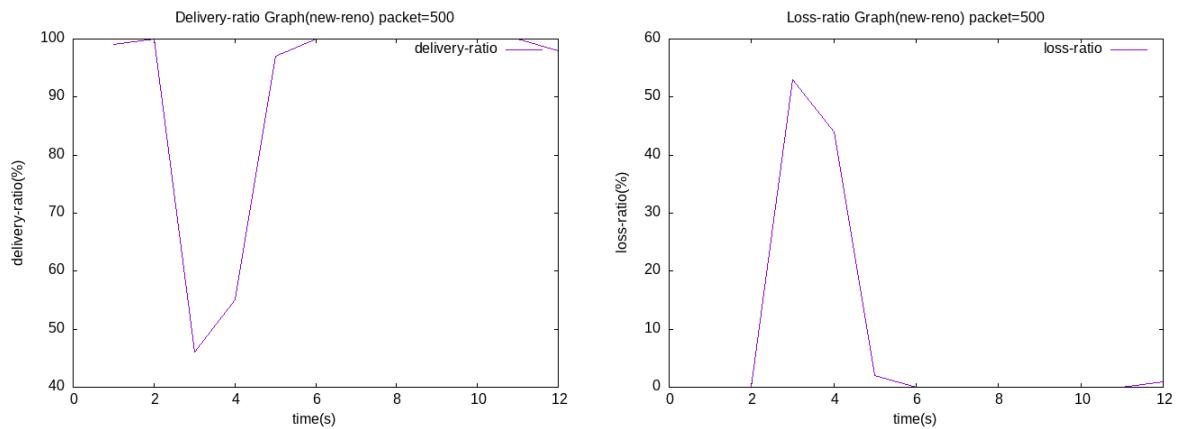
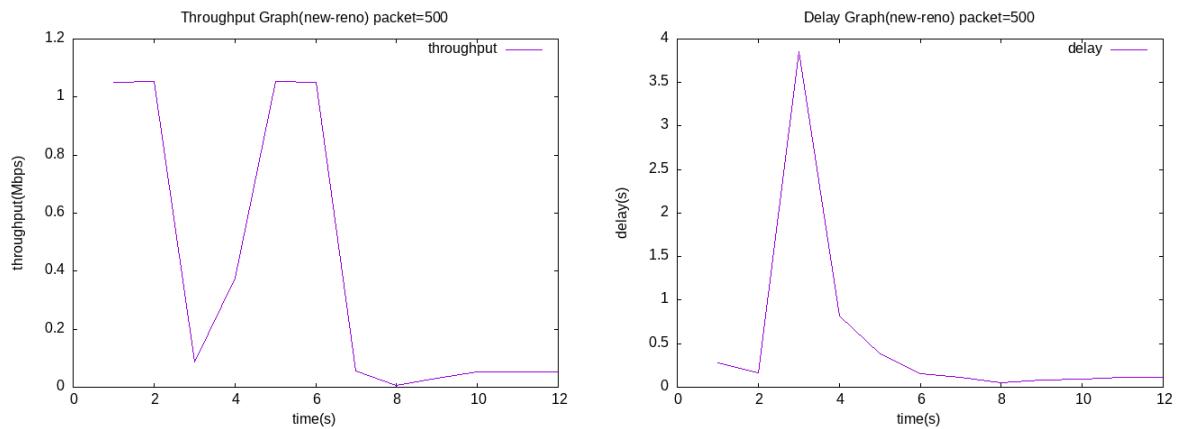


### iv) Number of packets=400



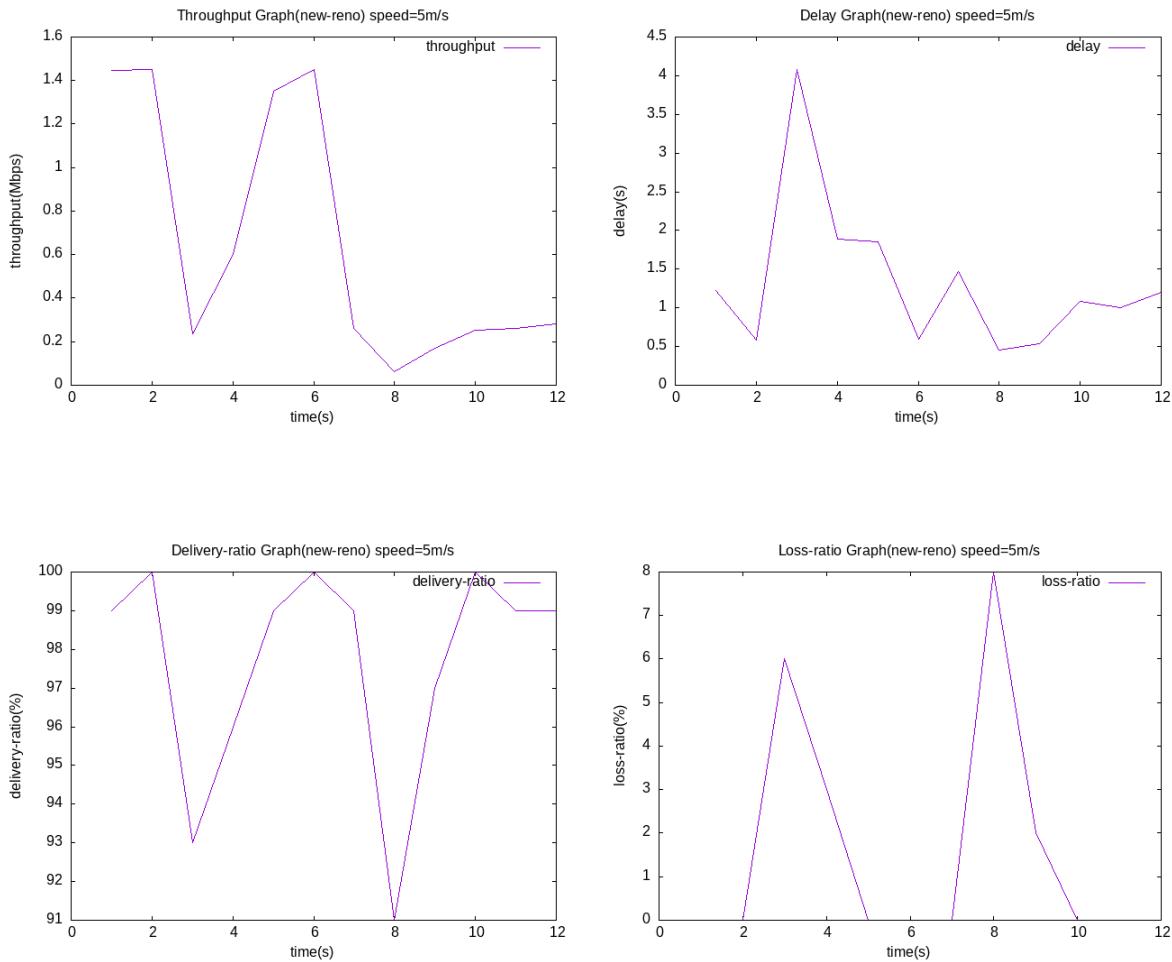


v) Number of packets=500

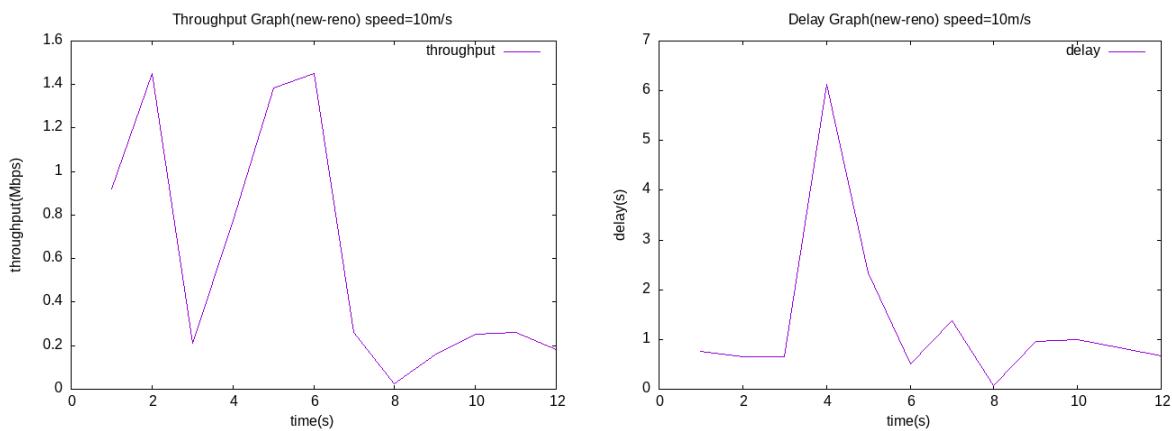


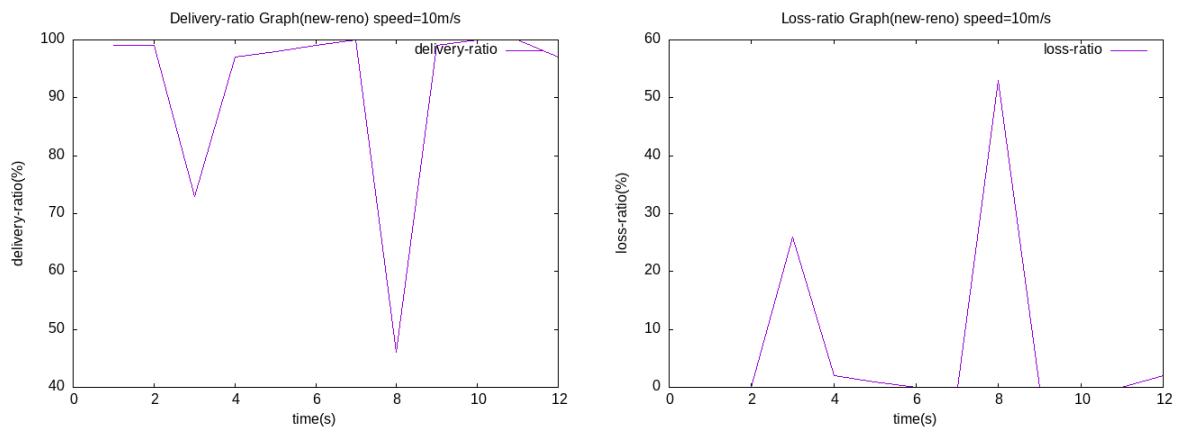
### 3) Varying speed of nodes:

i) Speed of nodes=5 m/s

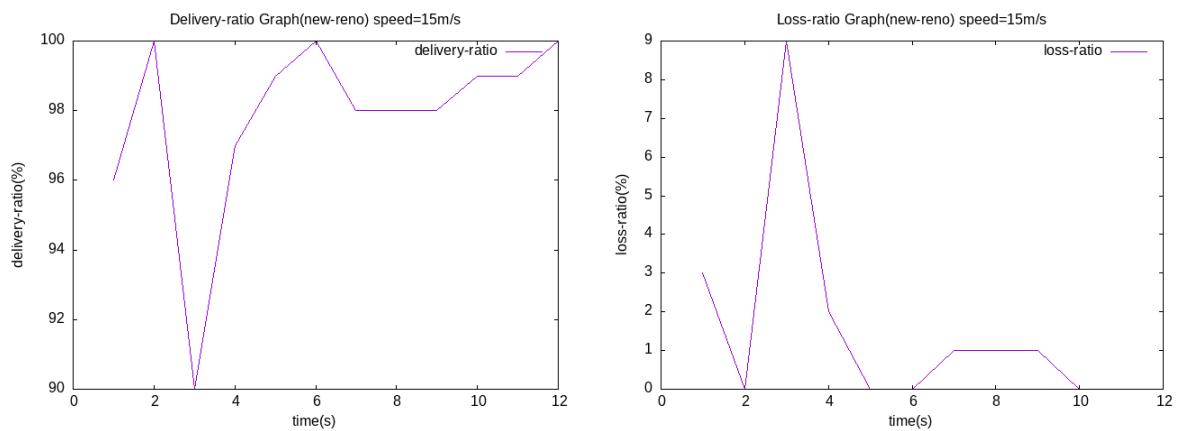
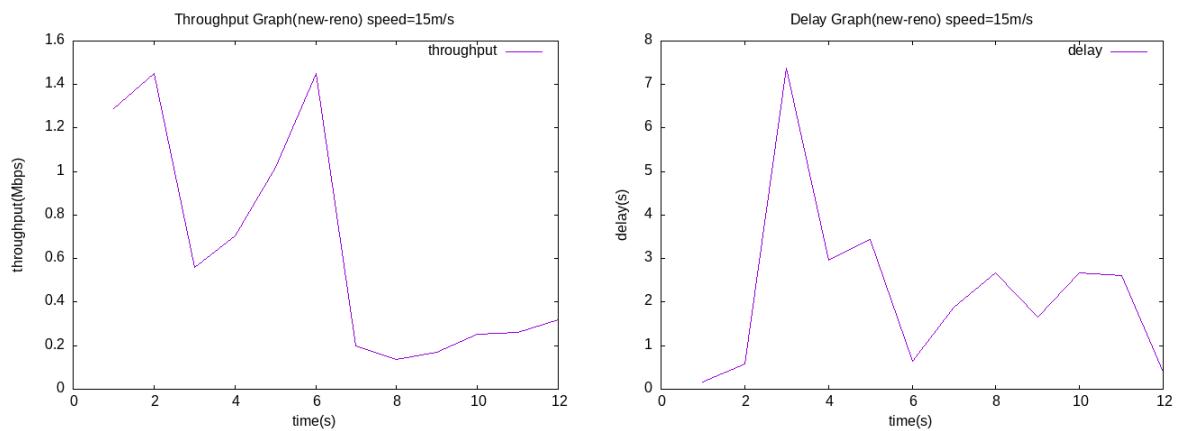


ii) Speed of nodes=10 m/s

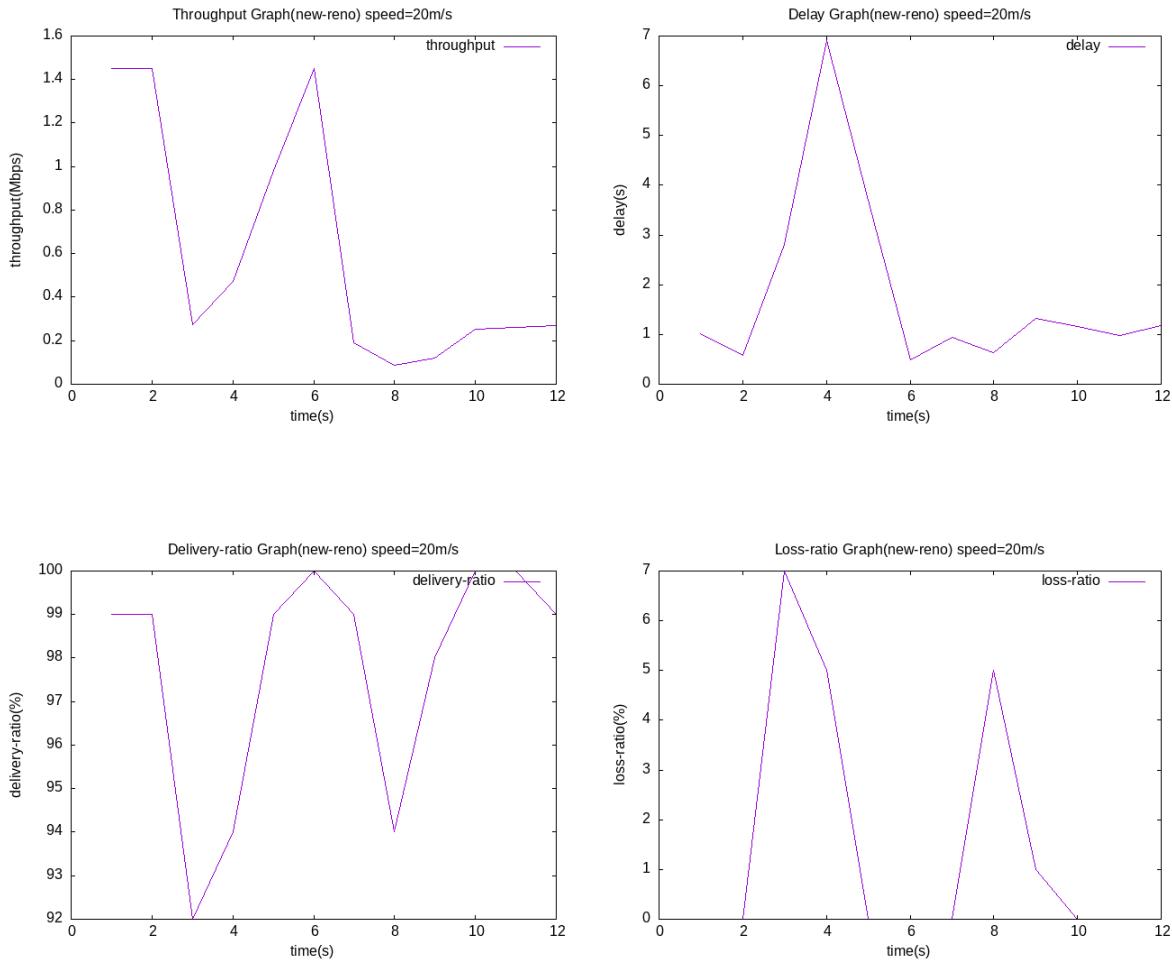




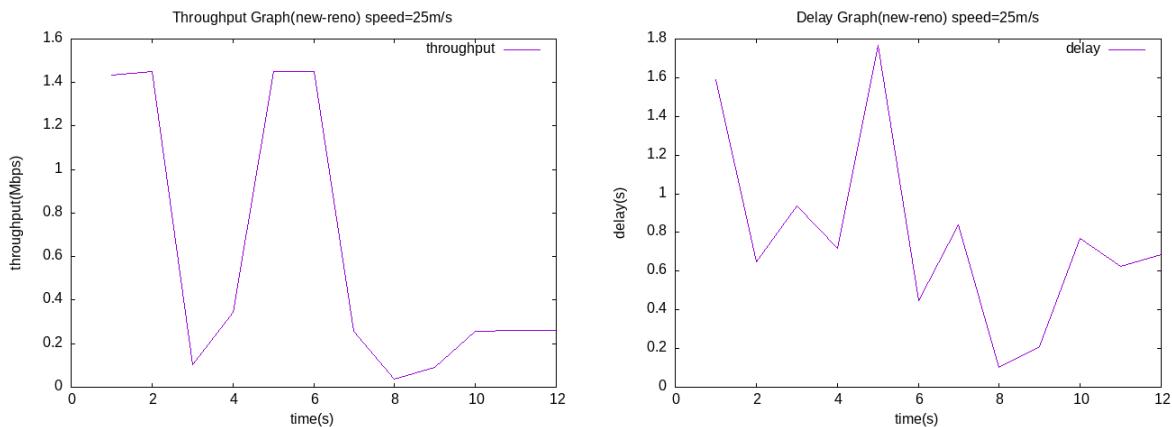
iii) Speed of nodes=15 m/s

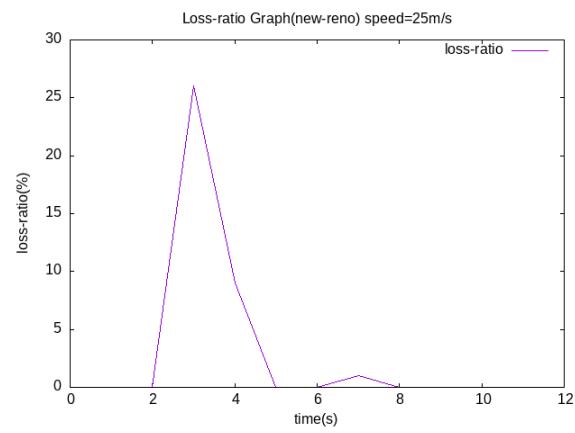
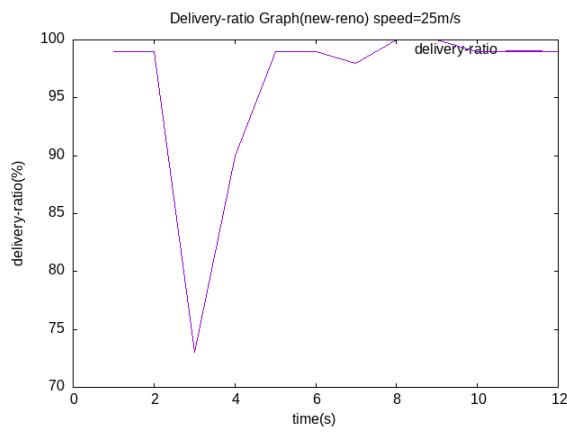


#### iv) Speed of nodes=20 m/s



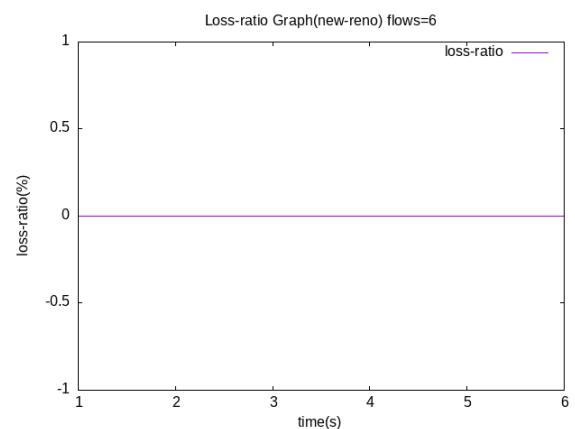
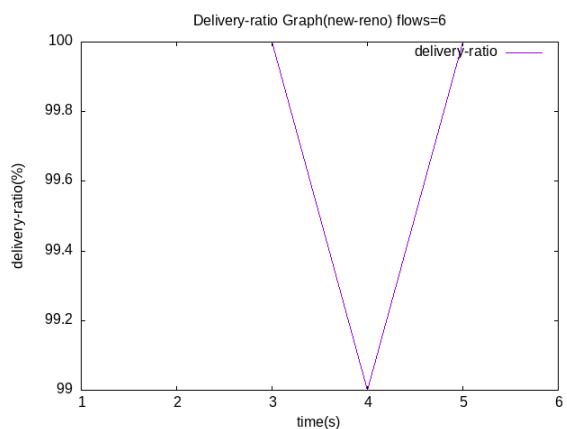
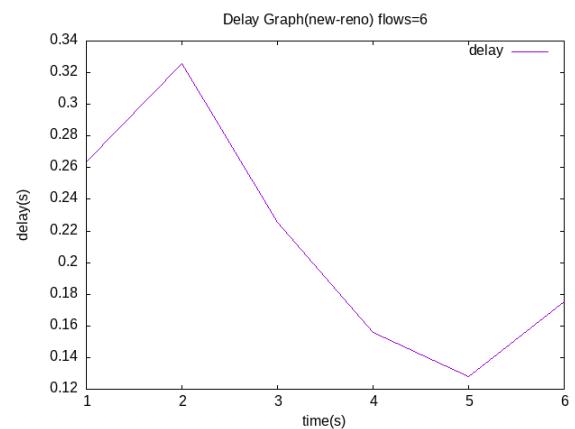
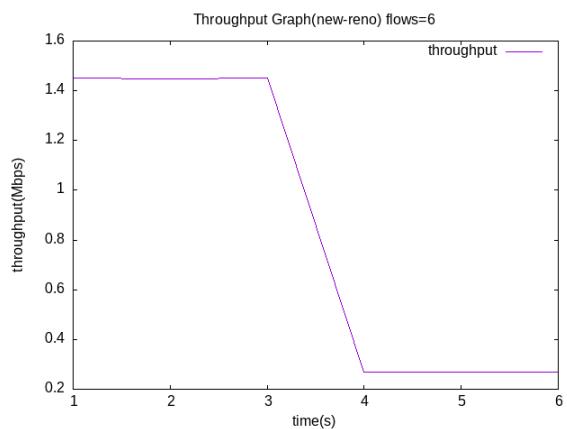
#### v) Speed of nodes=25 m/s



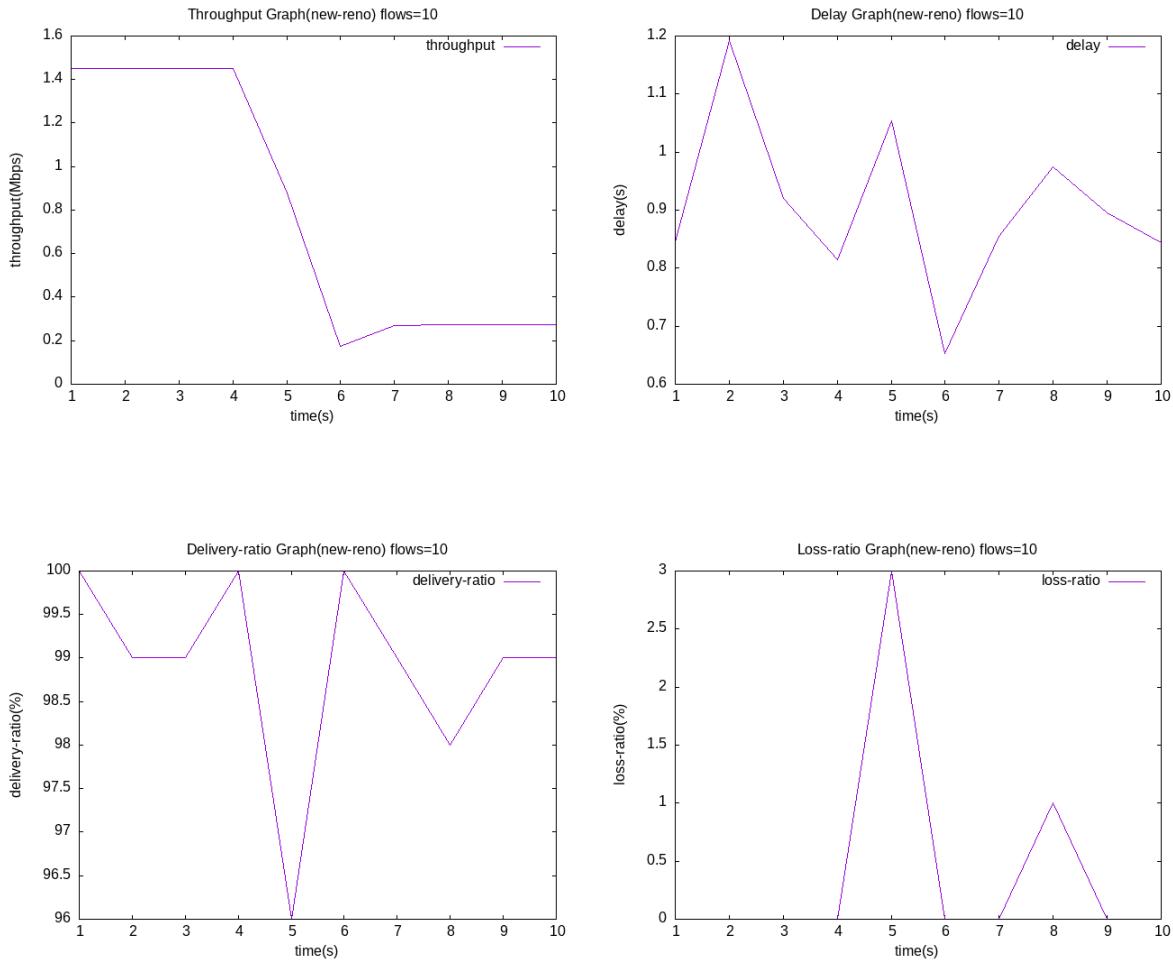


## 4)Varying number of flows:

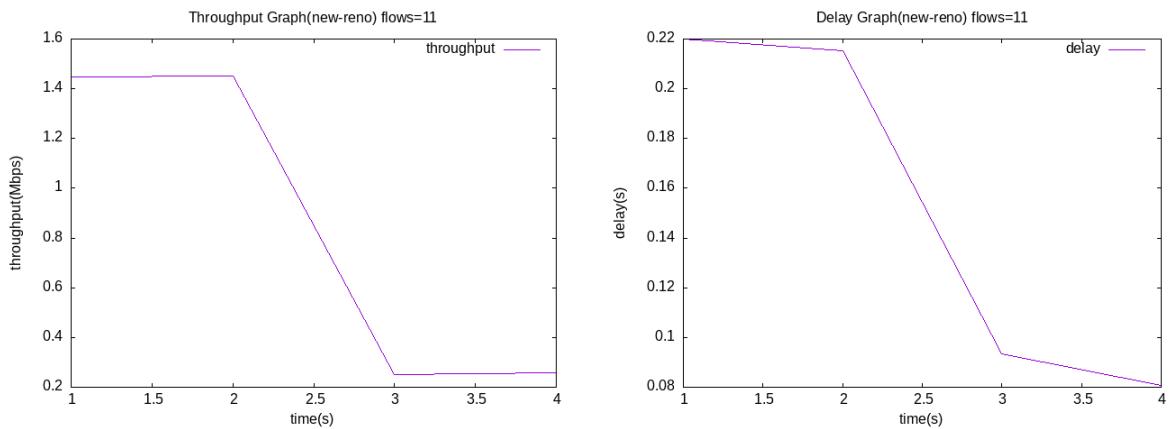
i)Number of flows=6

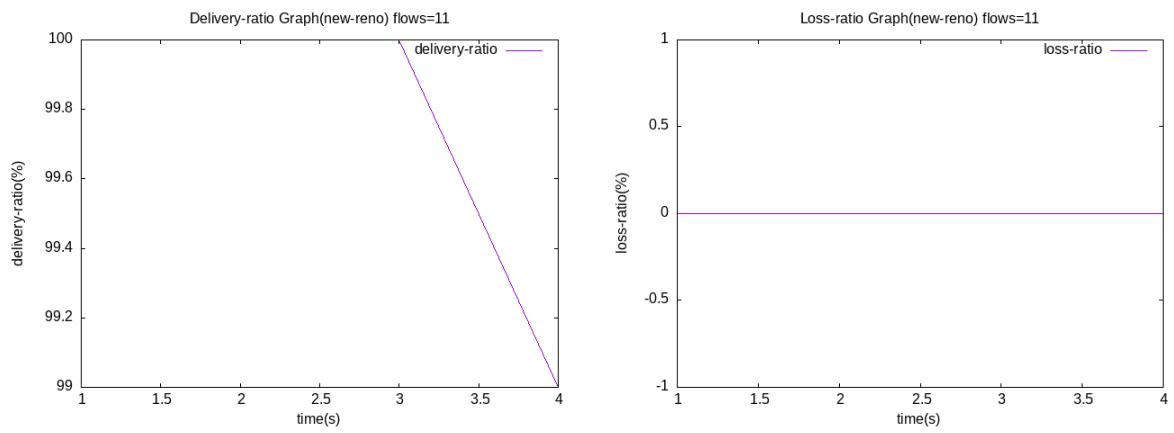


## ii) Number of flows=10

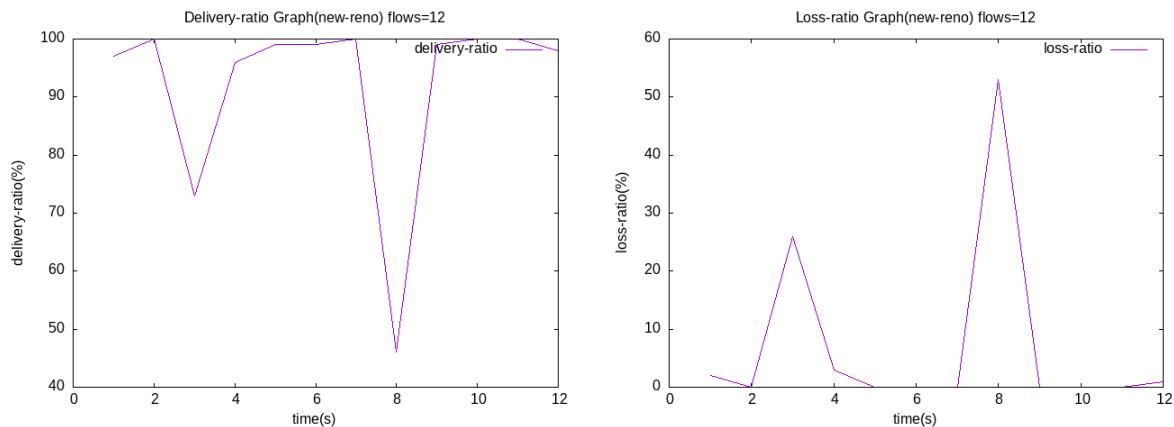
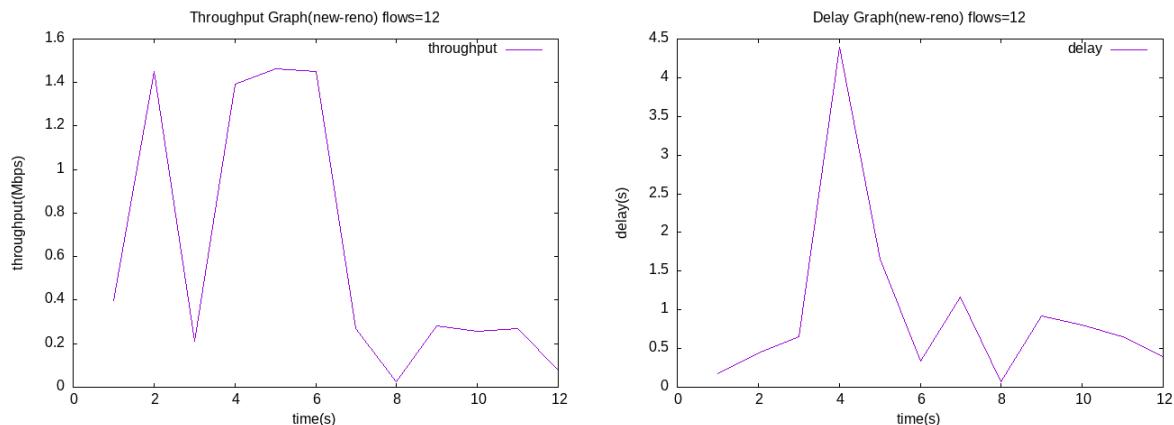


## iii) Number of flows=11

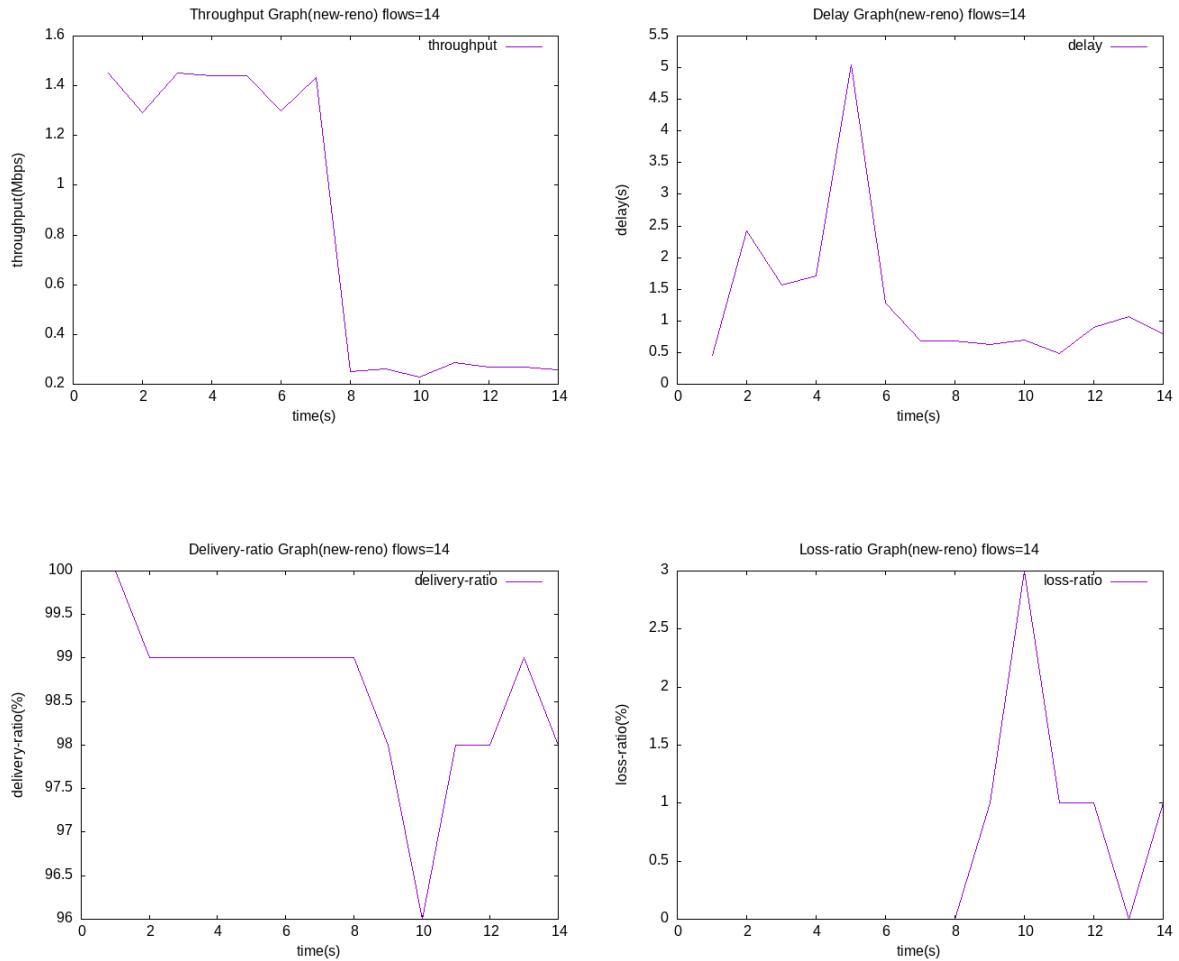




#### iv) Number of flows=12



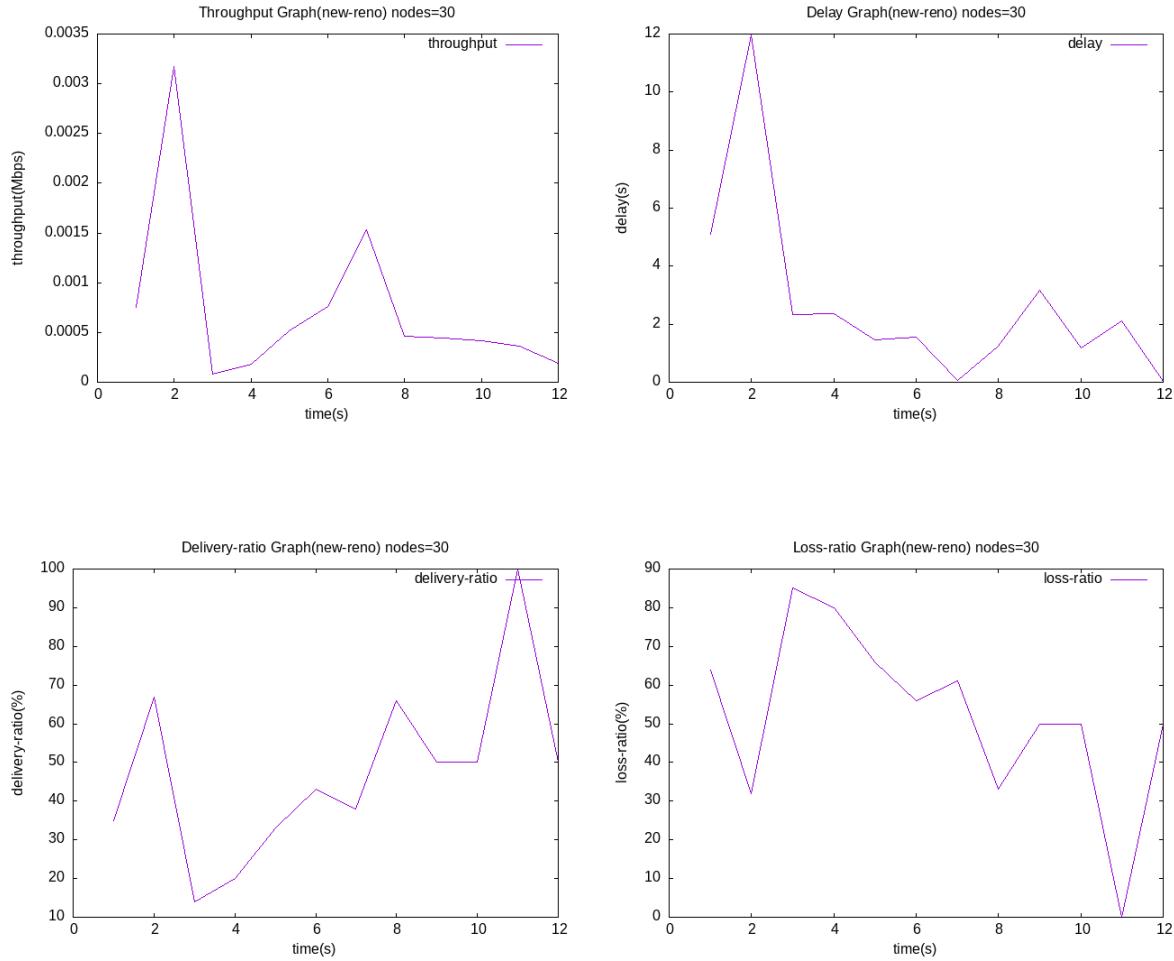
v) Number of flows=14



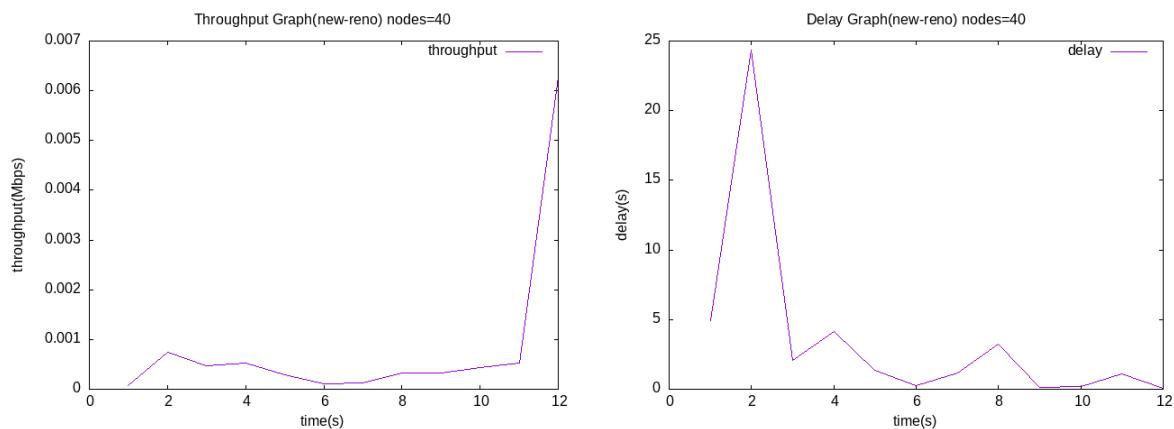
## Wireless low-rate (e.g., 802.15.4) (mobile) --- LR-WPAN

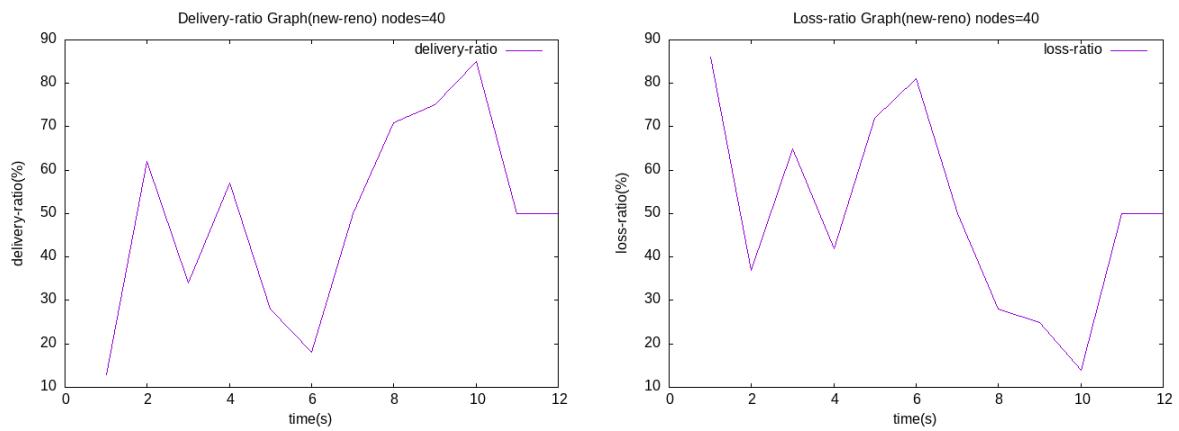
### 1)Varying number of nodes:

#### i)Number of nodes=30

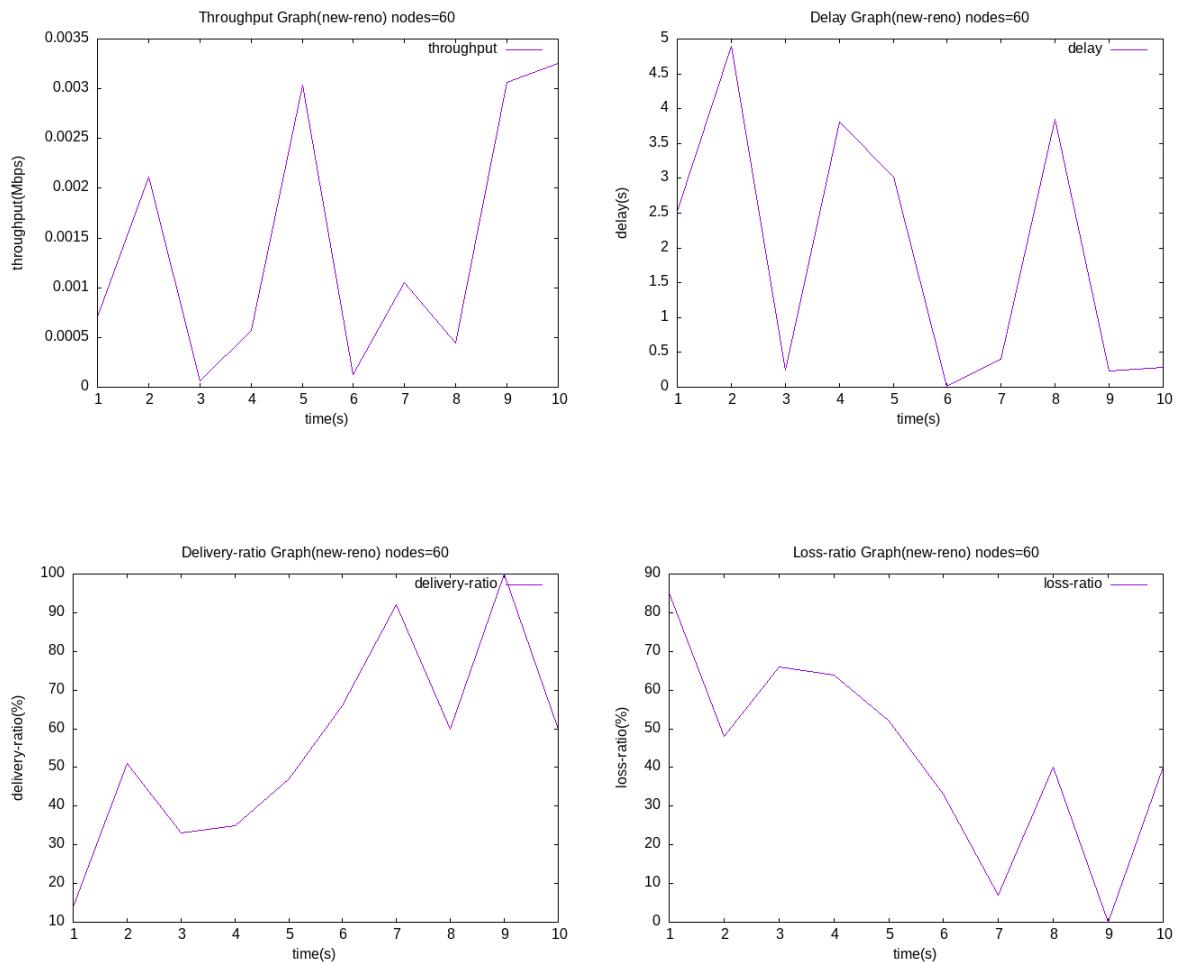


#### ii)Number of nodes=40

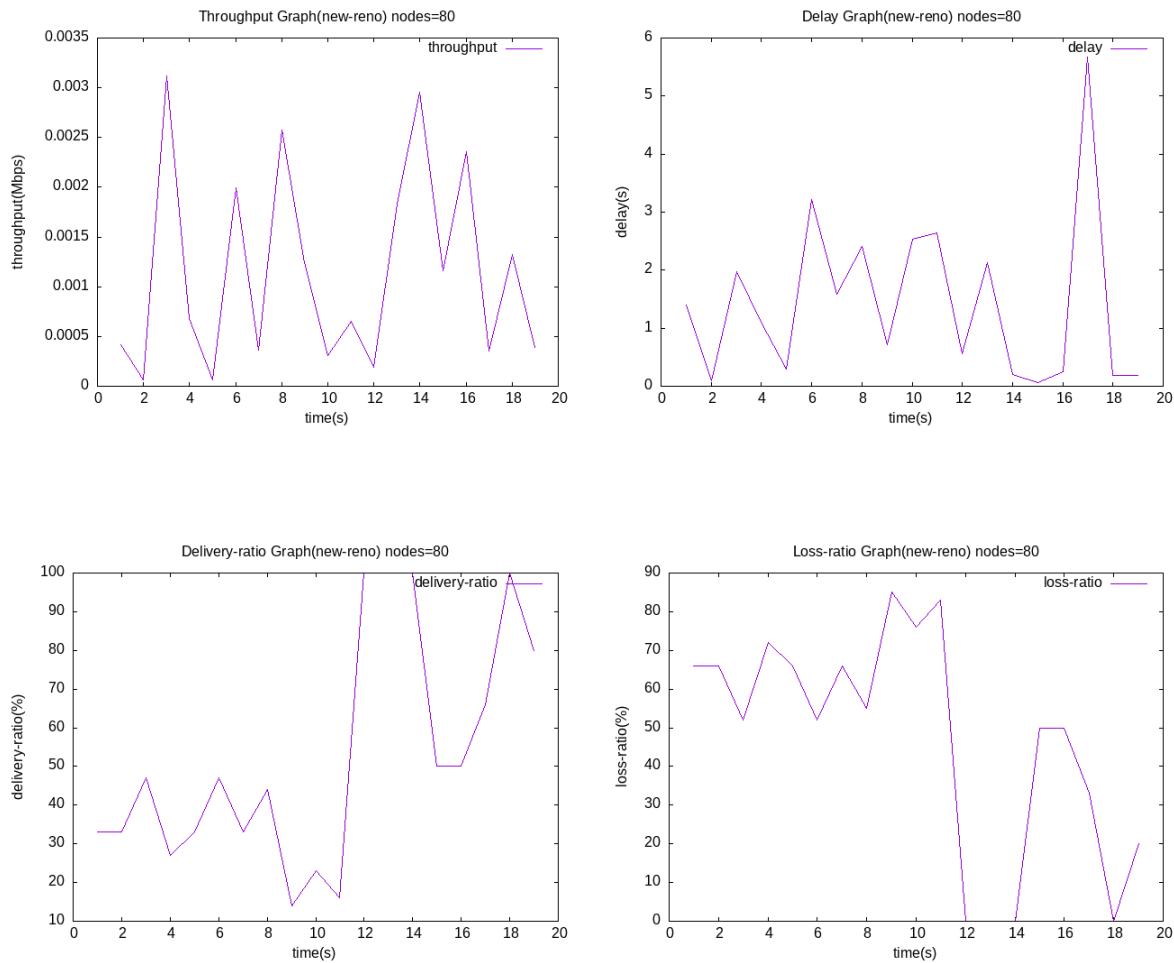




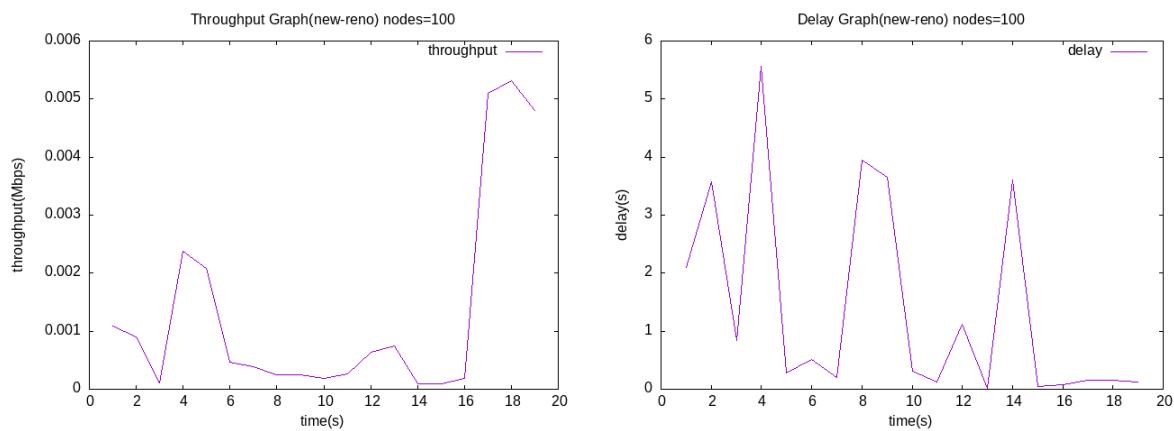
### iii) Number of nodes=60

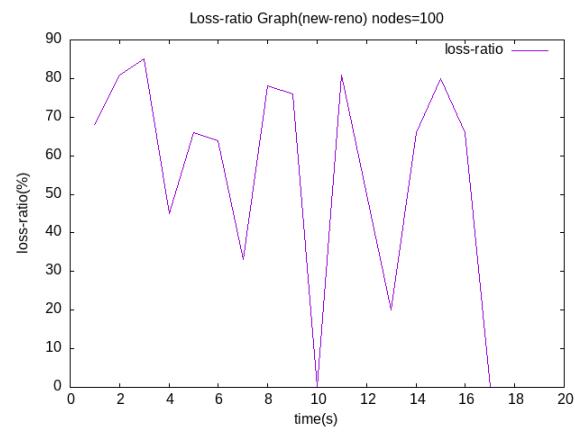
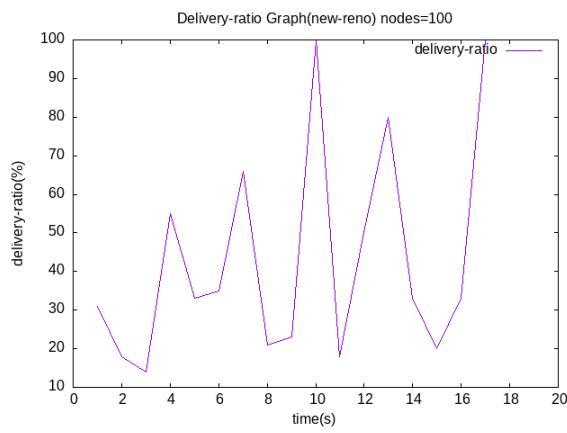


#### iv) Number of nodes=80



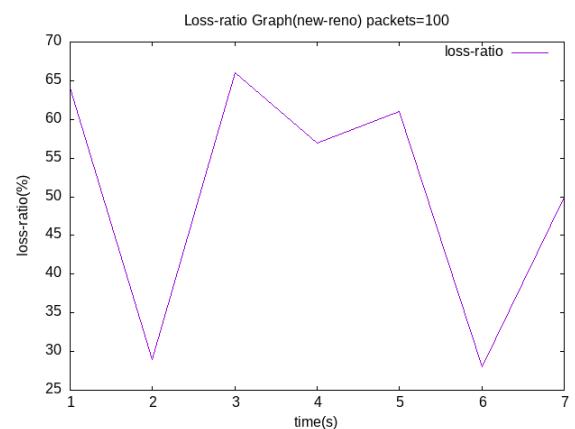
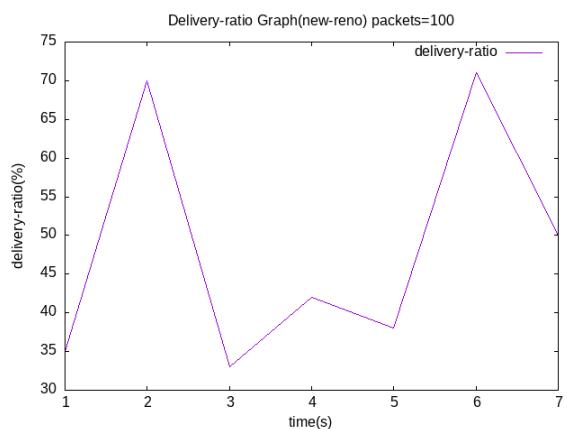
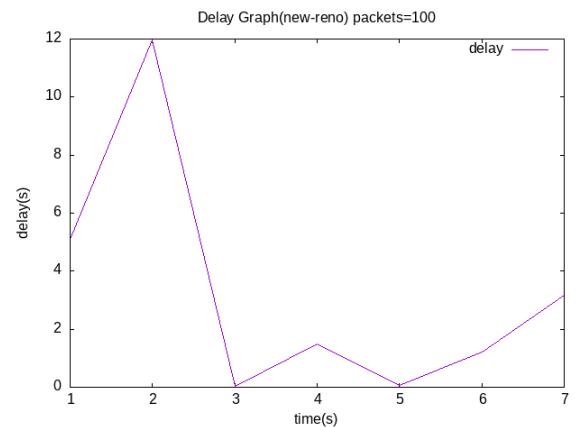
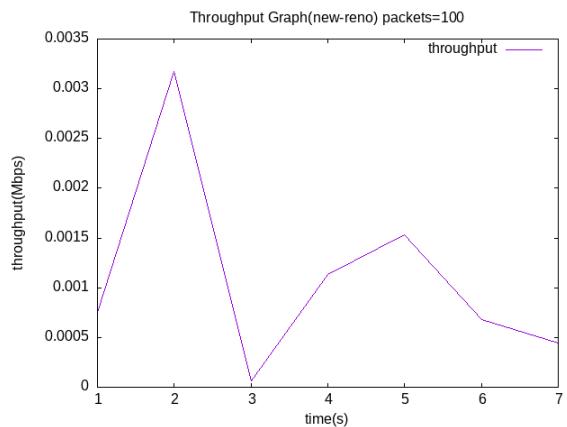
#### v) Number of nodes=100



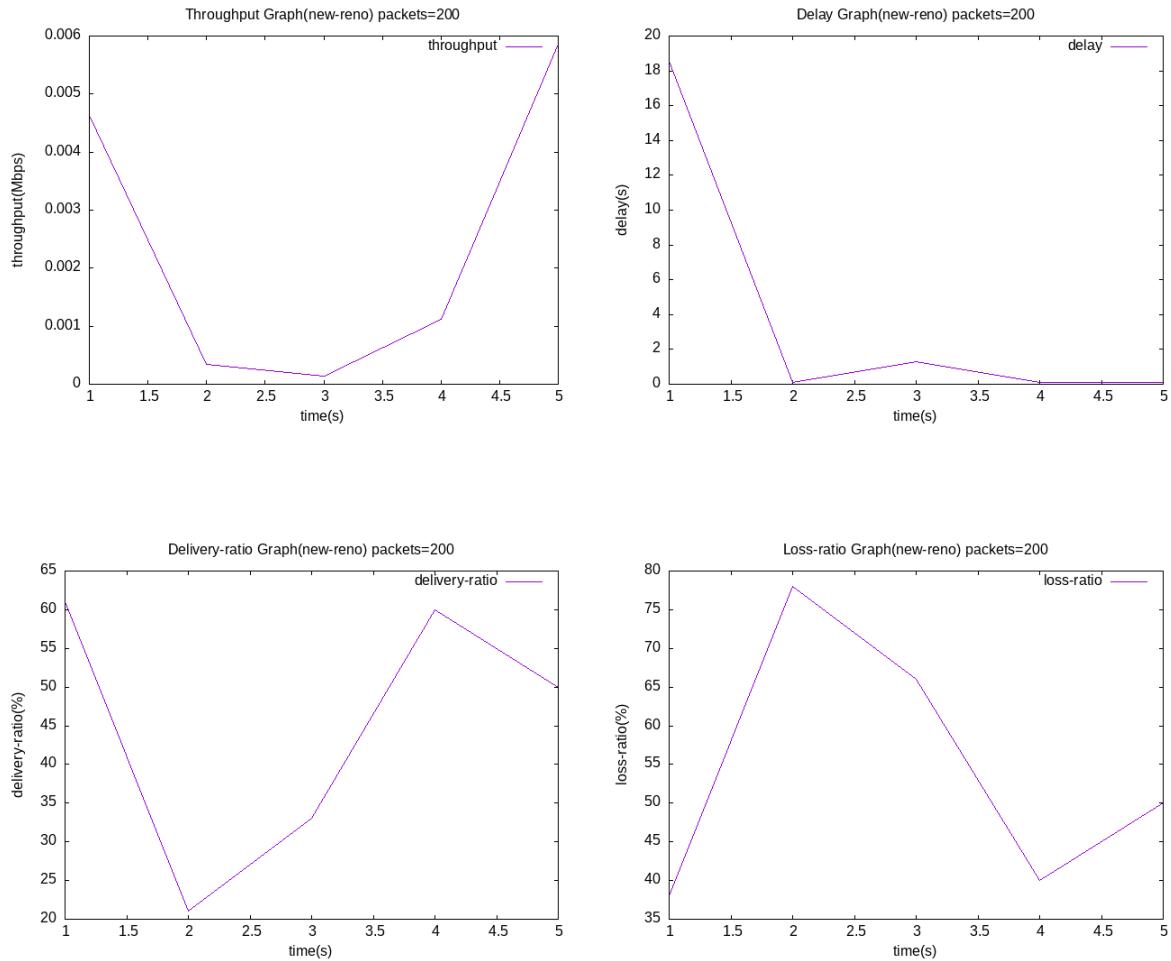


## 2)Varying number of packets per size:

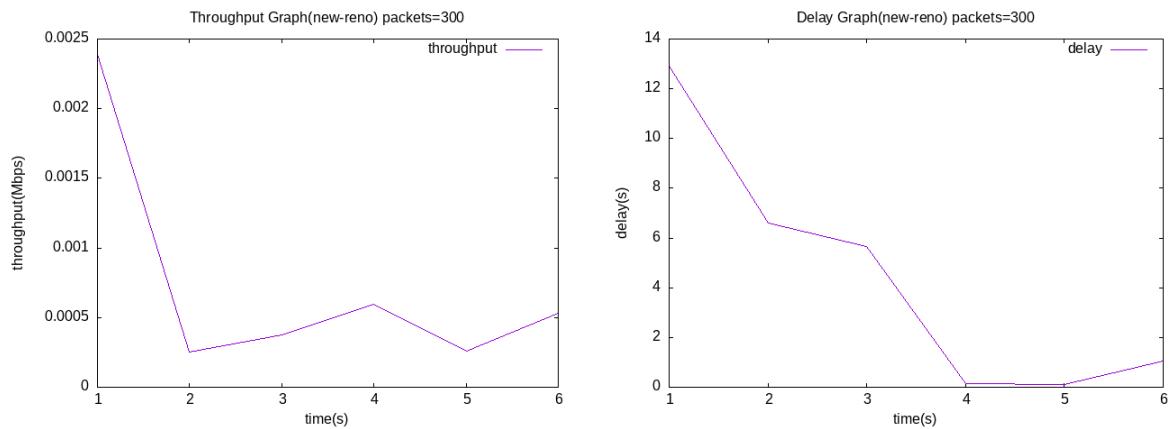
i)Number of packets=100

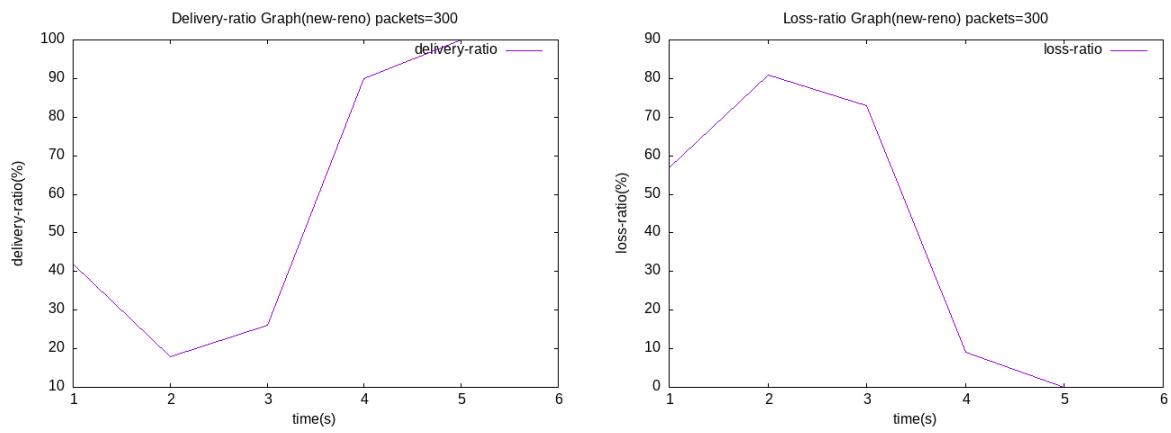


## ii) Number of packets=200

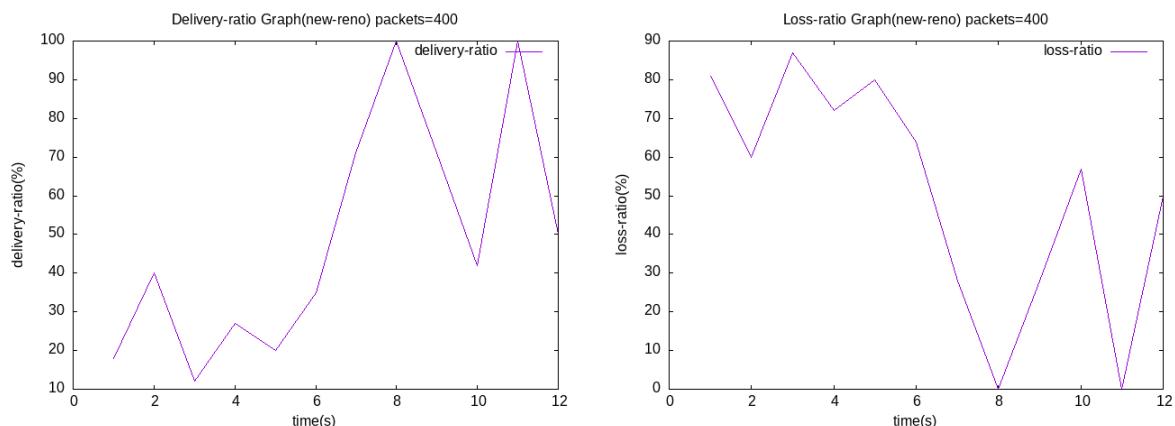
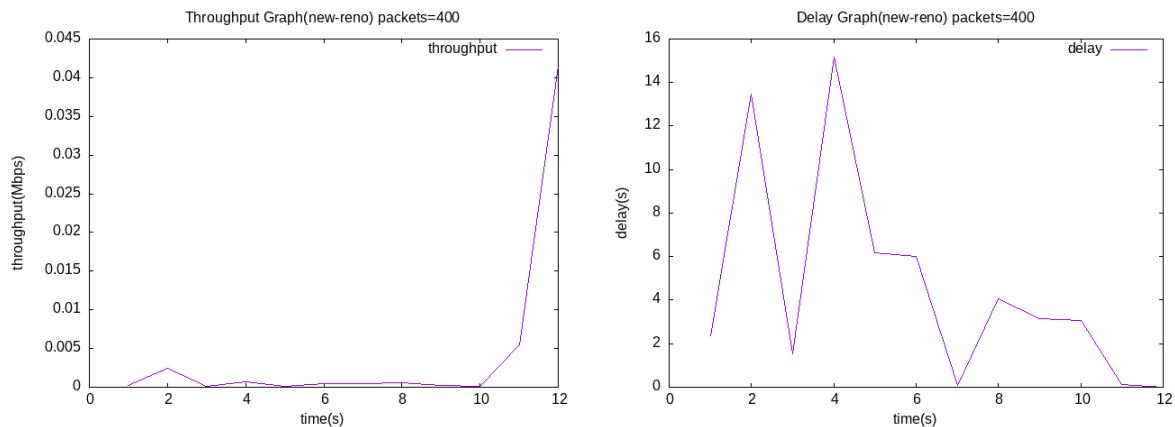


## iii) Number of packets=300

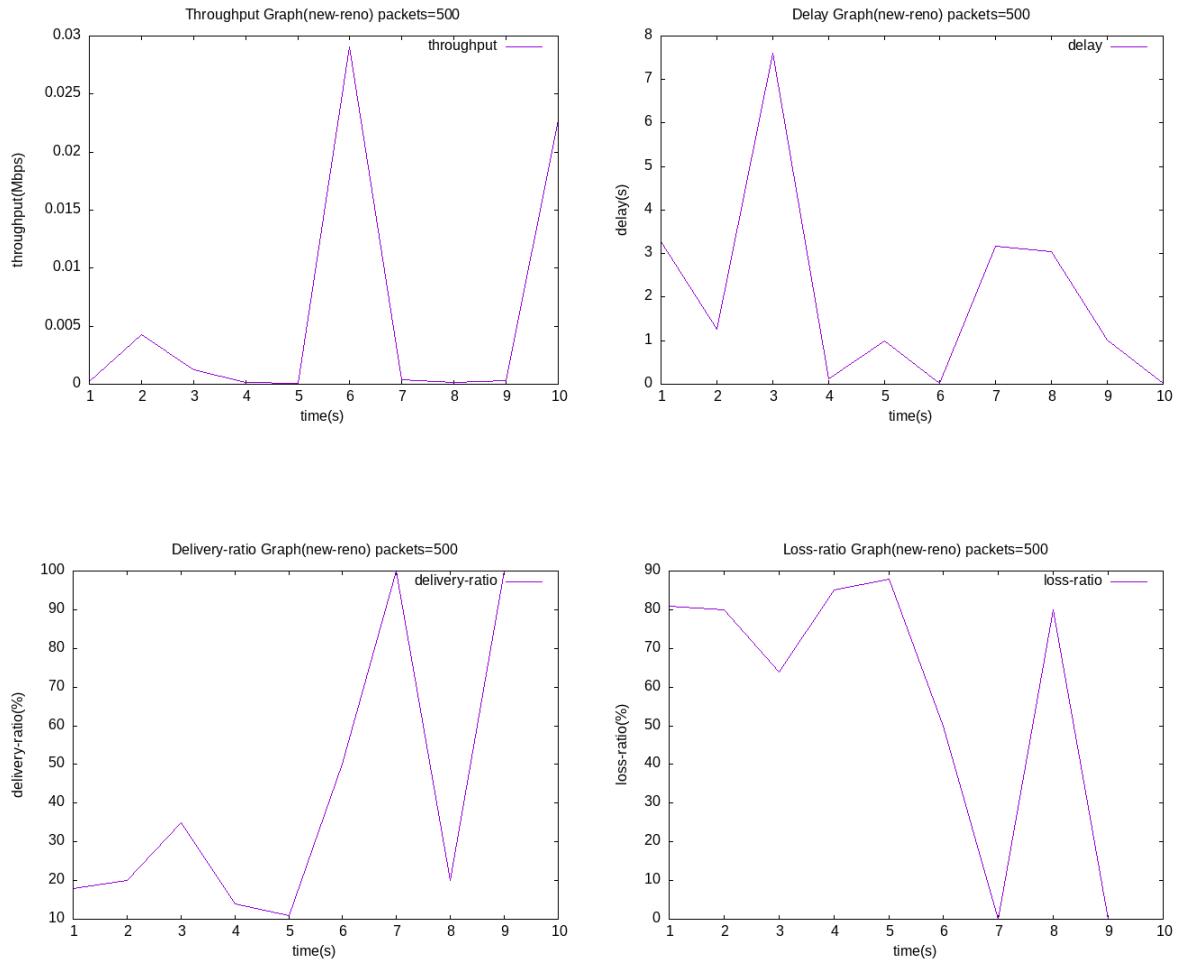




iv) Number of packets=400

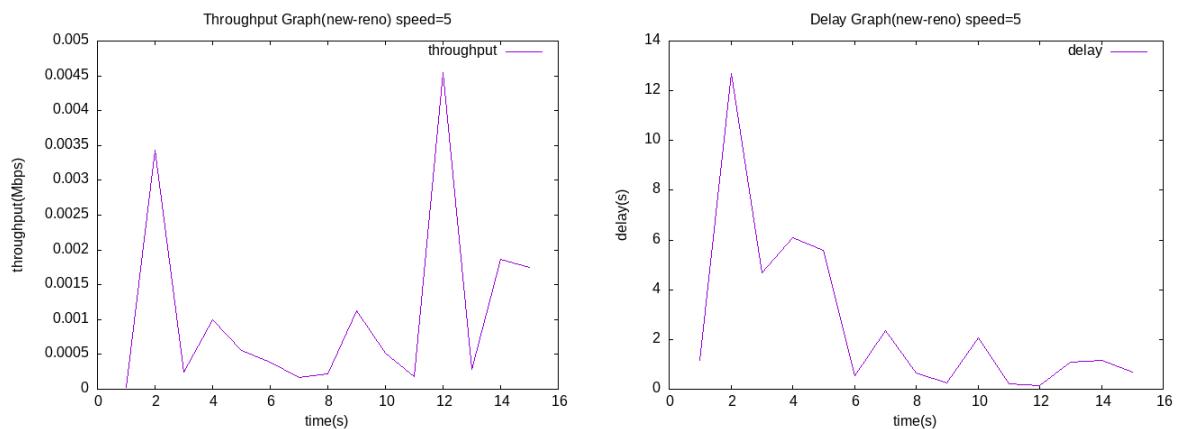


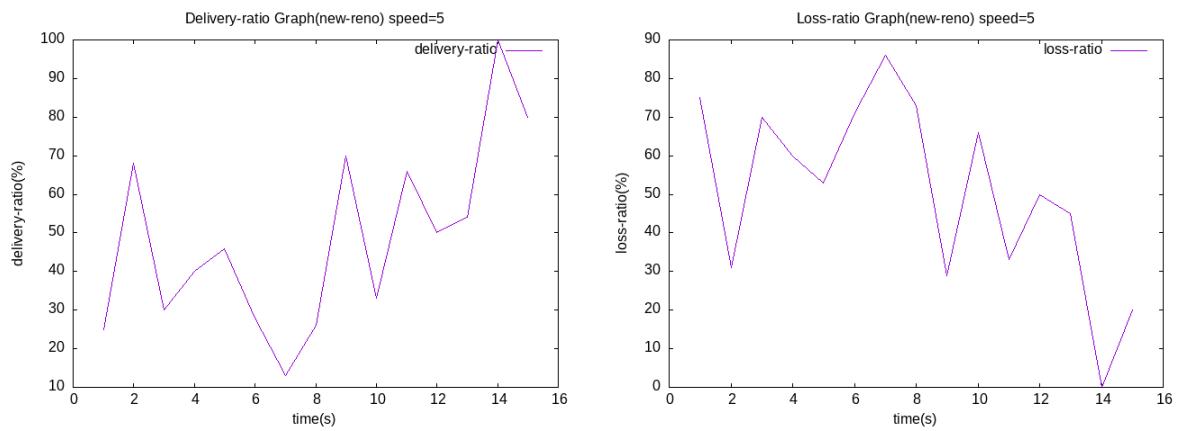
## v) Number of packets=500



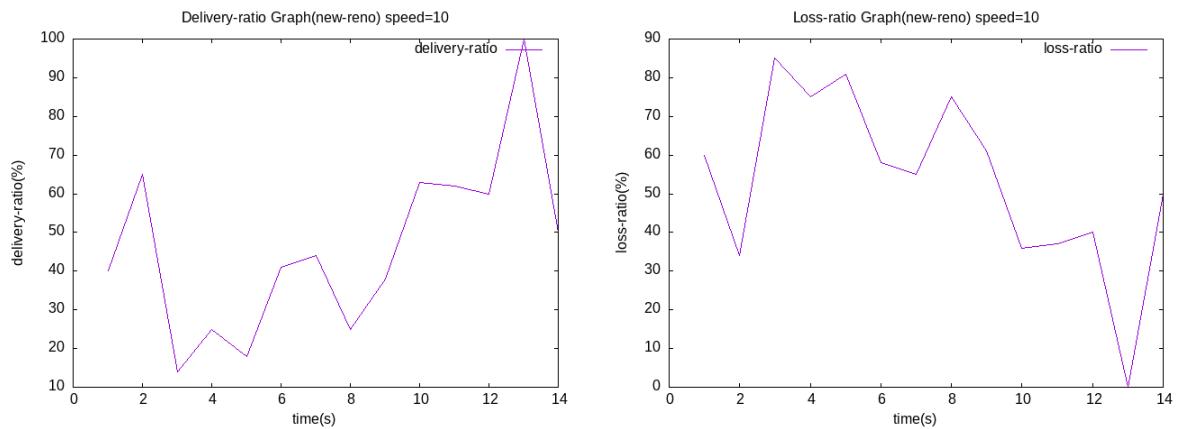
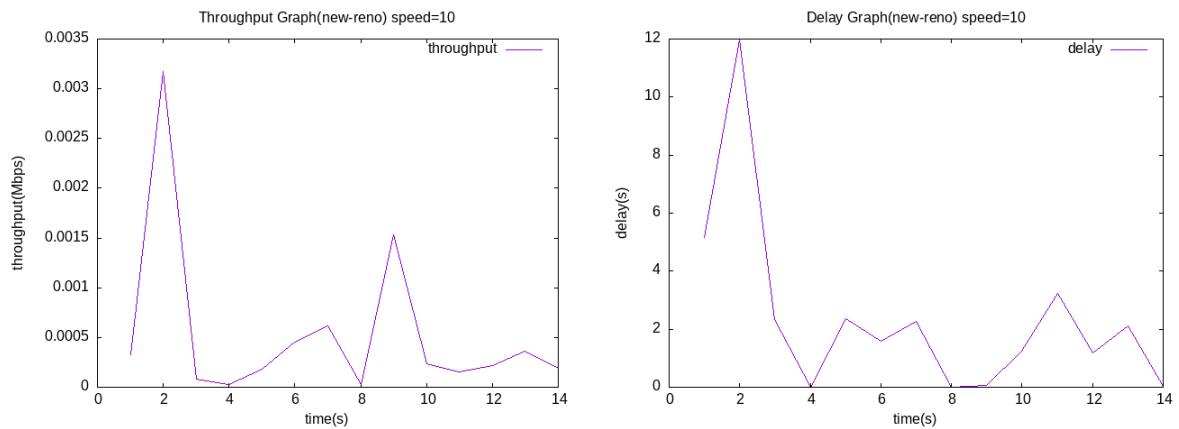
## 3) Varying speed of nodes:

### i) Speed of nodes=5 m/s

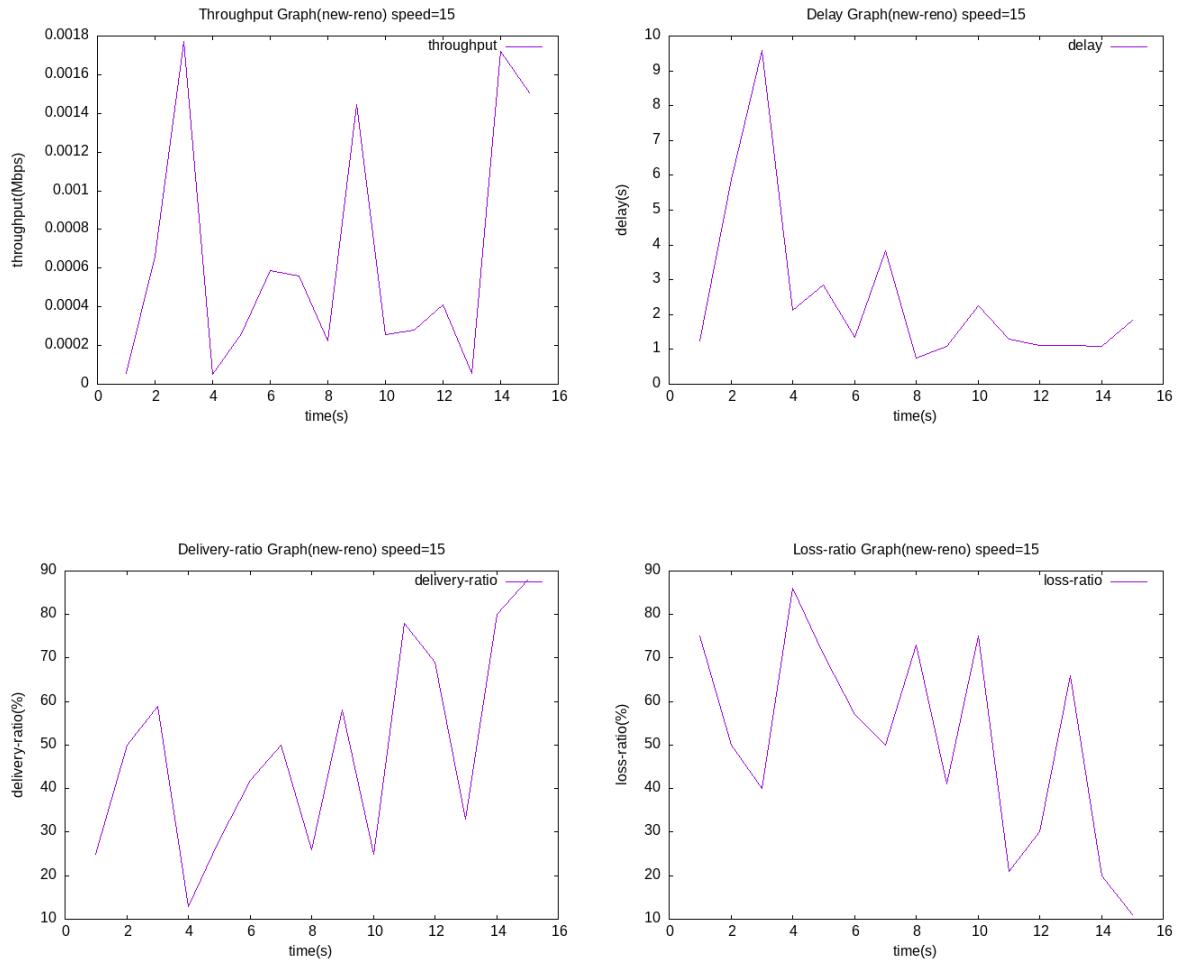




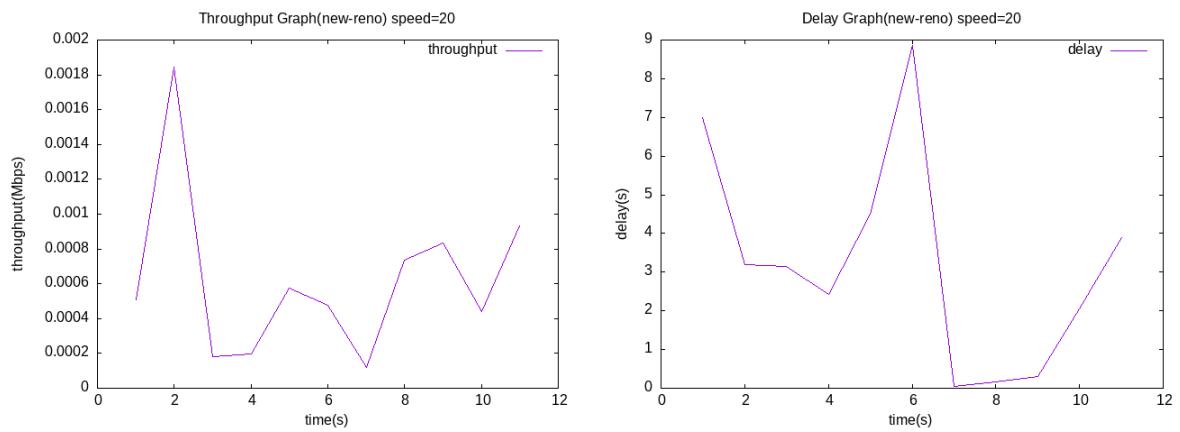
ii) Speed of nodes=10 m/s

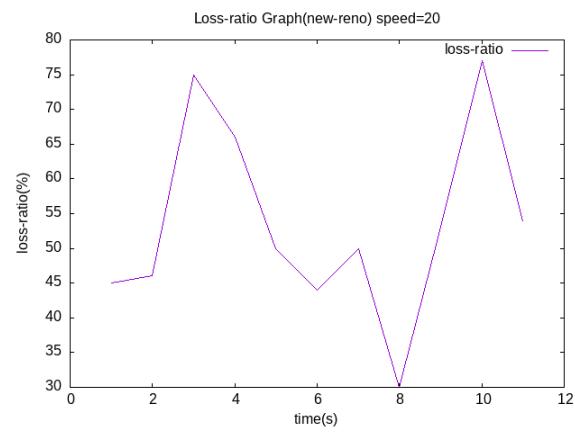
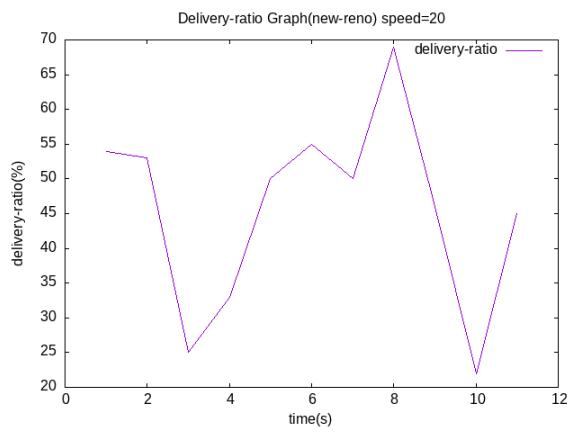


### iii) Speed of nodes=15 m/s

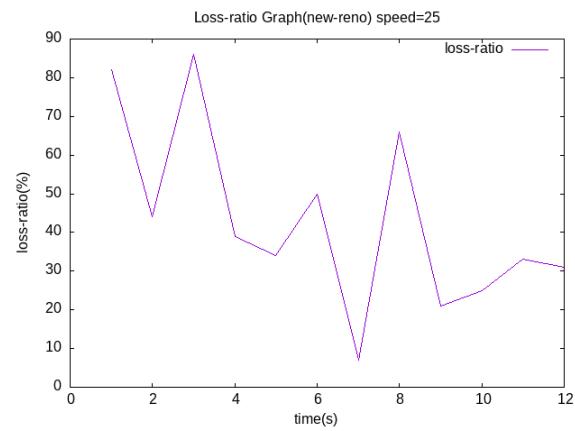
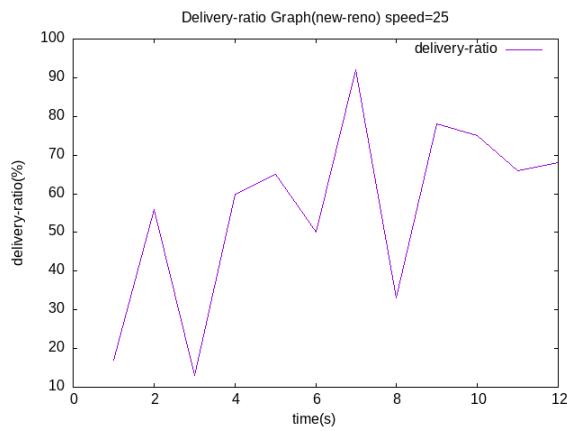
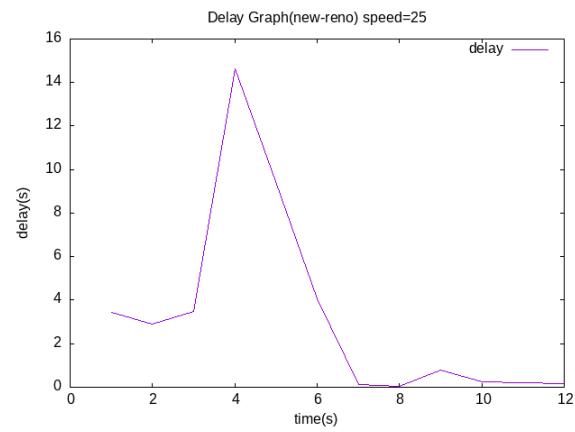
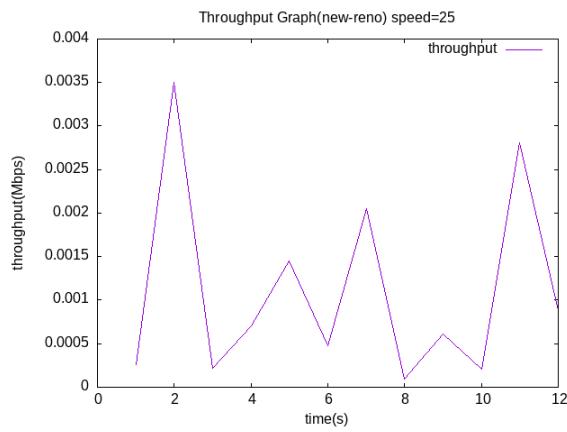


### vi) Speed of nodes=20 m/s



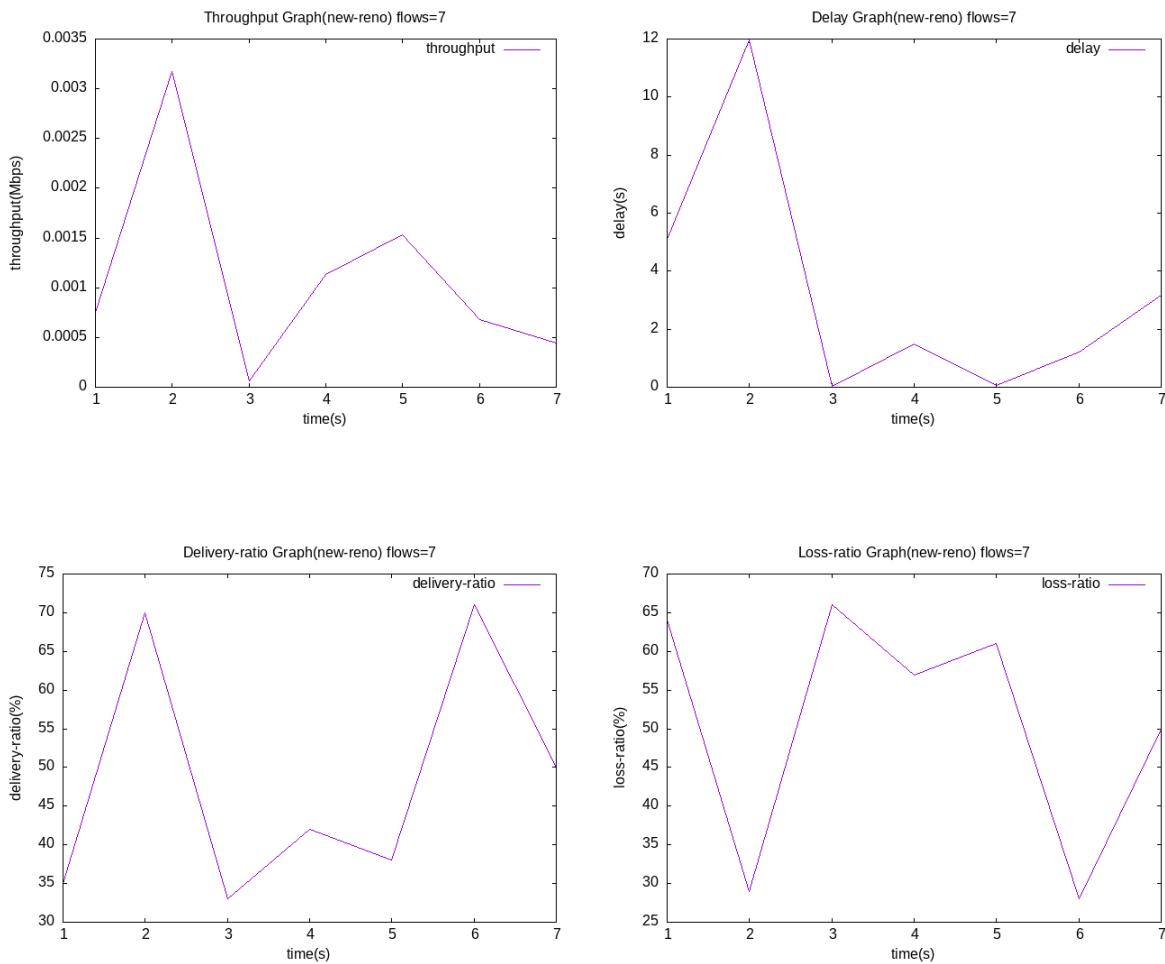


v) Speed of nodes=25 m/s

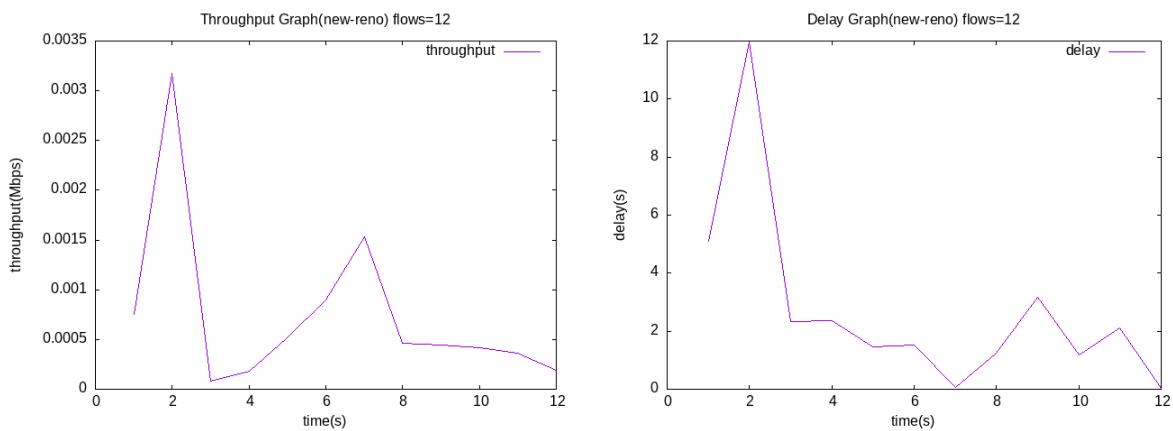


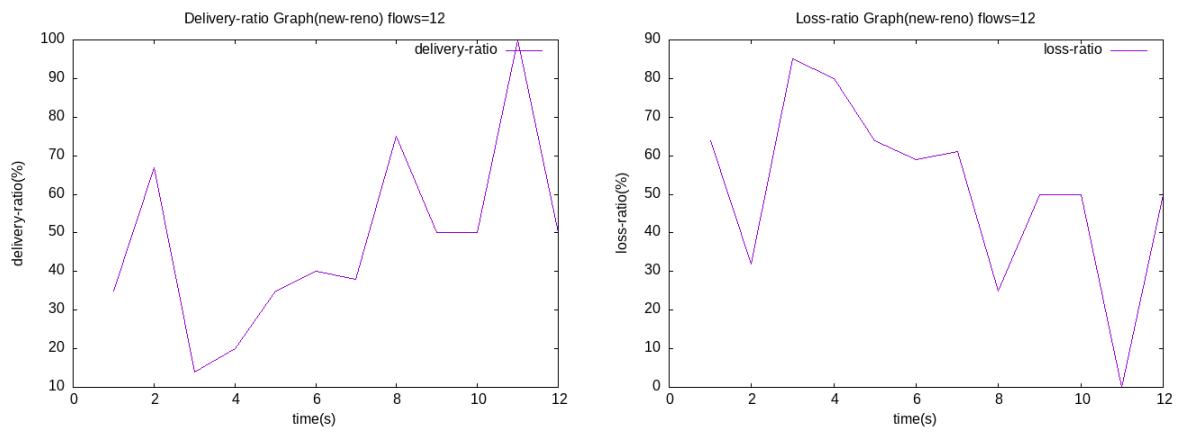
## 4)Varying number of flows:

### i)Number of flows=7

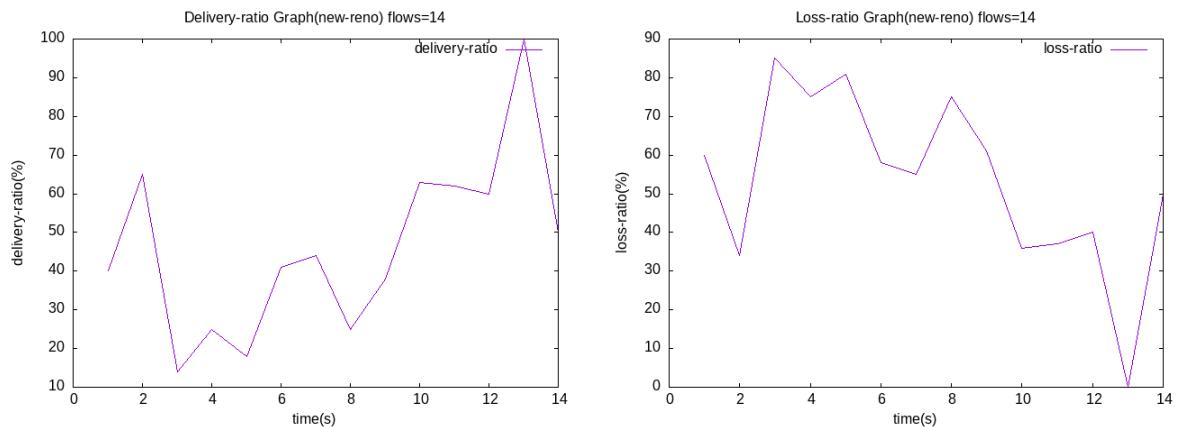
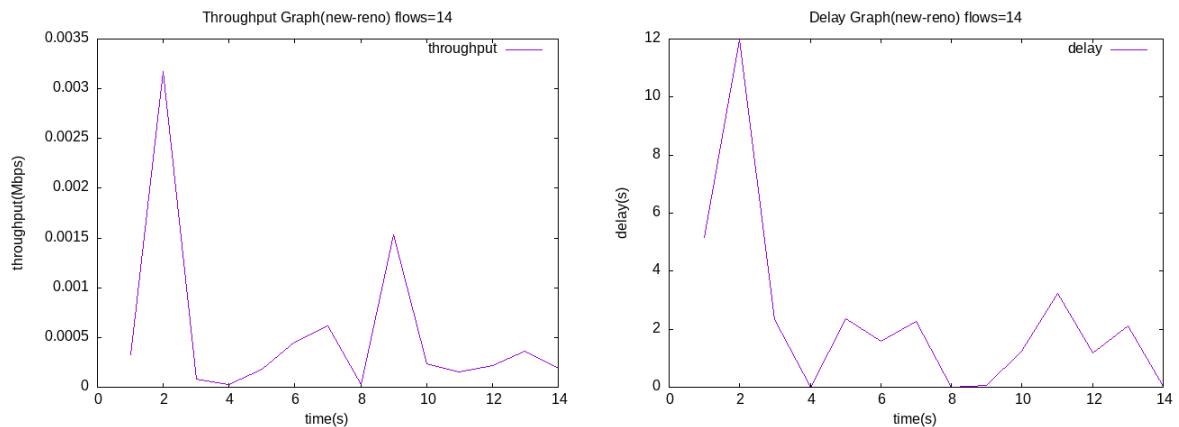


### ii)Number of flows=12

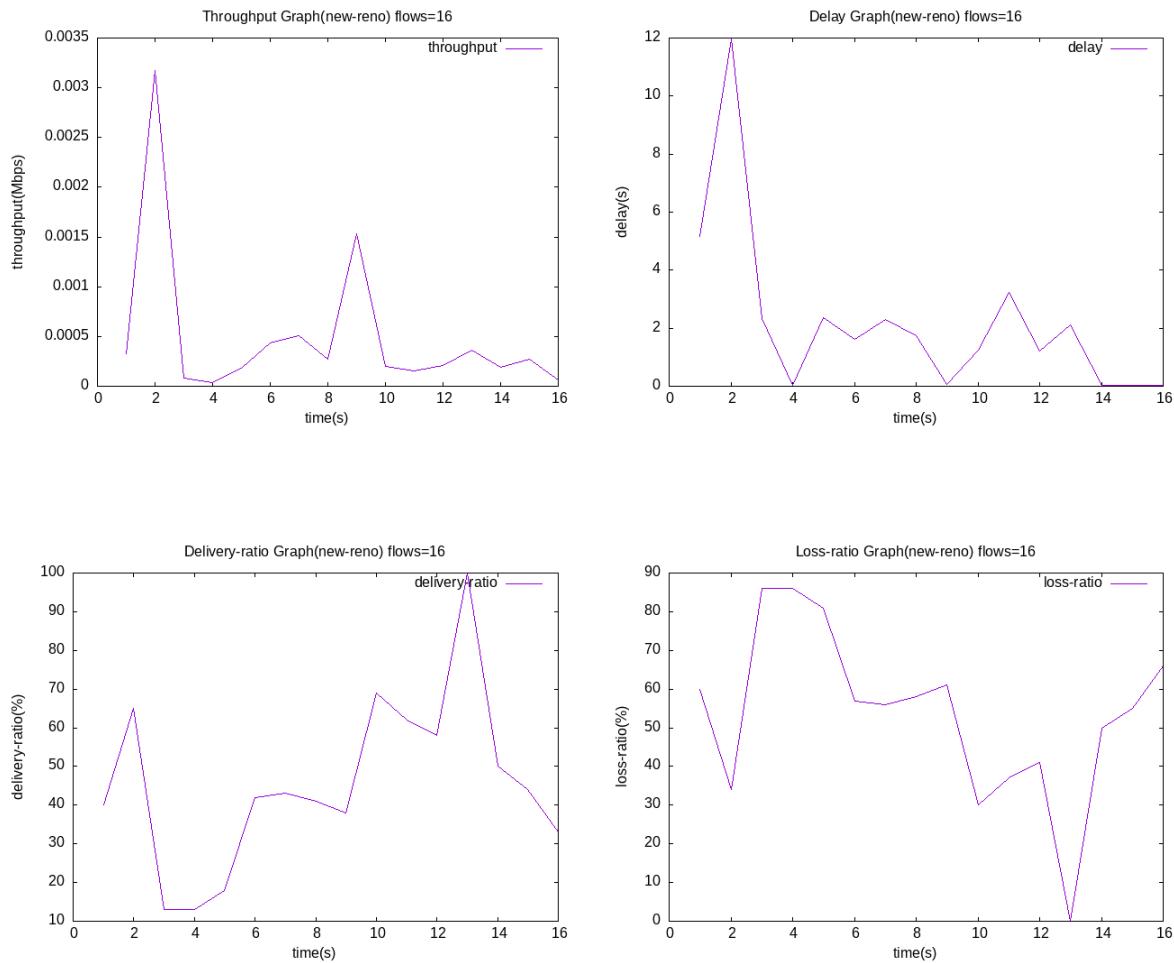




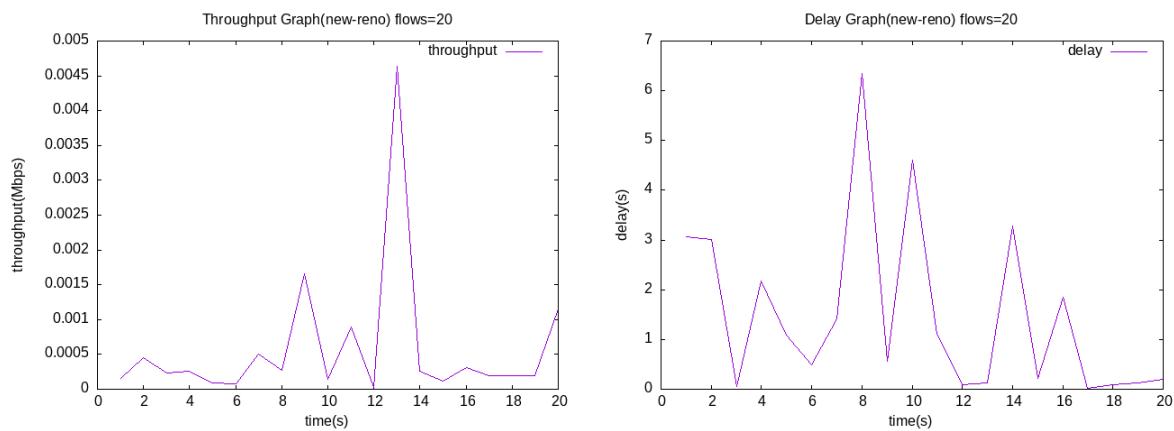
### iii) Number of flows=14

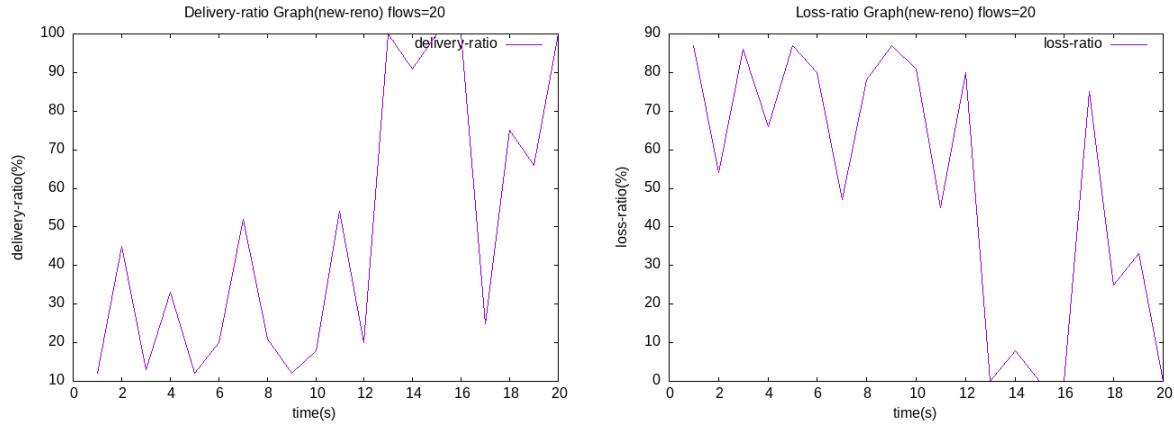


#### iv) Number of flows=16



#### v) Number of flows=20





## Section G: Explanation of results of Task A

**(Task A has been conducted with TCP New-Reno)**

For the Wi-Fi network, we can see that when number of nodes increases, throughput decreases, end-to-end delay time increases, delivery-ratio increases and loss-ratio remains almost same. When increasing packets per second, throughput slightly differs, end-to-end delay time & delivery-ratio remains almost same and loss-ratio decreases. For the case of increment of nodes speed, throughput slightly changes, end-to-end delay time increases, the delivery-ratio increases and the loss-ratio decreases. At last, if we increase number of flows, we will observe slightly decrement of throughput, inconsistent delay time, decrement of delivery-ratio and increment of loss-ratio.

For the LR-WPAN network, it's clear that when number of nodes increases, throughput , end-to-end delay time & delivery-ratio increases and loss-ratio slightly decreases. When increasing packets per second, throughput decreases, end-to-end delay increases, delivery-ratio decreases and loss-ratio first decreases (for <400), then increases(for 400,500). For the case of increment of nodes

speed, throughput slightly changes, end-to-end delay time remains almost same, delivery-ratio increases and loss-ratio slightly decreases. At last, if we increase number of flows, we will observe decrement of throughput, slightly increment of delay time, increment of delivery-ratio and decrement of loss-ratio.

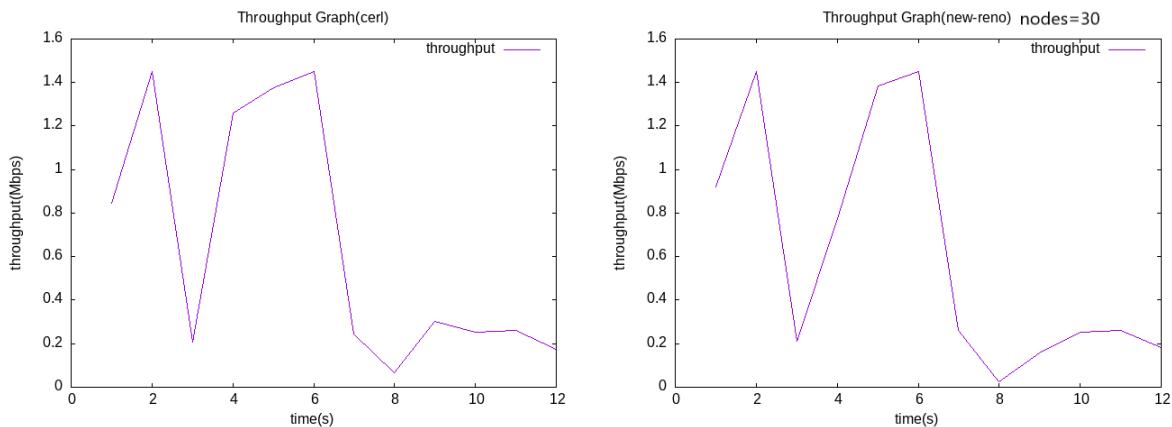
## Section H: Results & Comparison with graphs (Task B)

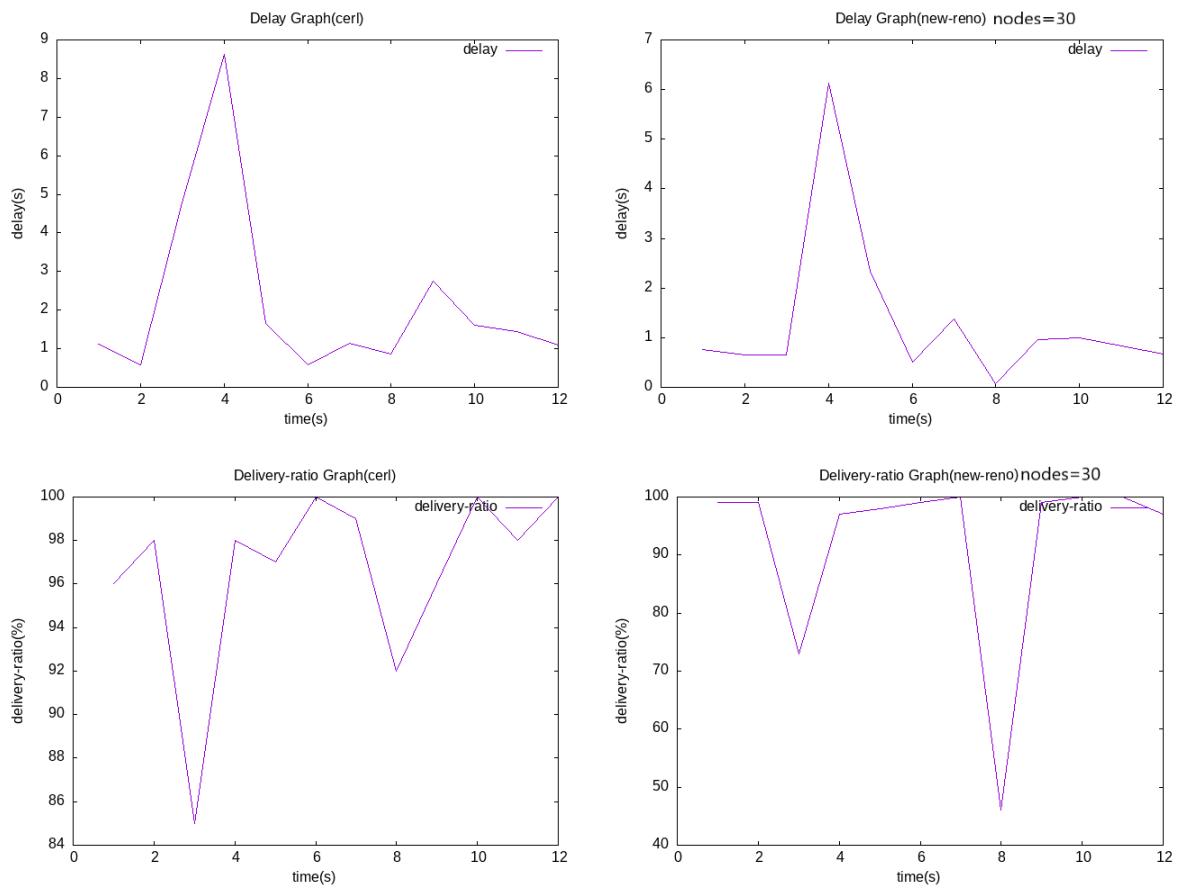
To compare TCP CERL with TCP NewReno, **three** performance metrics (throughput, end-to-end-delay & packet-delivery ratio) are measured varying **two** parameters (# of nodes & speed of nodes). The comparison is showed below using graphs.

### For Wireless high-rate (e.g., 802.11) (mobile) --- Wi-Fi

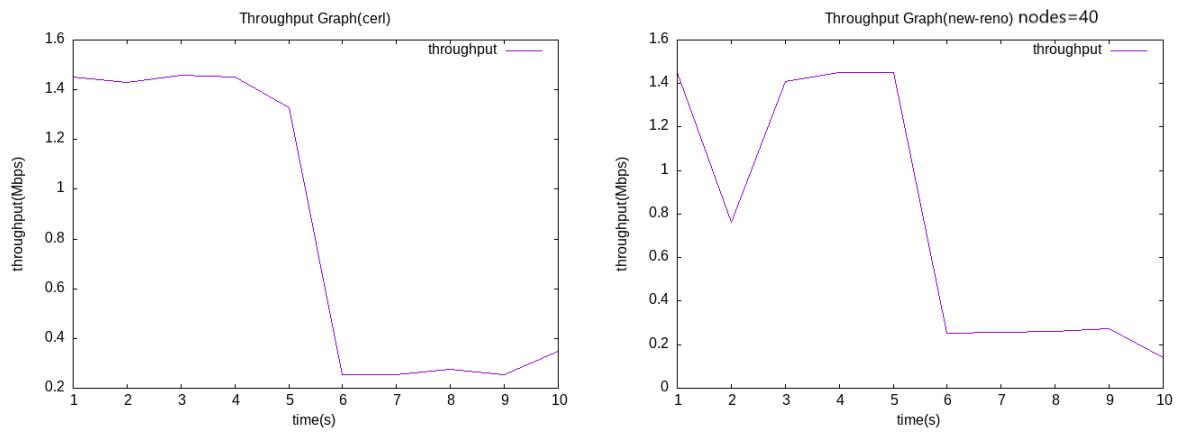
#### 1)Varying number of nodes:

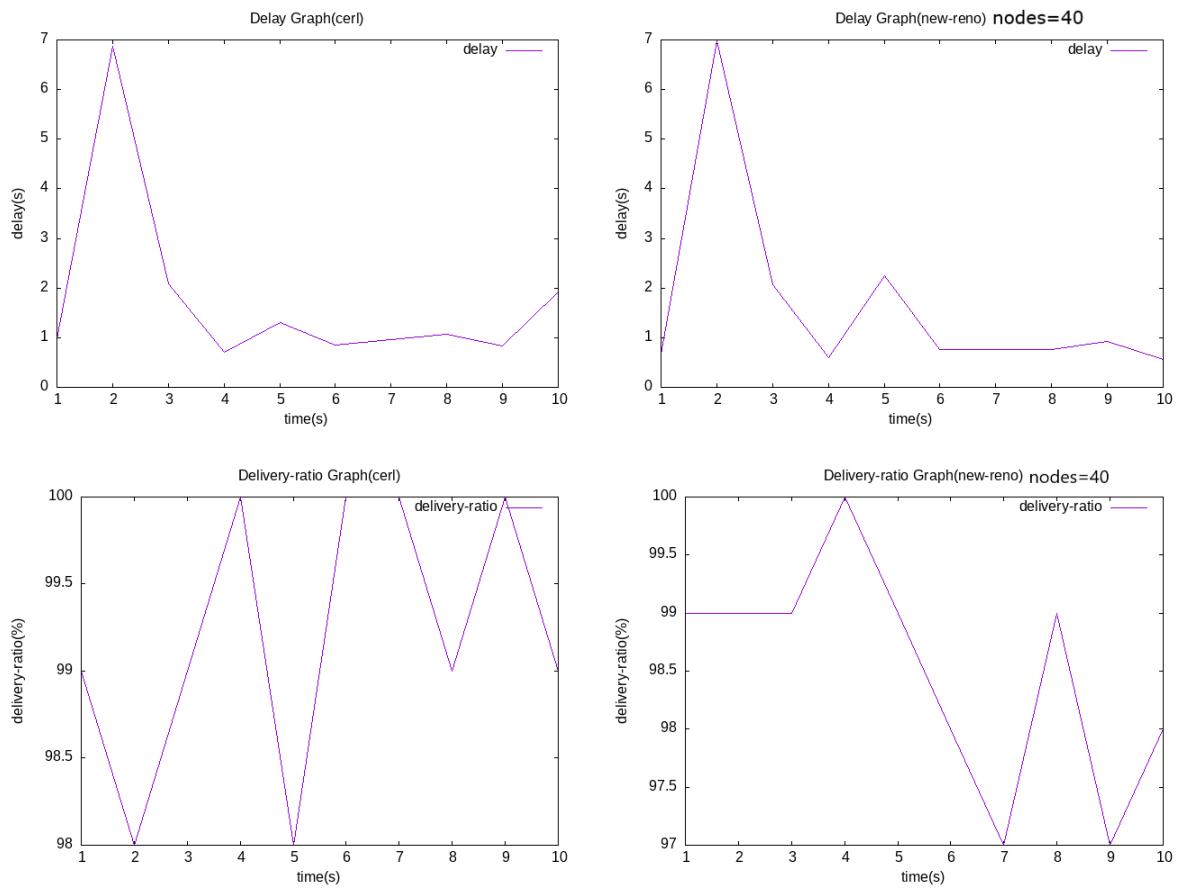
i)Number of nodes=30



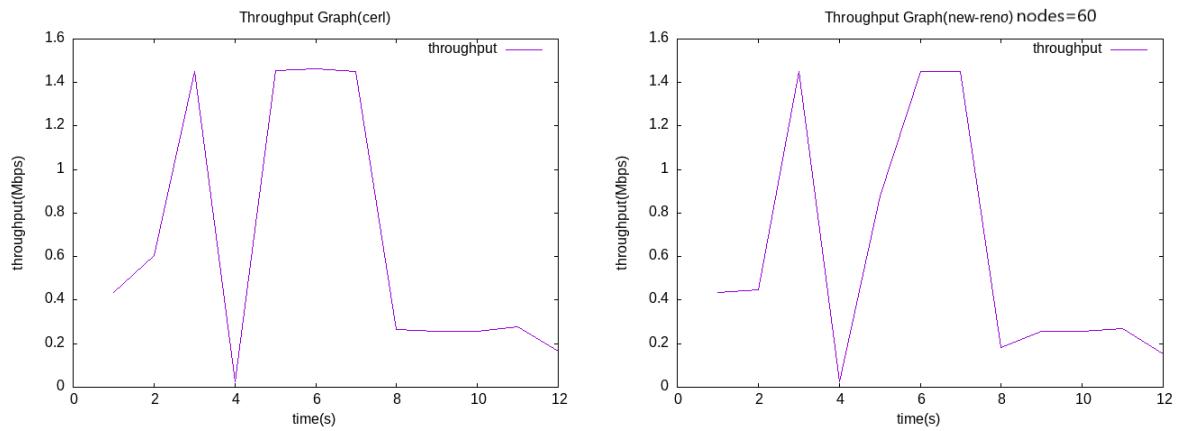


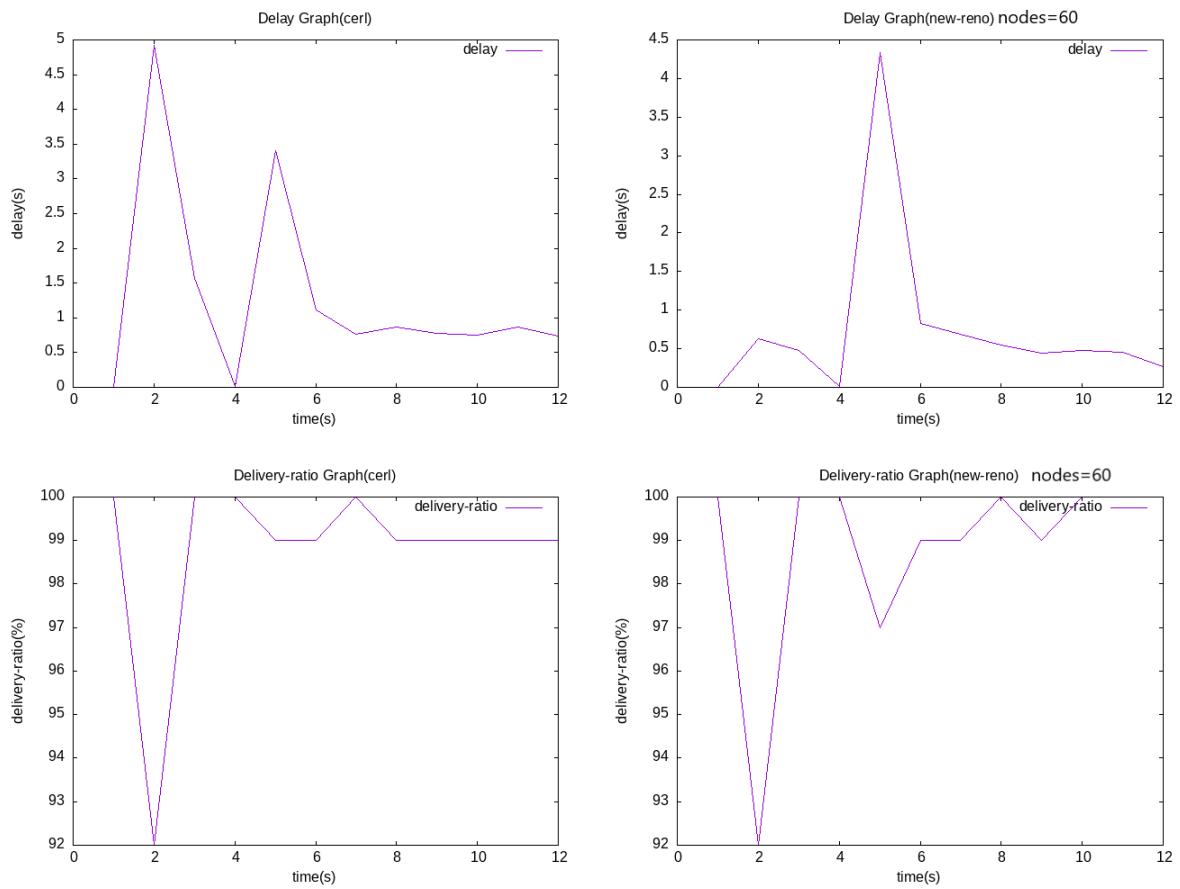
## ii) Number of nodes=40



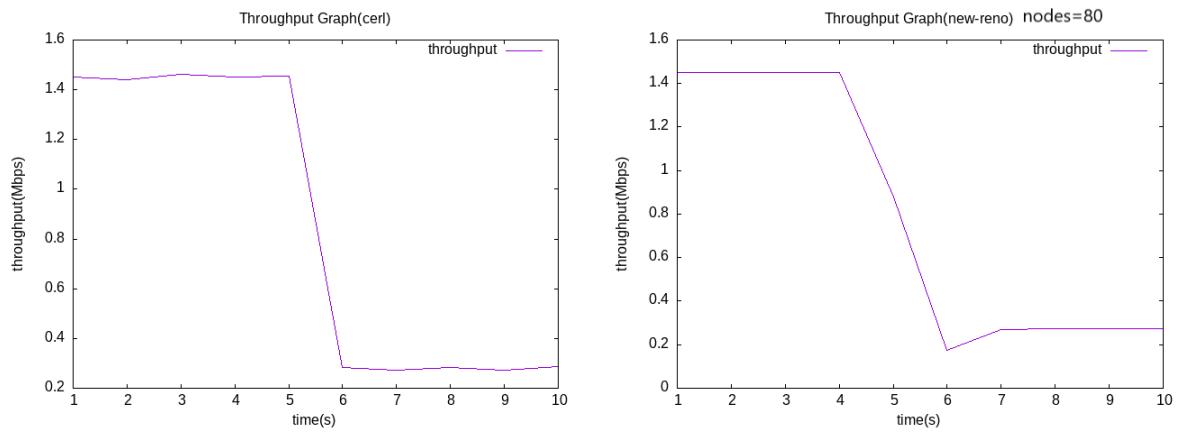


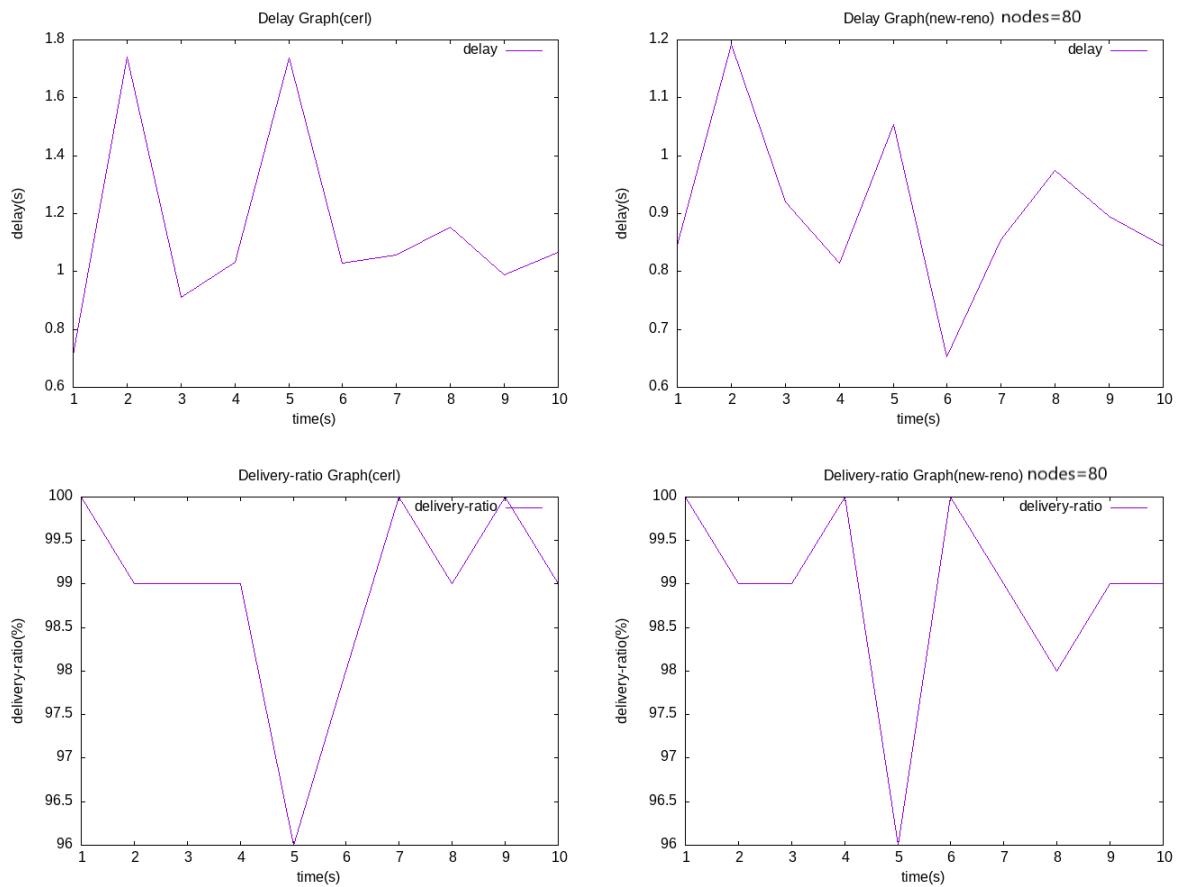
### iii) Number of nodes=60



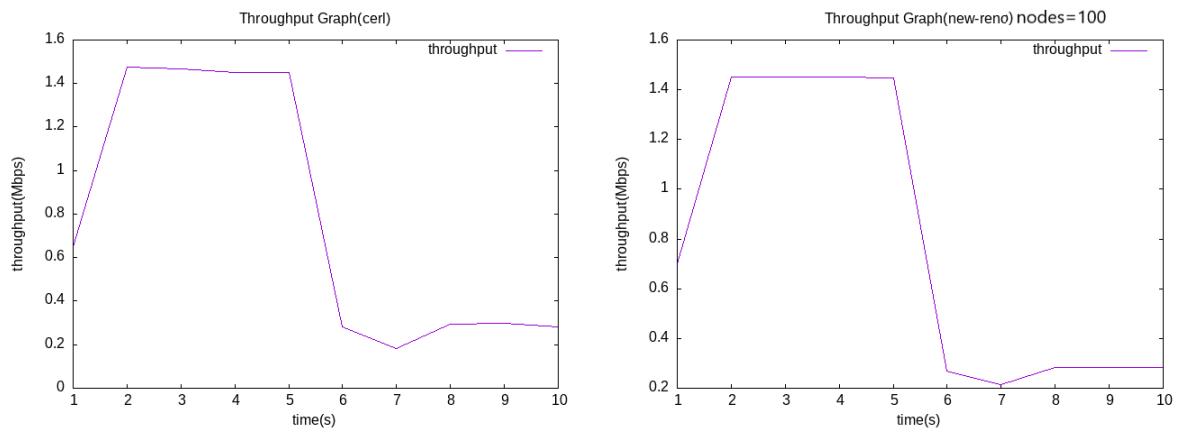


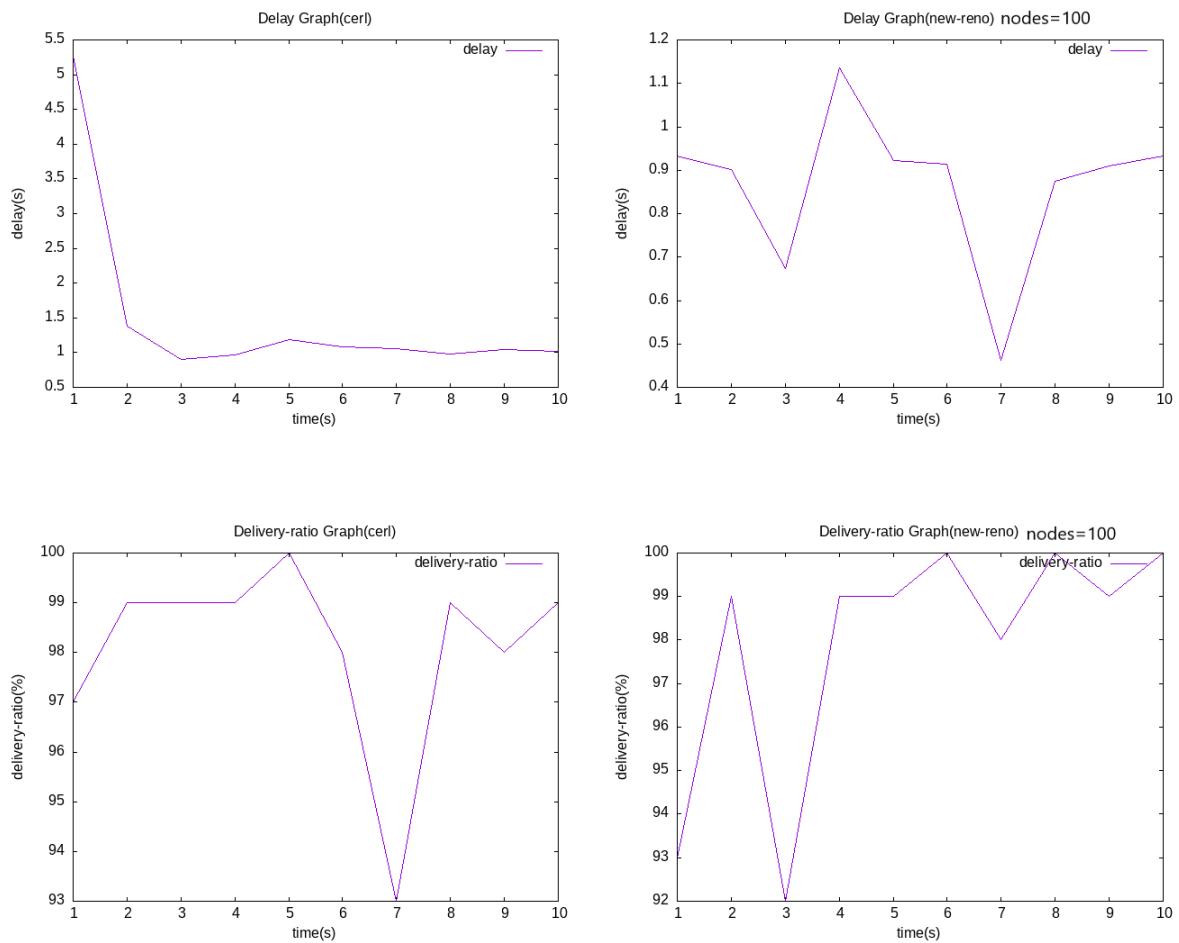
#### iv) Number of nodes=80





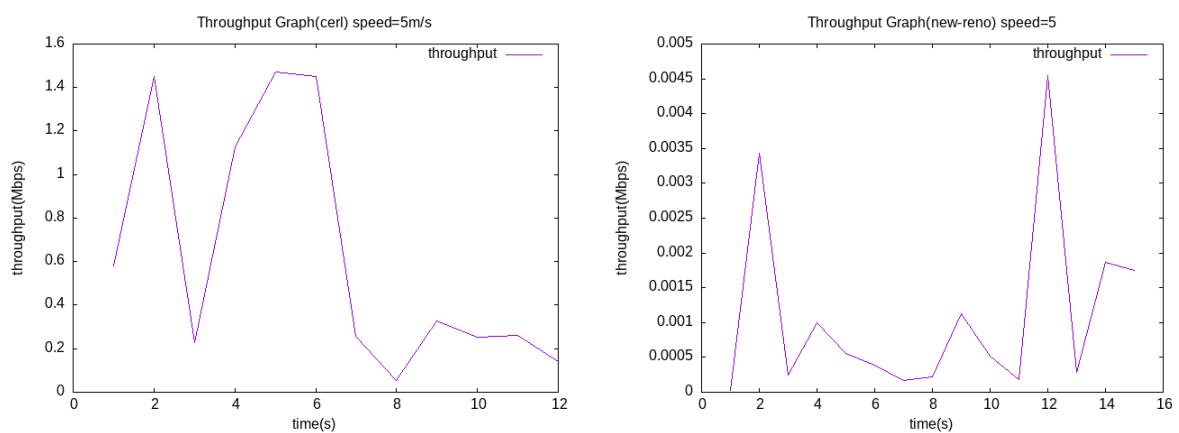
v) Number of nodes=100

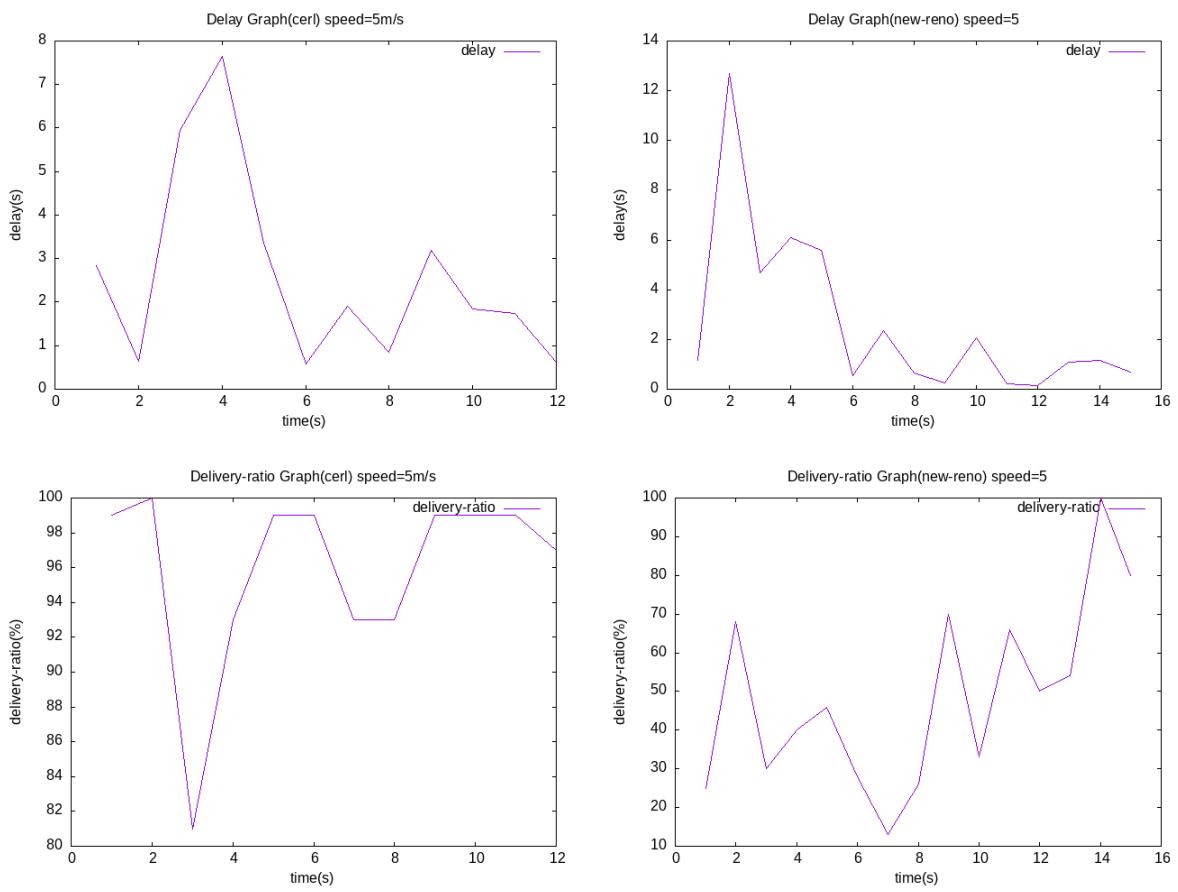




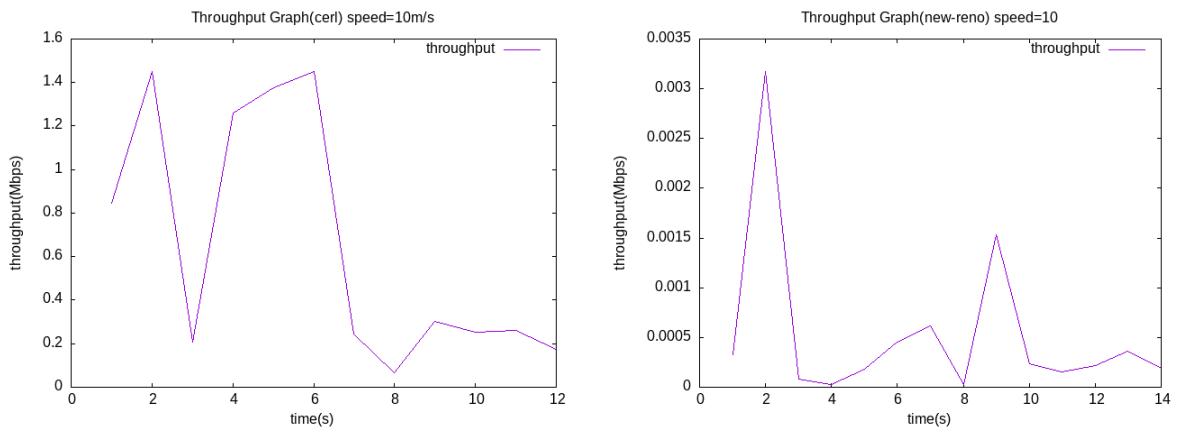
## 2)Varying speed of nodes:

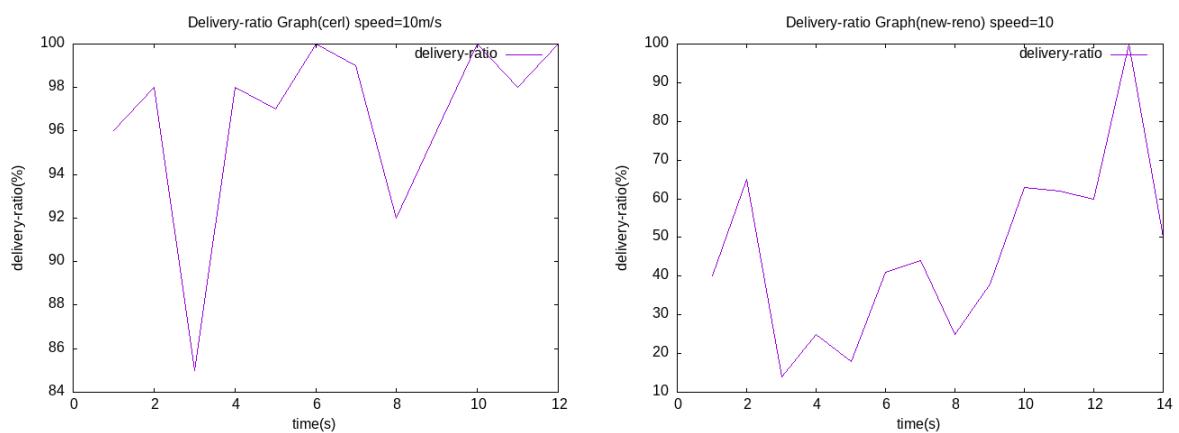
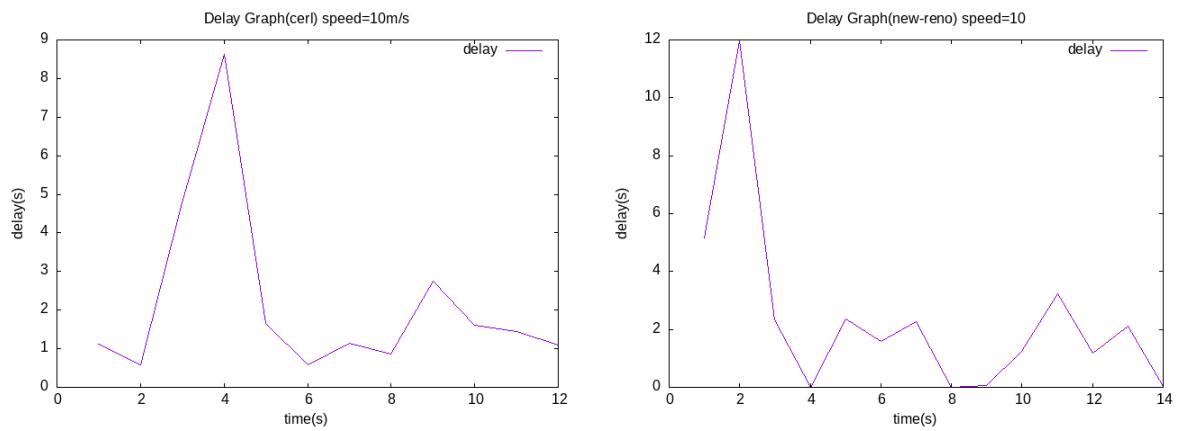
i) Speed of nodes=5 m/s



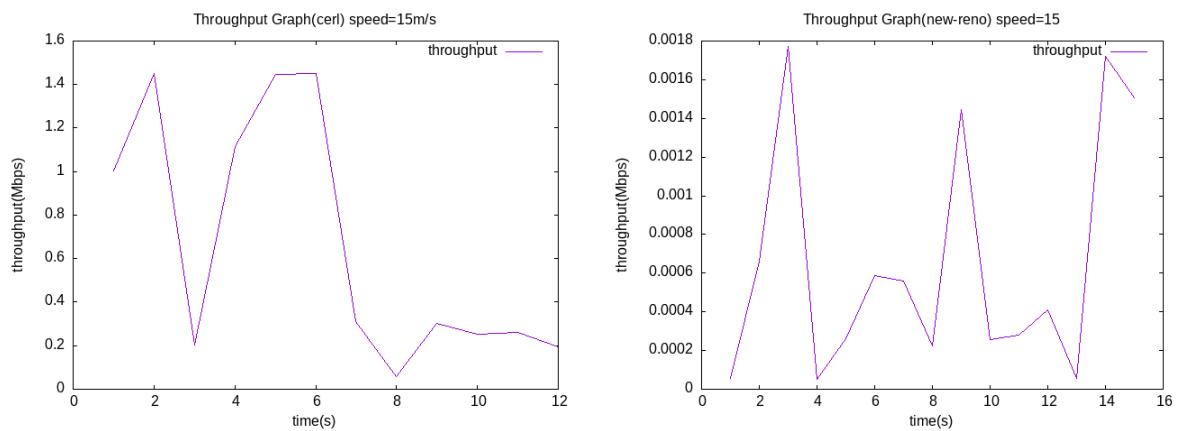


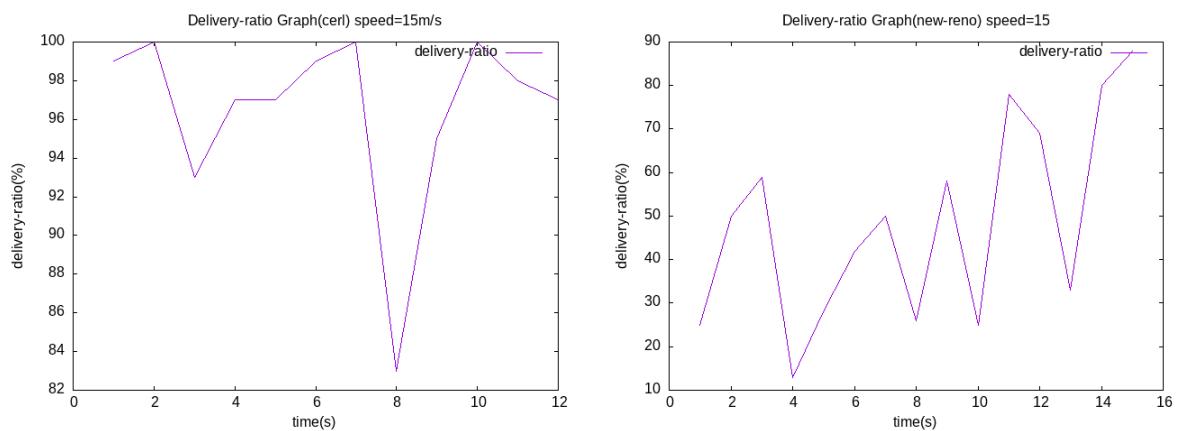
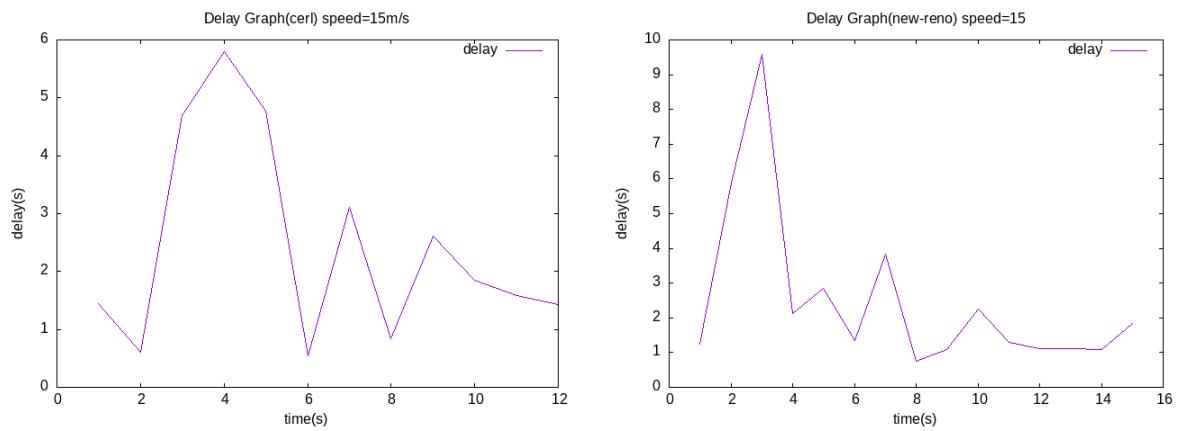
ii) Speed of nodes=10 m/s



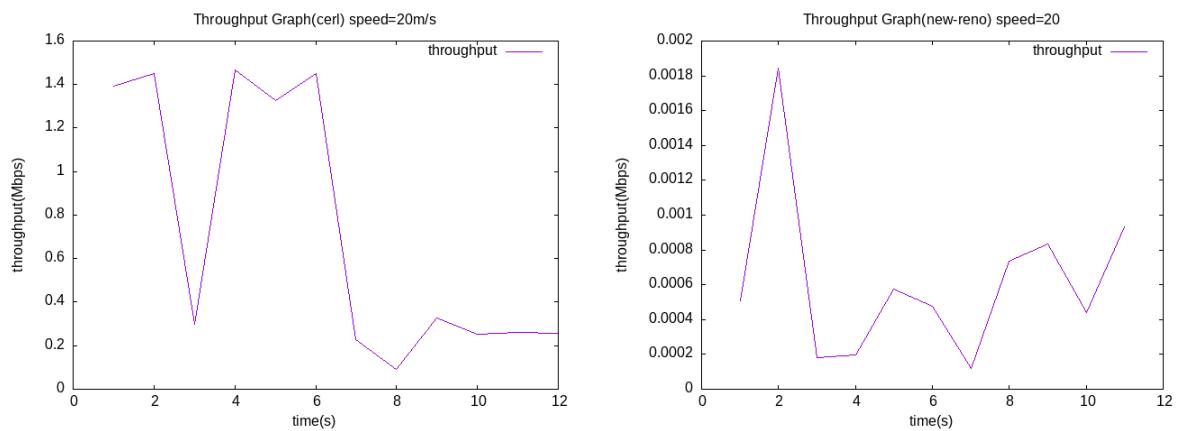


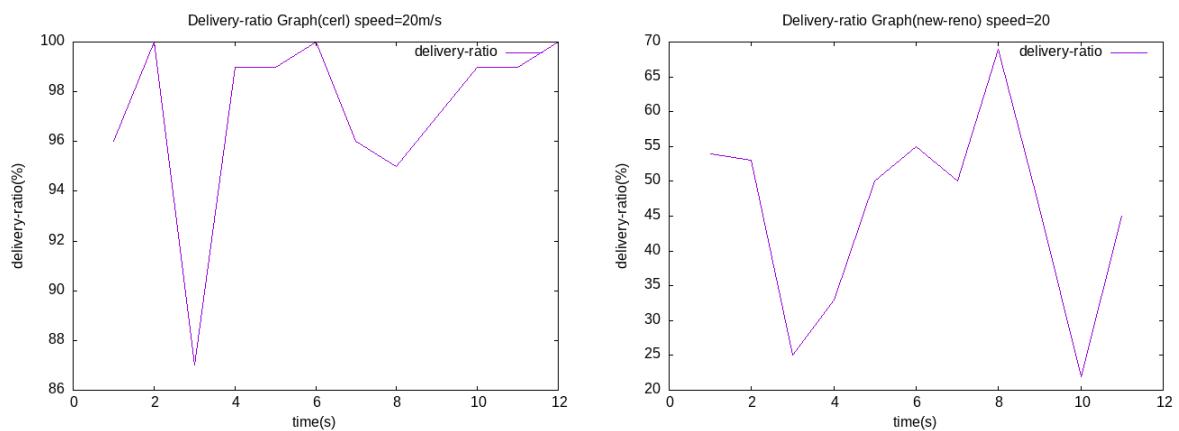
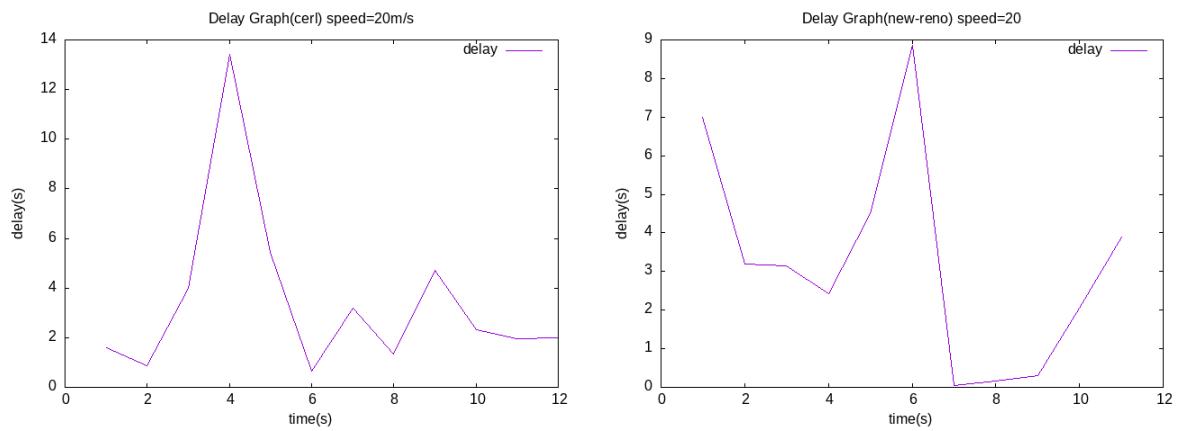
### iii) Speed of nodes=15 m/s



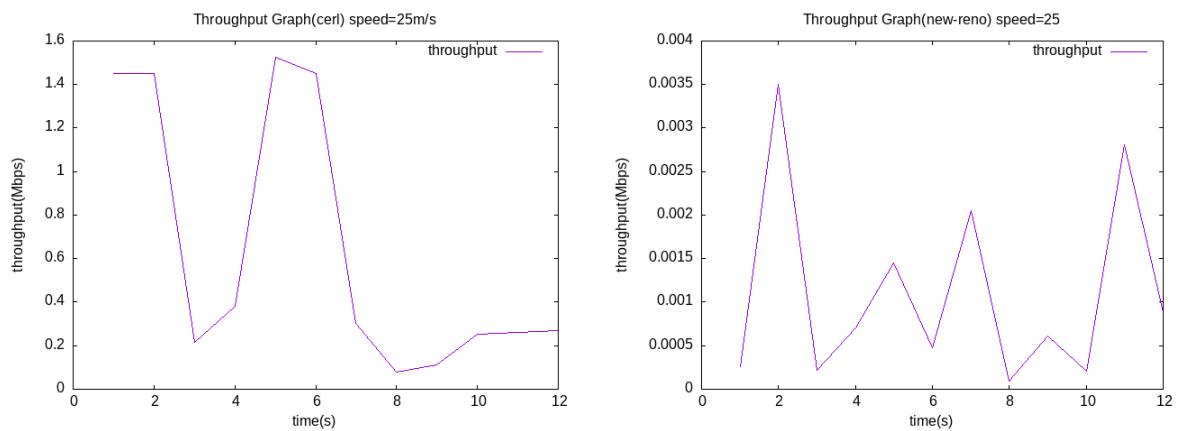


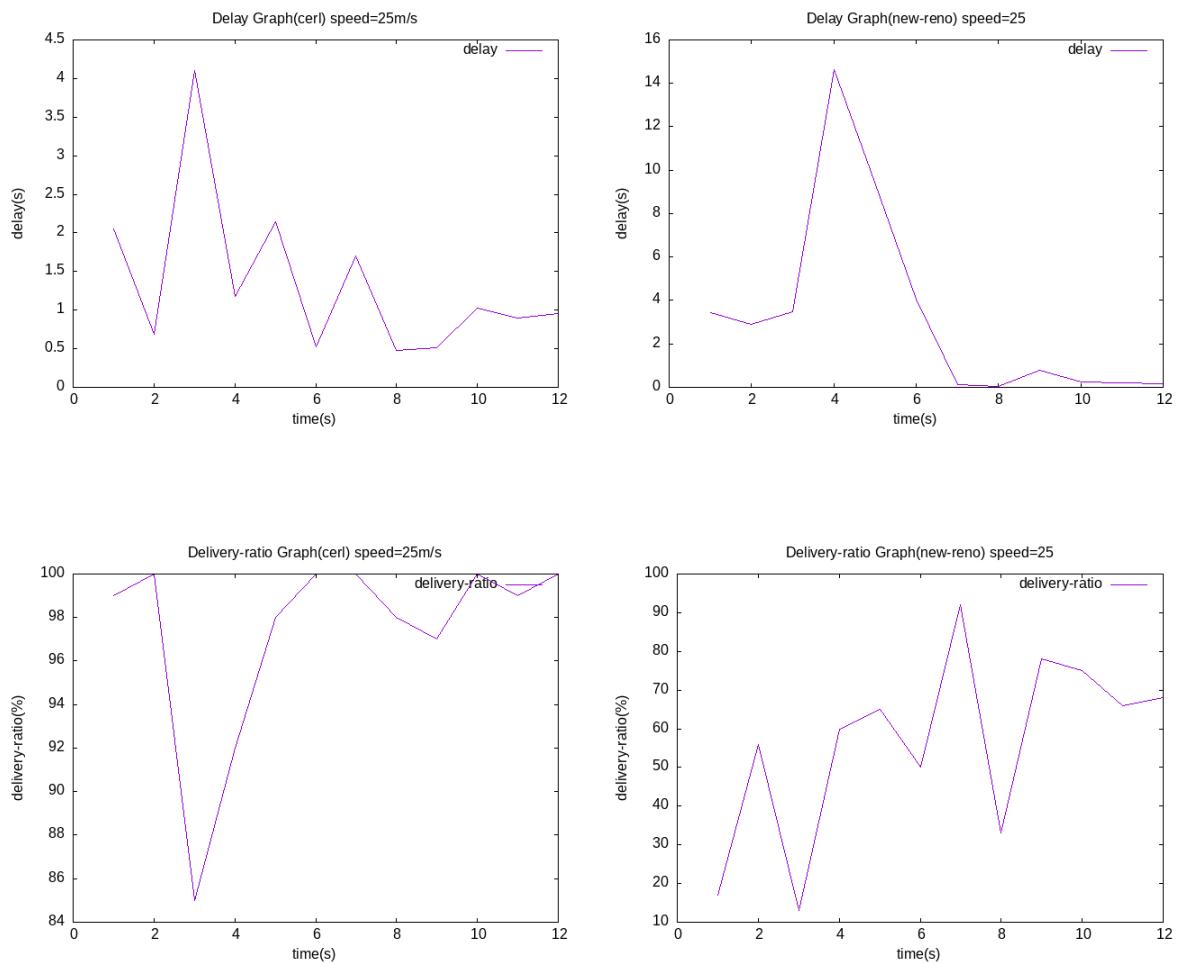
vi) Speed of nodes=20 m/s





v) Speed of nodes=25 m/s

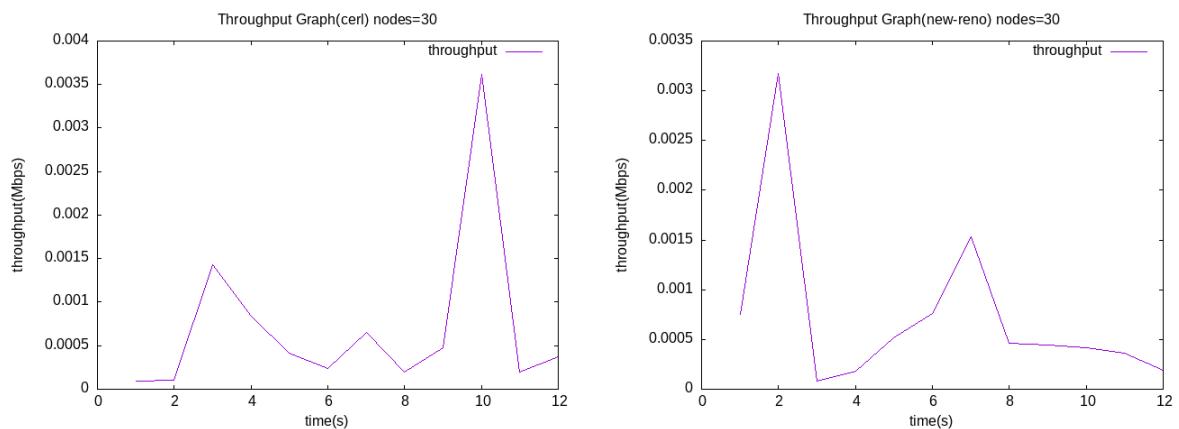


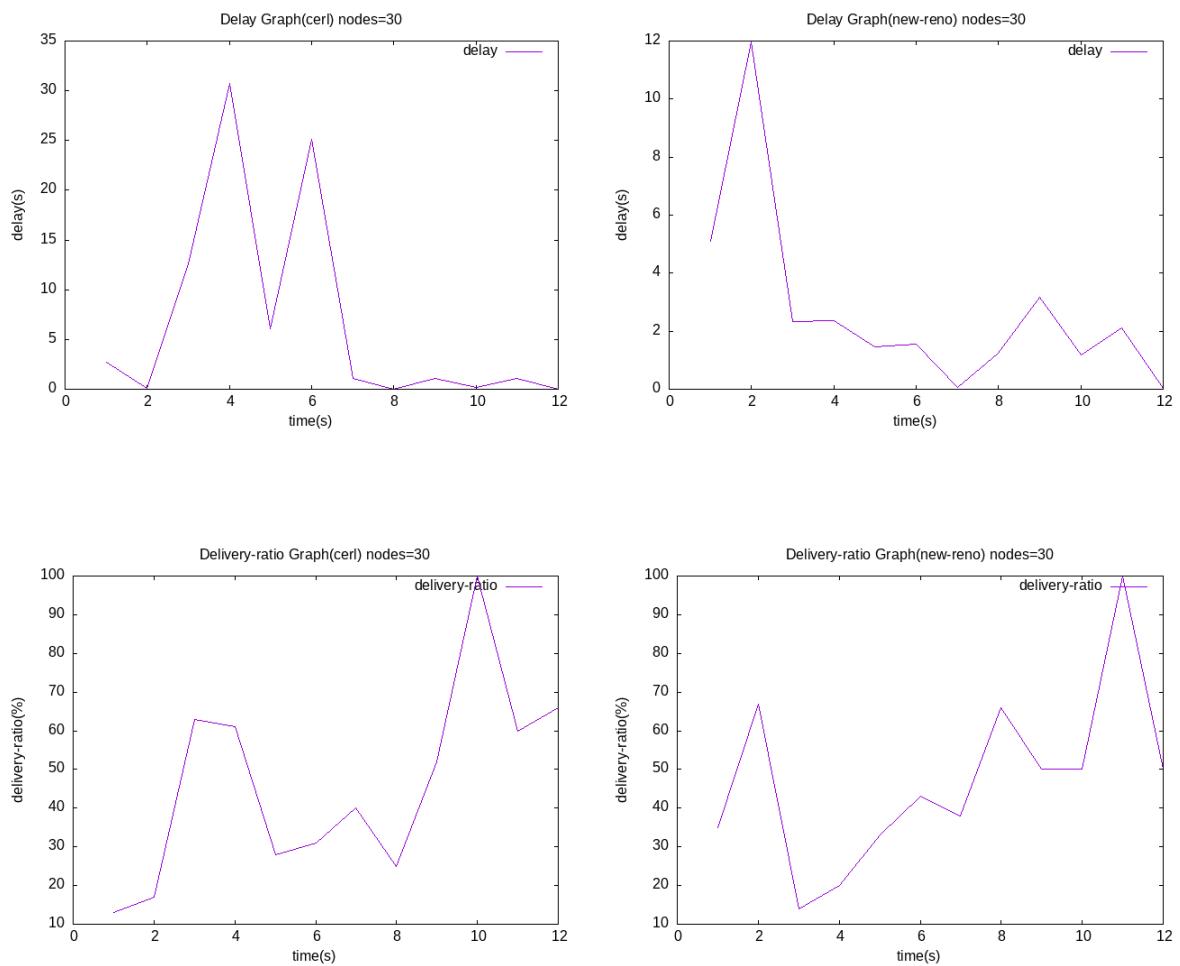


## For Wireless low-rate (e.g., 802.15.4) (mobile) --- LR-WPAN

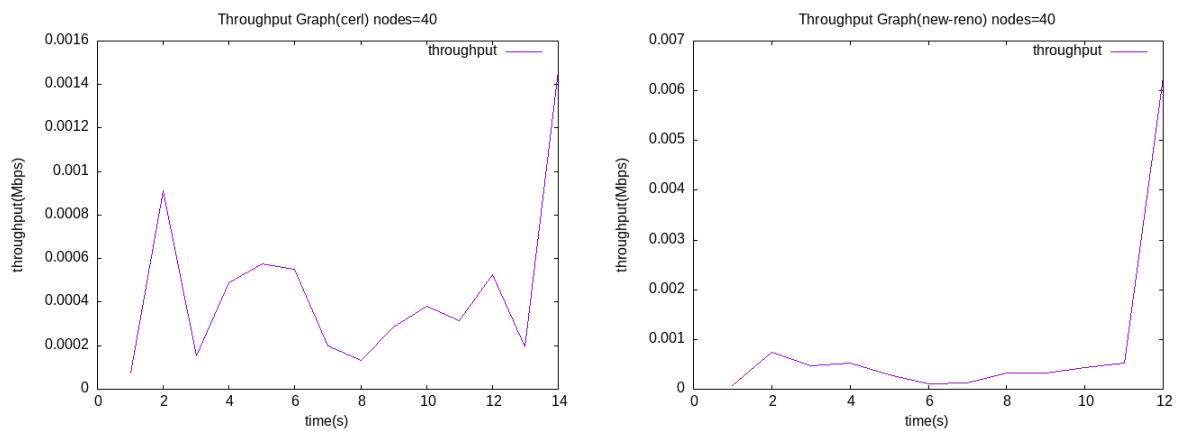
### 1)Varying number of nodes:

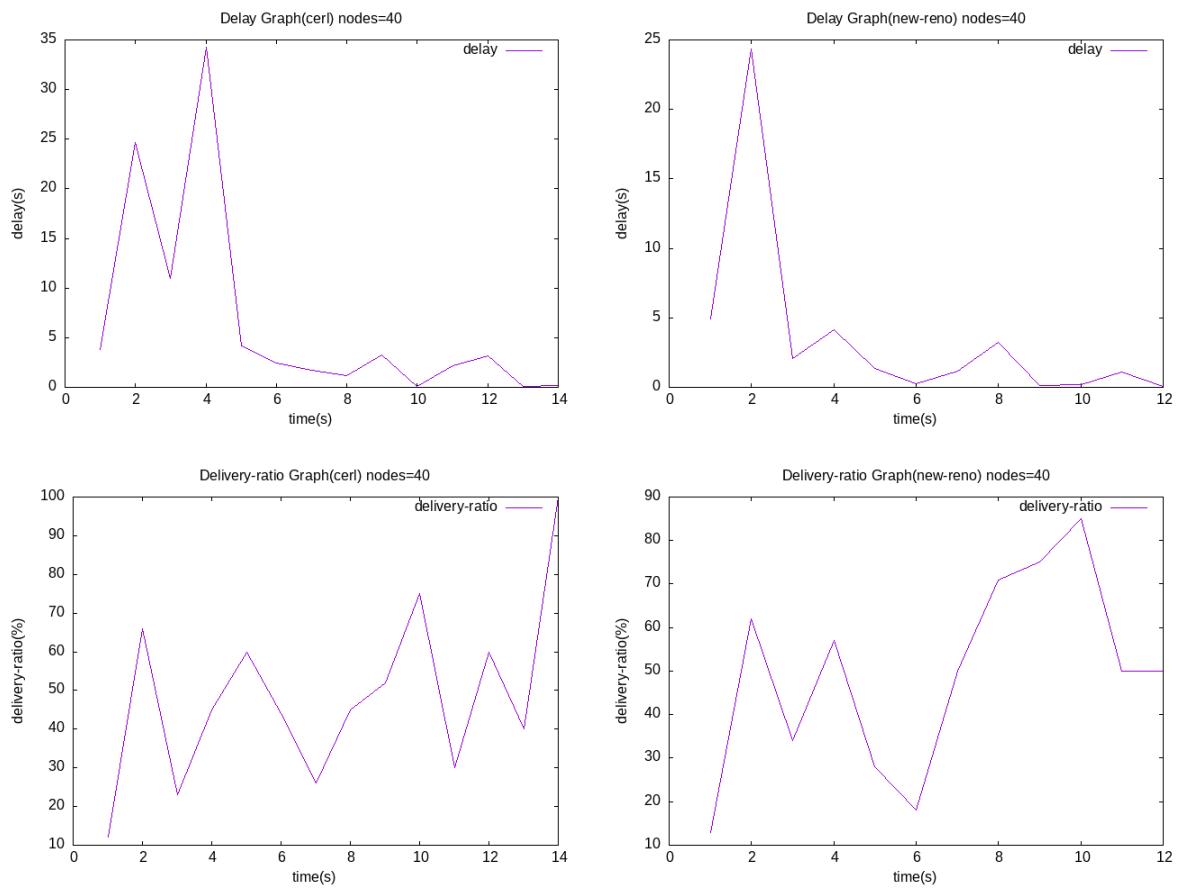
#### i)Number of nodes=30



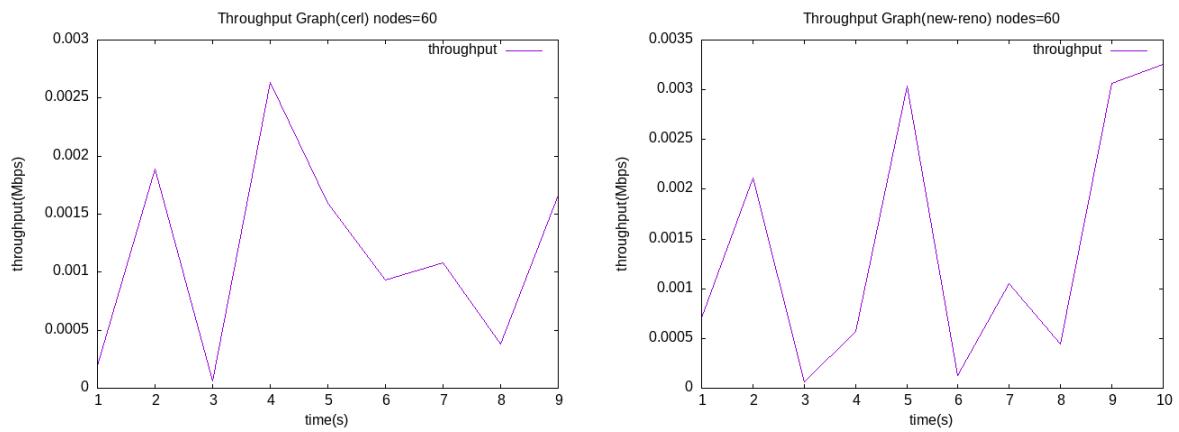


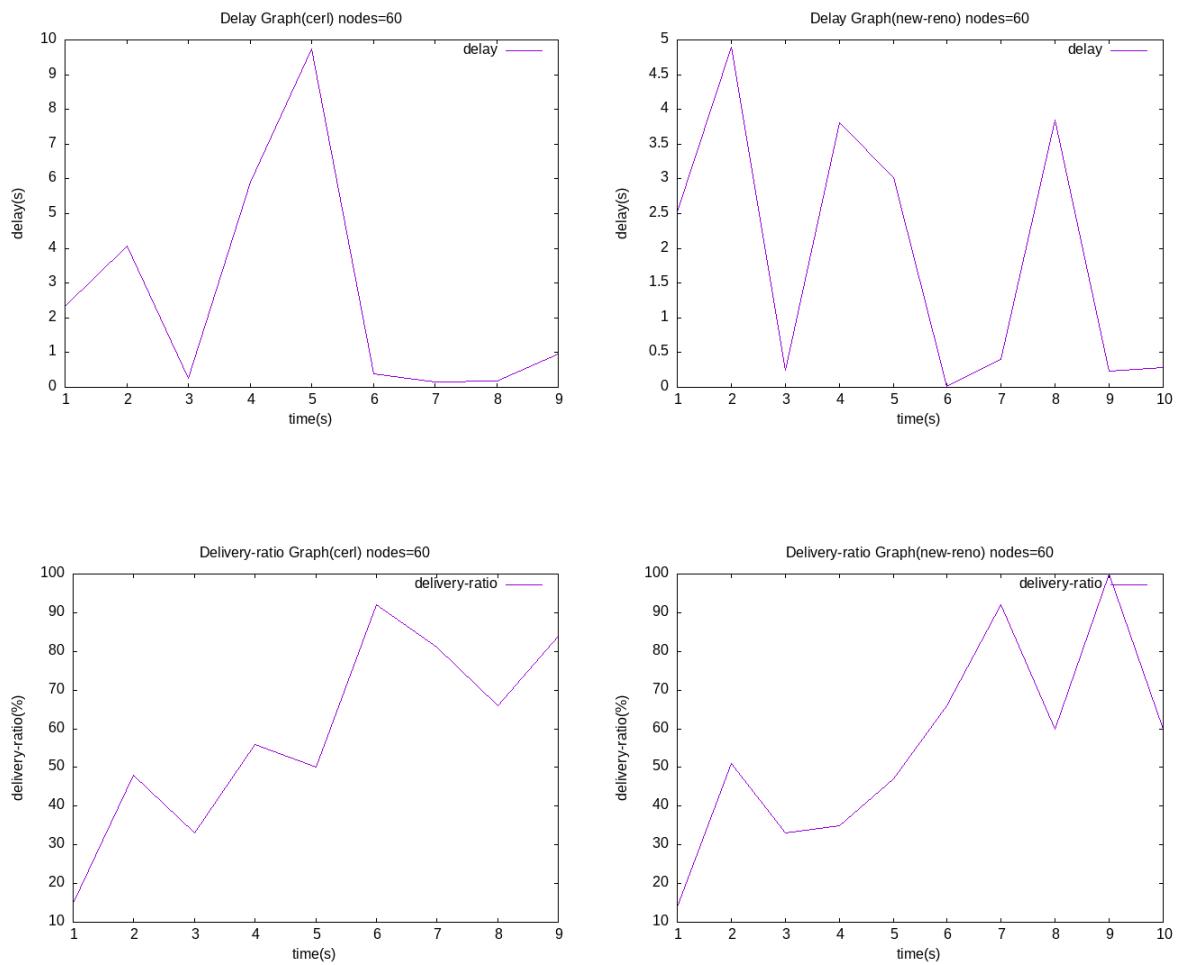
## ii) Number of nodes=40



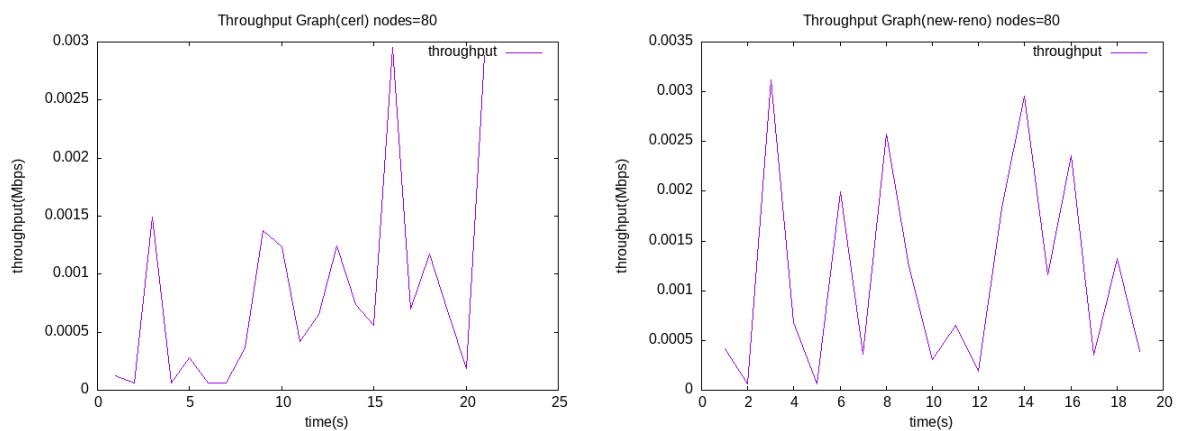


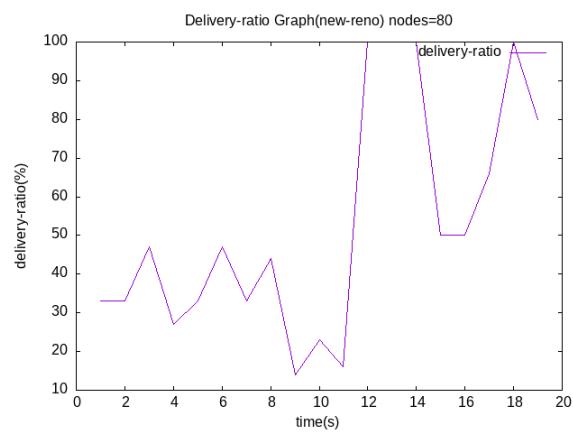
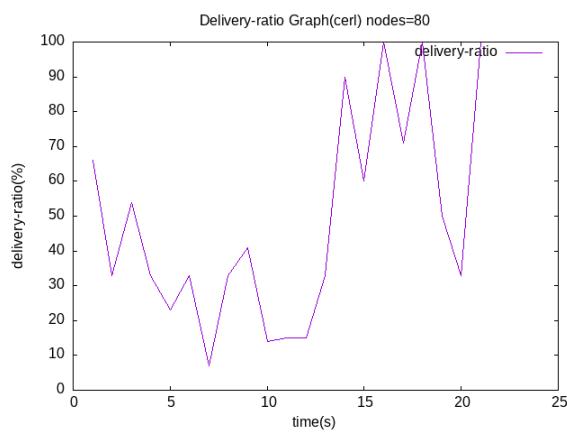
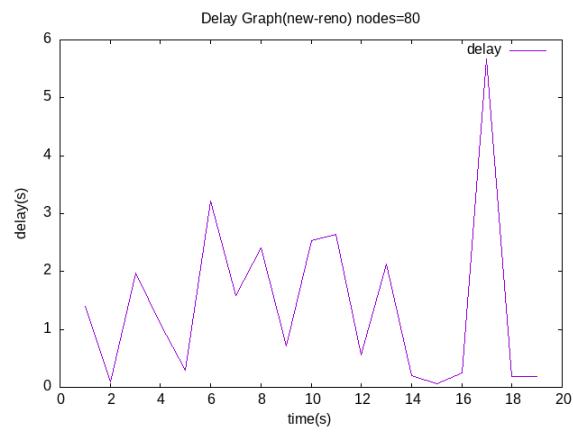
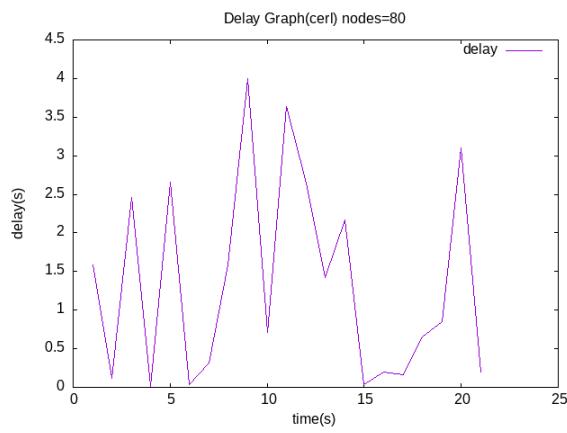
### iii) Number of nodes=60



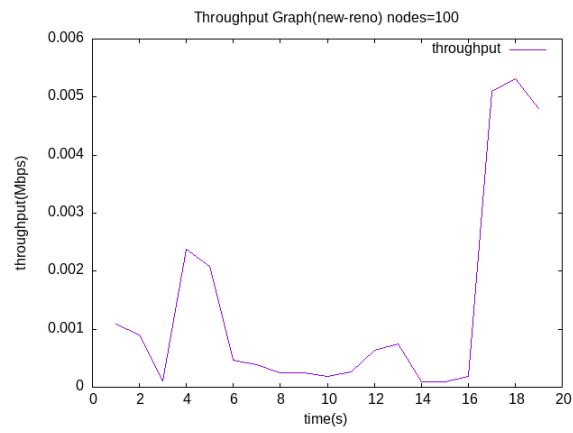
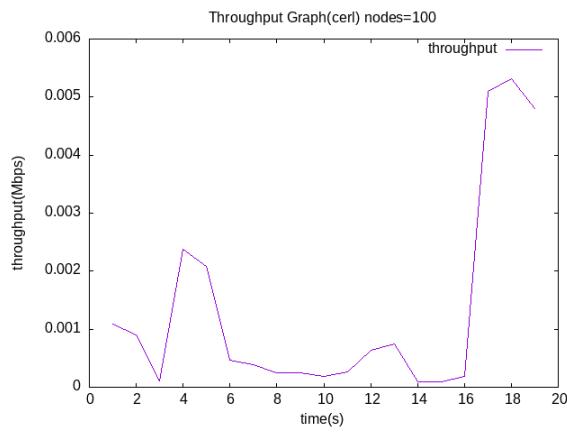


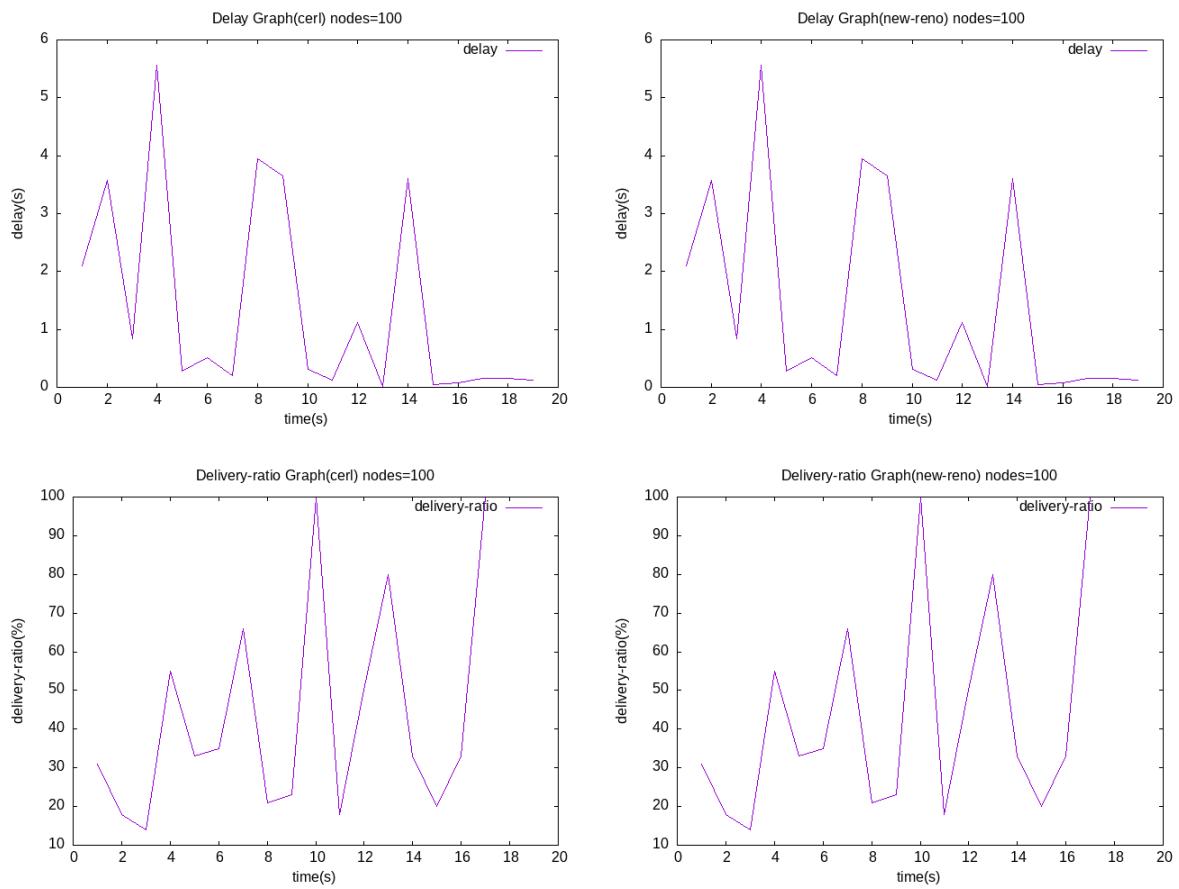
#### iv) Number of nodes=80





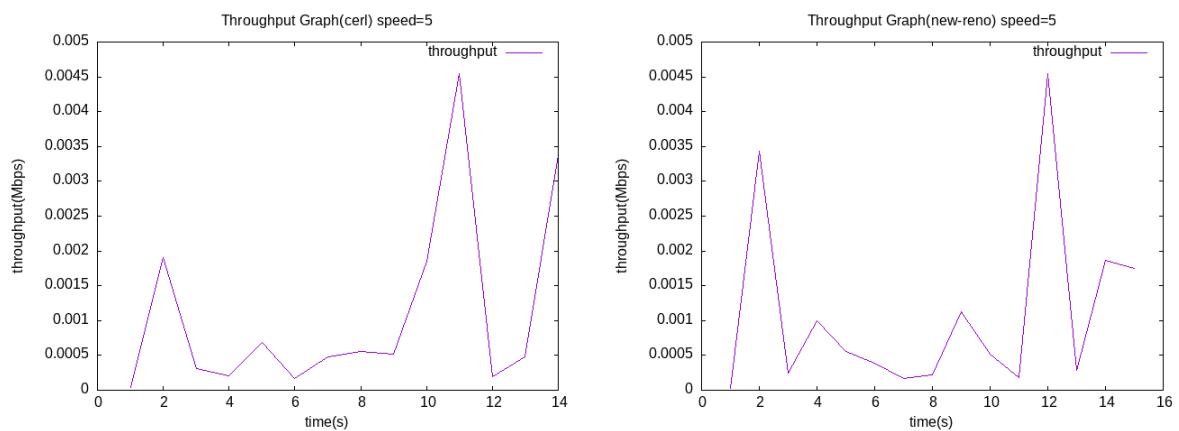
## v) Number of nodes=100

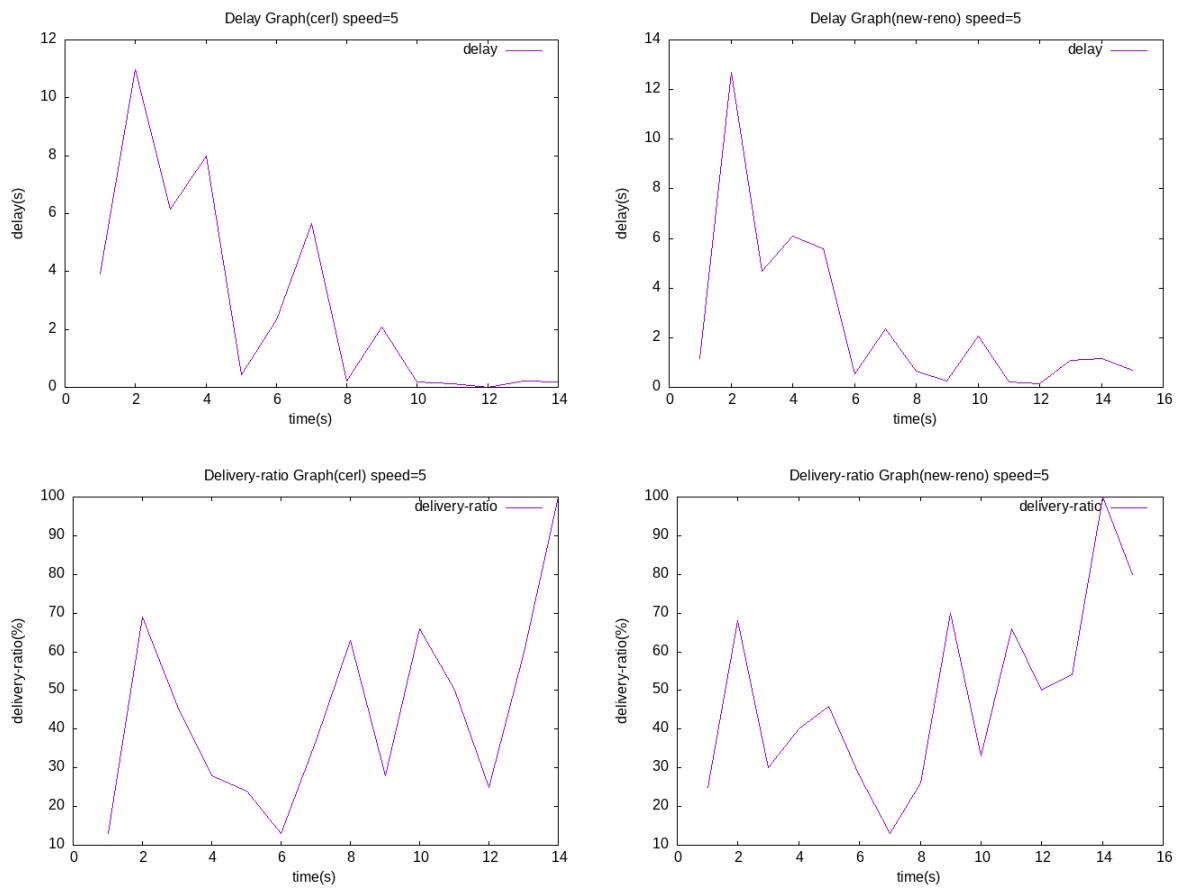




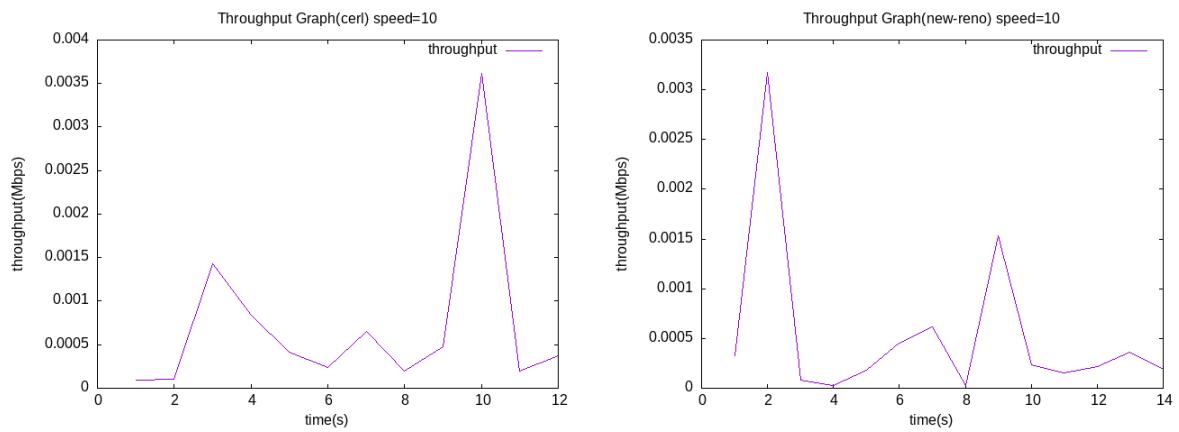
## 2)Varying speed of nodes:

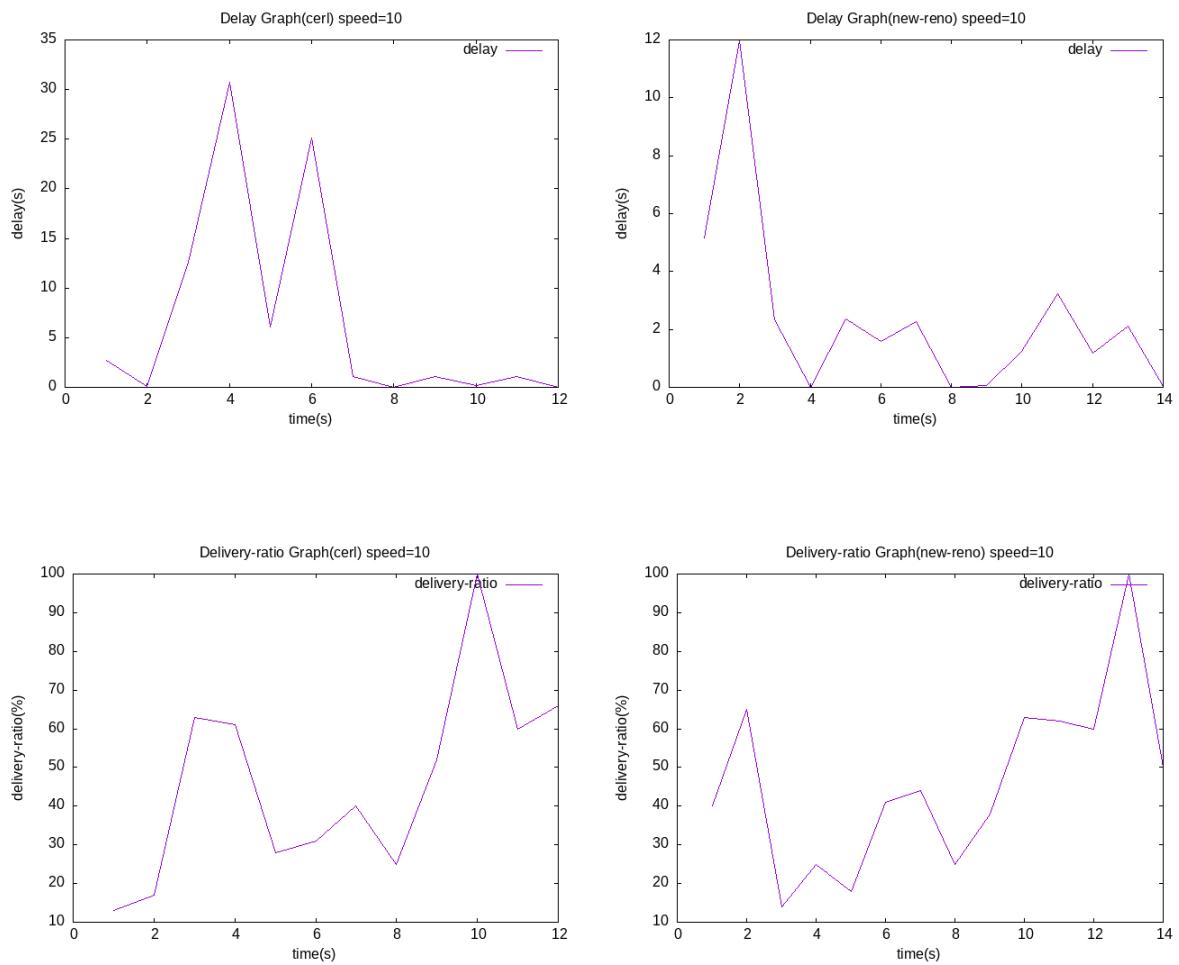
i) Speed of nodes=5 m/s



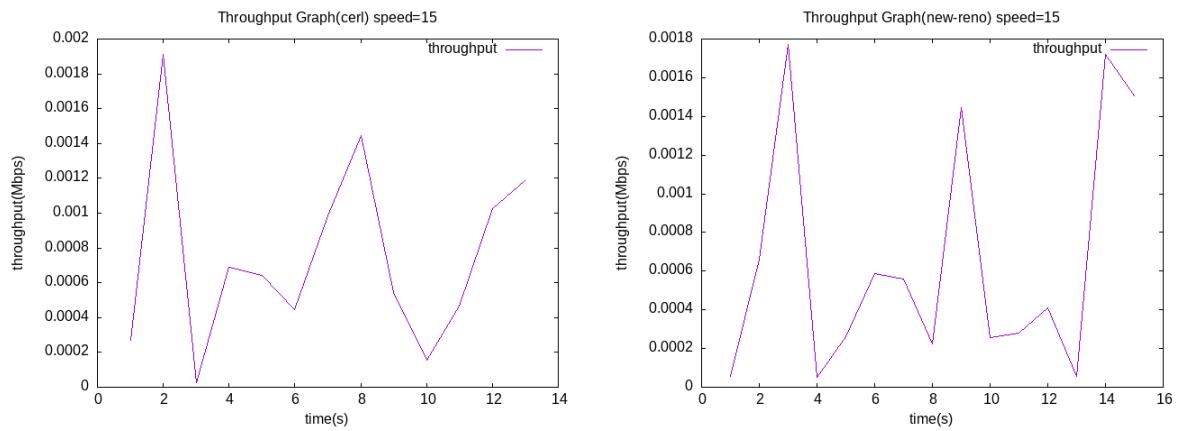


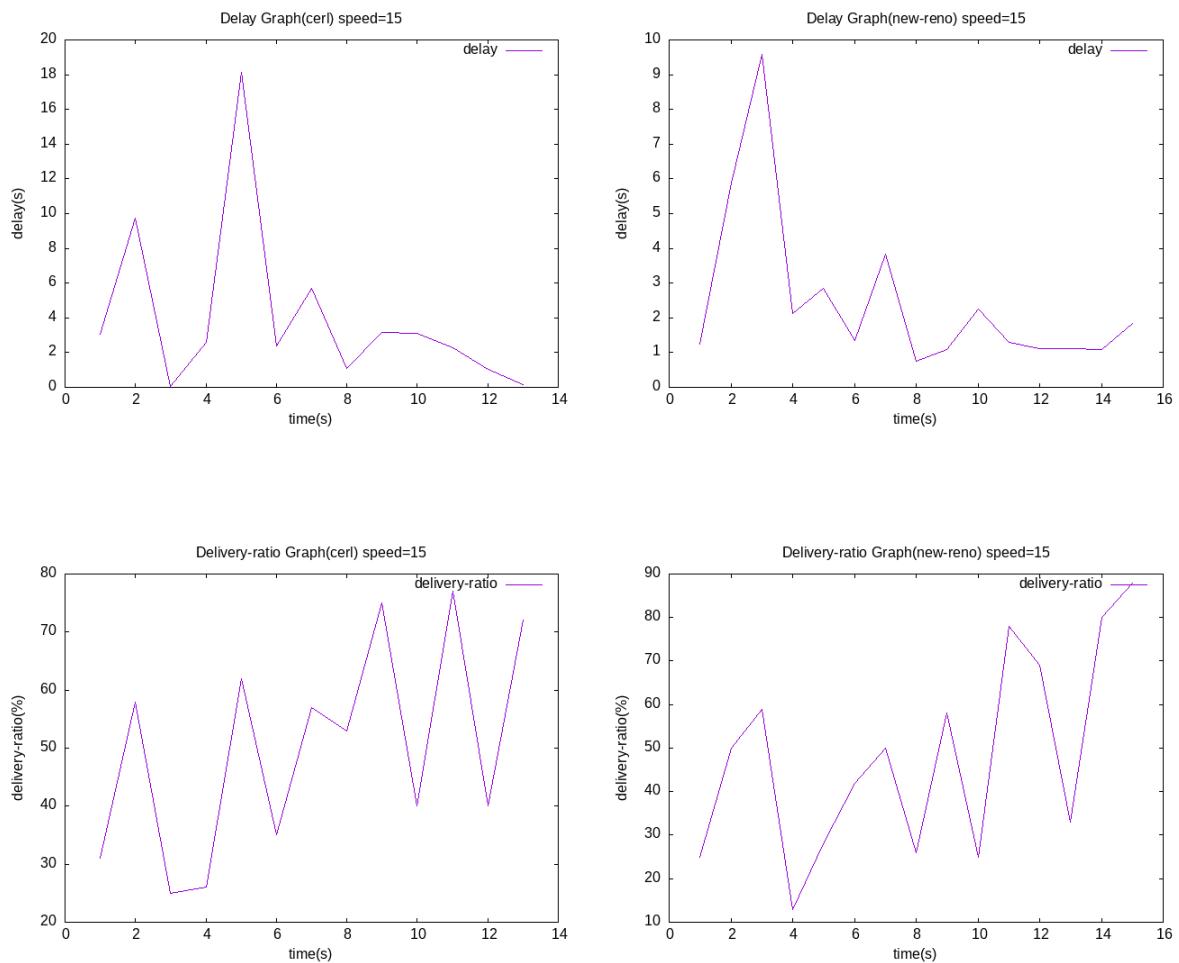
ii) Speed of nodes=10 m/s



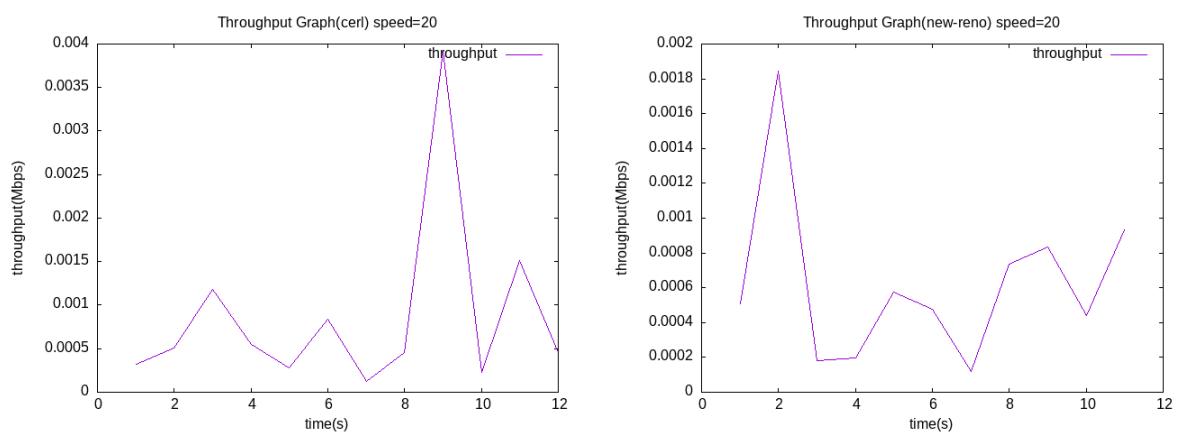


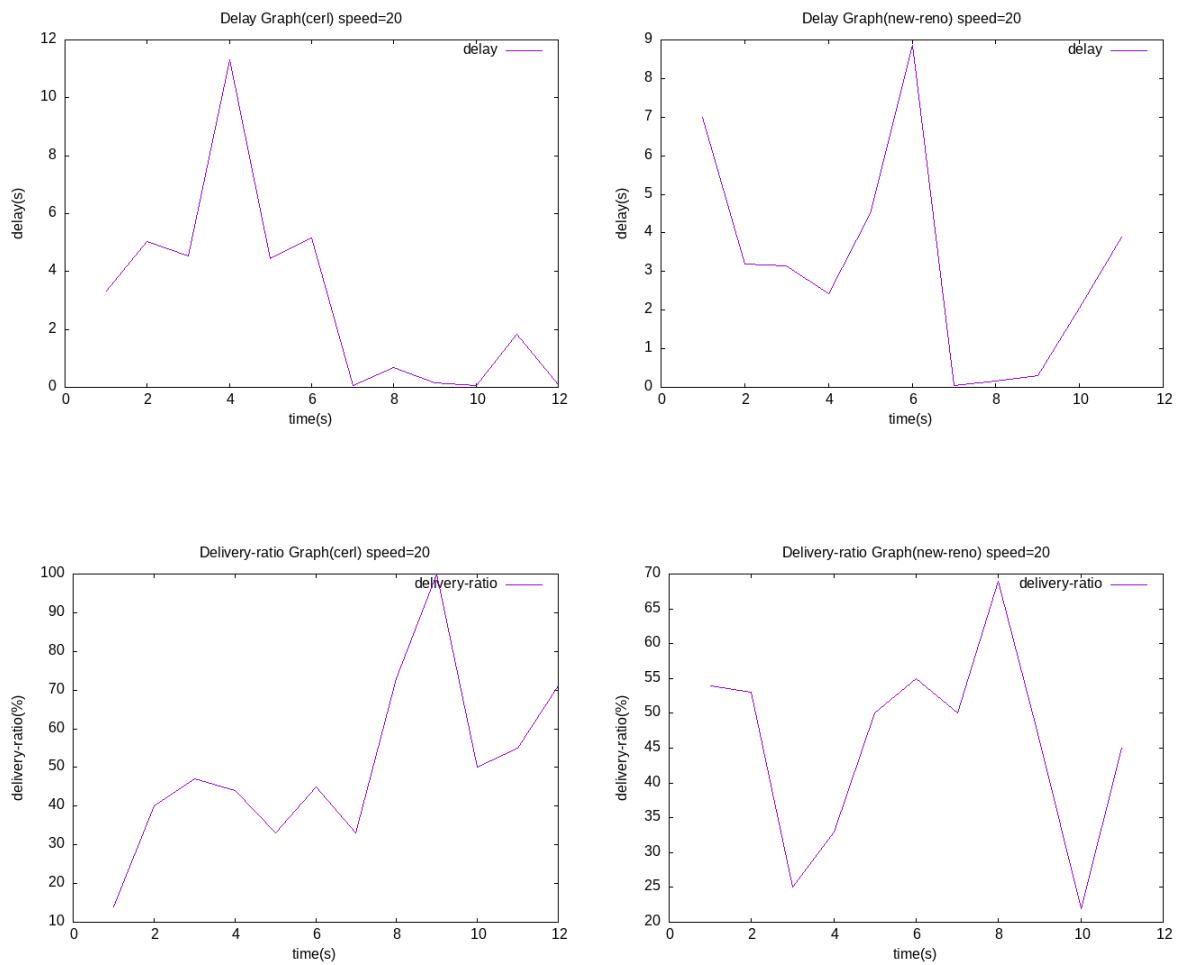
### iii) Speed of nodes=15 m/s



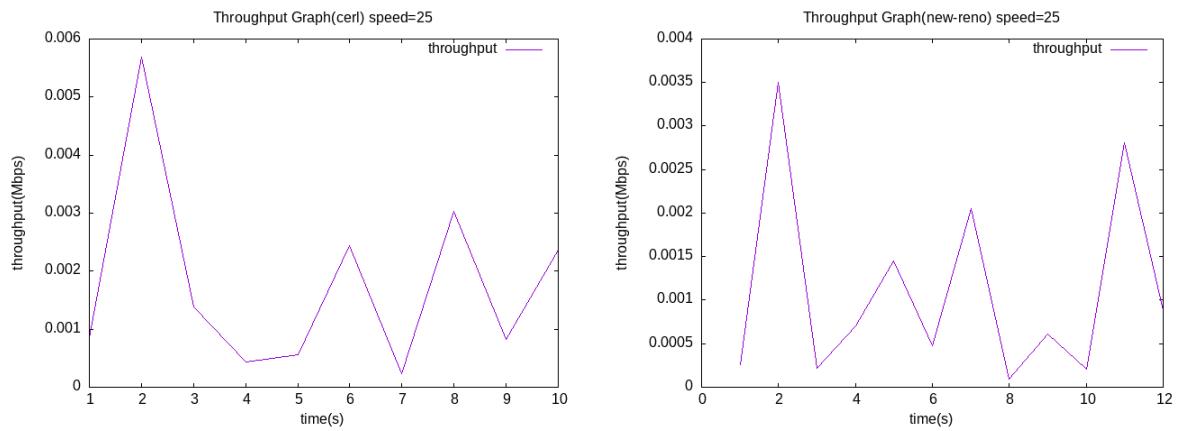


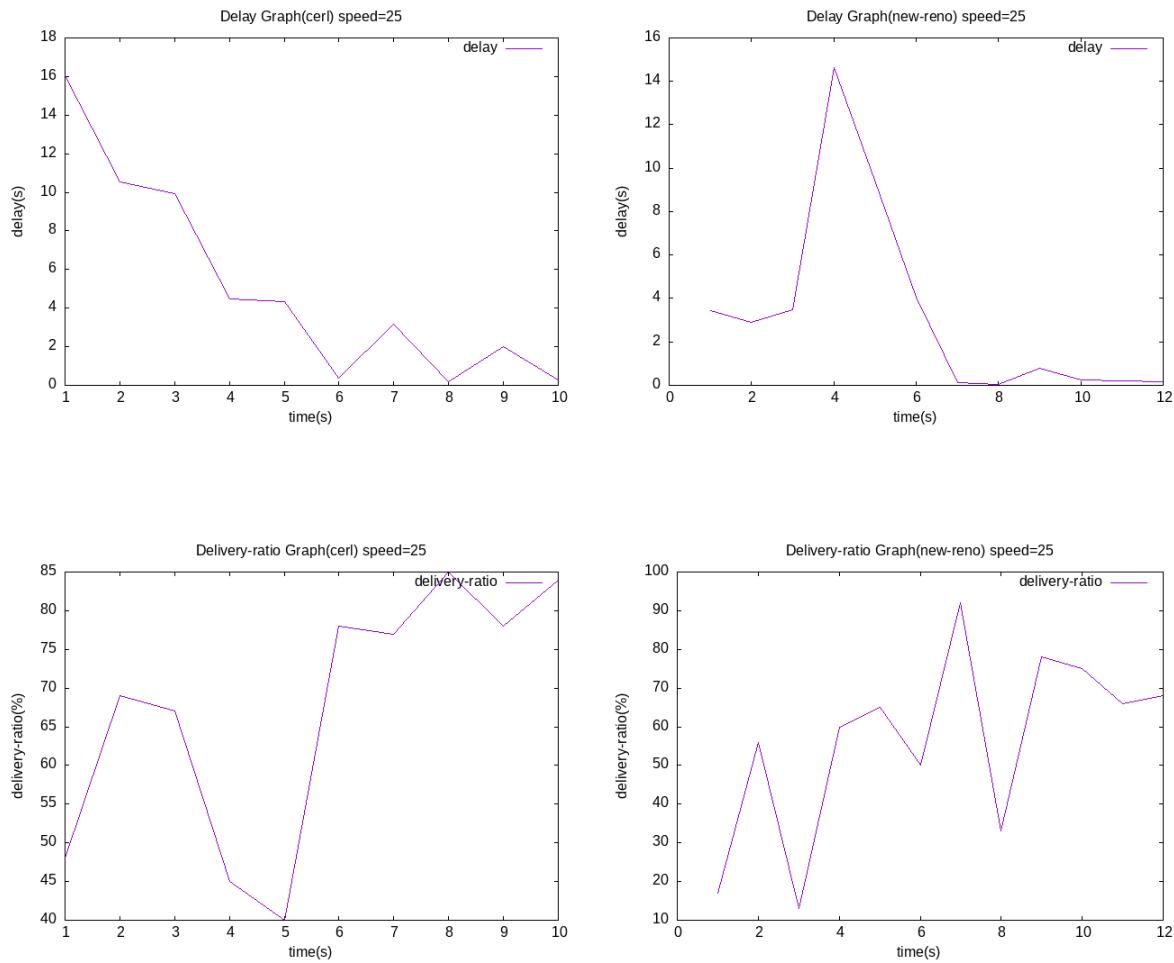
vi) Speed of nodes=20 m/s





v) Speed of nodes=25 m/s





## Section I: Explanation of results of Task B and Comparison with TCP-NewReno

*For the Wi-Fi network, we can see that when number of nodes increases, CERL gives better throughput than NewReno. End-to-end delay slightly differs. But delivery-ratio is much better for CERL. When increasing speed of nodes, throughput & delivery-ratio becomes surprisingly higher than New-Reno, though it takes much time.*

*For the LR-WPAN network*, it's clear that when number of nodes increases, at first CERL gives better throughput than NewReno but gradually it becomes almost same. End-to-end delay slightly differs like in Wi-Fi. But delivery-ratio is same or sometimes better for CERL. When increasing speed of nodes, throughput & delivery-ratio becomes better than New-Reno, but not as much in previous Wi-Fi network. As expected from observation, CERL takes relatively far more time than NewReno.

## **Section J: Comparing results with Paper**

In the paper, the researcher claims (*at section 5 Conclusion, page-15*) that TCP CERL is able to achieve excellent throughput gain than new-reno. The results and observations of the Task B, stated above, **completely agree** with their claim. Further, this project depicts that CERL gives far better performance in delivery-ratio also. But it's true that CERL takes much time for transmitting packets.