Bangladesh University of Engineering and Technology

CSE 322

Computer Networks Sessional

Curvilinear RED

An Improved RED Algorithm

Al Amru Bil Maruf
Student ID: 1805046
Section: A2
27 February 2023

Supervised by
Meher Afroz Sworna
Lecturer
Department of CSE, BUET

Introduction

Random Early Detection (RED) is an active queue management (AQM) scheme introduced to address the problem of network congestion in Internet routers. The motivation of Active Queue Management (AQM) comes from the philosophy that detecting congestion early and sending a notification before an actual scenario of congestion occurs.

Normally, The routers accept packets for a queue until the queue is full. They reject all the incoming packets after that until the queue decreases. The packet that arrived most recently is dropped when the queue is full. In an AQM scheme such as RED, the packages start getting dropped with a drop probability before the queue is full. This is an implicit way of declaring there is a chance that the queue is getting full and packets might redirect their path to another router to avoid congestion or getting dropped.

Random Early Detection: RED

The operation of RED is quite simple. For each new packet that arrives in the gateway, RED computes the average queue size (the average no. of packets in the buffer of the router) and compares the result with two preset thresholds: a minimum threshold (min_{th}) and a maximum threshold (max_{th}).

Then the following actions are taken:

I. If the average is less than the minimum threshold, the package is not dropped.

II. If the average is between the minimum threshold and the maximum threshold, the package is dropped with a probability,

$$max_p\left(\frac{avg-min_{th}}{max_{th}-min_{th}}\right)$$

where max_p is the maximum probability

III. If the average is more than the maximum threshold, the package is dropped.

The drop probability function can be written as,

$$P = \begin{cases} 0 & ; \quad avg < min_{th} \\ max_p\left(\dfrac{avg-min_{th}}{max_{th}-min_{th}}\right) & ; \quad min_{th} \le avg < max_{th} \\ 1 & ; \quad max_{th} < avg \end{cases}$$

Curvilinear Random Early Detection: CLRED

There are a few RED algorithms around such as GRED, NLRED, MRED etc. each of which implicitly tries to tackle congestion by dropping packets with the help of the probability function. For Curvilinear RED, the drop probability function was updated and divided into two parts: a curve and a linear part. Hence came the name ‘Curvilinear RED’. A new threshold (mid_{th}) is taken from the average of the min_{th} and max_{th}. Then the drop probability works in a parabolic (curve) way if the average queue length falls between min_{th} and mid_{th}. And if the average queue length falls between mid_{th} and max_{th}, the drop probability is changed in a linear manner.

The drop probability function can be written as,

$$P = \begin{cases} 0 & ; \quad avg < min_{th} \\ 4(1 - max_p)\left(\dfrac{avg - min_{th}}{max_{th} - min_{th}}\right)^2 & ; \quad min_{th} \le avg < mid_{th} \\ 1 - max_p) + 2max_p(\dfrac{avg - mid_{th}}{max_{th} - min_{th}}) & ; \quad mid_{th} \le avg < max_{th} \\ 1 & ; \quad max_{th} < avg \end{cases}$$

Network Topologies

- Wireless 802.11 static: Nodes were divided into rectangular grids with random source communicating with a particular sink.

No. of Nodes	Grid
20	5×4
40	8×5
60	10×6
80	10×8
100	10×10

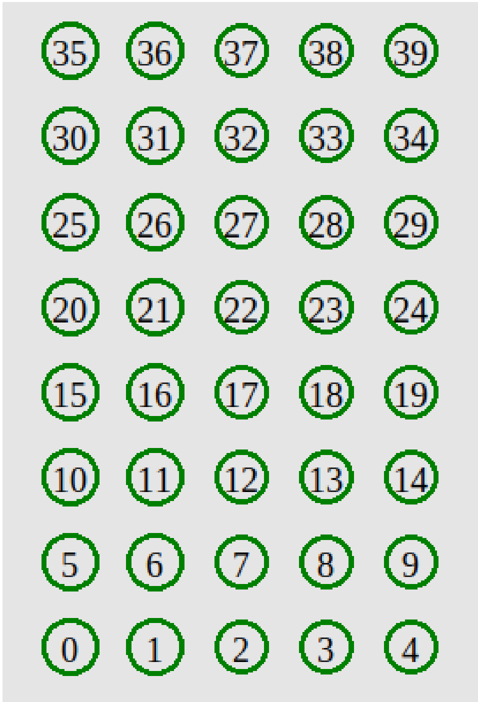

Fig. Grid topology for 40 wireless nodes

- Wired: Dumbbell shaped with two bottleneck nodes. Two nodes were chosen as bottlenecks. The rest were divided into two groups. One group was connected to one bottleneck node and the other group was connected to another bottleneck node. A node from one group was made to communicate with another node from another group through the two bottleneck nodes.

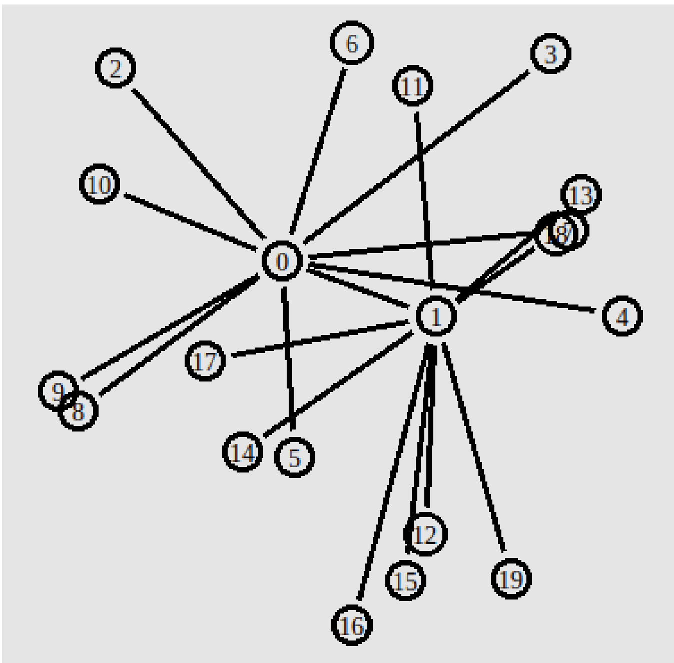

Fig. Dumbbell topology for 20 wired nodes

Parameters

General parameters that were varied to retrieve different outputs:

1. No. of nodes: It was varied from 20 to 100 with an increment of 20 in each iteration, i.e., 20, 40, 60, 80, 100.
2. No. of flows: It was varied from 10 to 50 with an increment of 20 in each iteration, i.e., 10, 20, 30, 40, 50.
3. No. of packets per second : It was varied from 100 to 500 with an increment of 20 in each iteration, i.e., 100, 200, 300, 400, 500.

Parameters that were varied for *Wireless 802.11 static* only:

1. Area: For a square area, the length of one side was varied from 250 m to 1250 m with an increment of 250 m in each iteration, i.e., 250, 500, 750, 1000, 1250.

Results and Graphs

Wired Topology

1. Throughput vs No. of Nodes in Wired topology

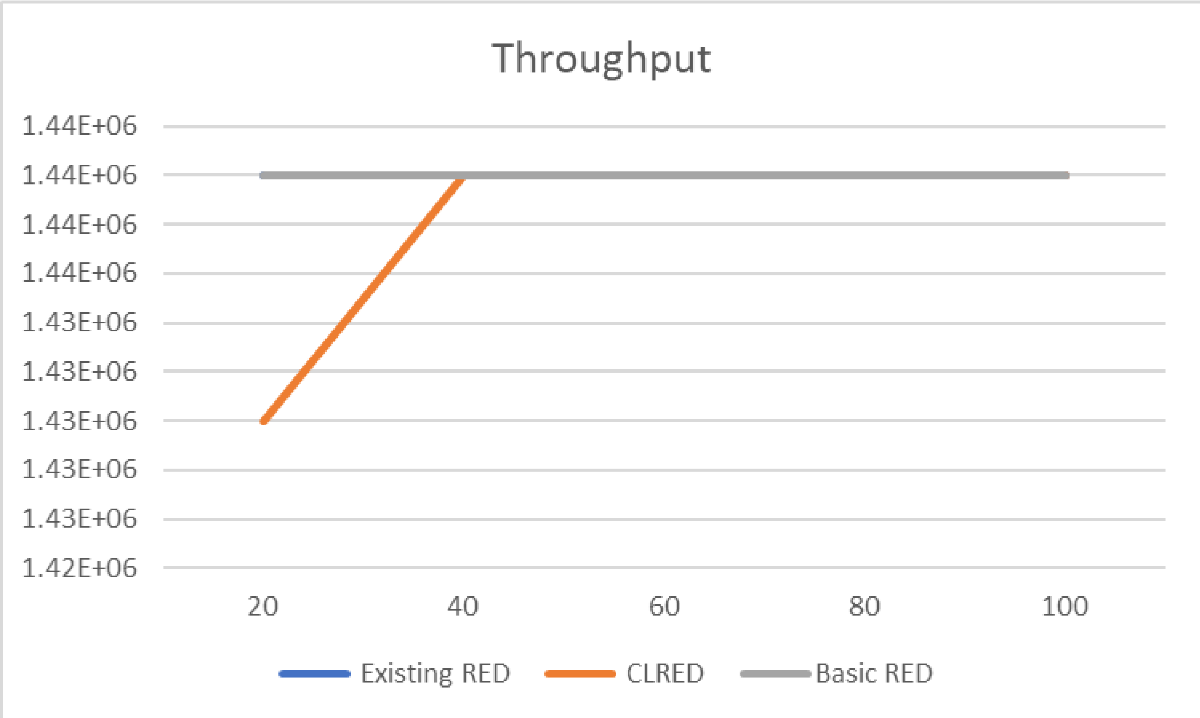

Fig. Throughput vs No. of Nodes in Wired topology

Remarks: All three RED mechanisms perform almost similarly for throughputs vs no. of nodes in Wired topology. In fact, they are almost inseparable from each other in the graph.

2. Average Delay vs No. of Nodes in Wired topology

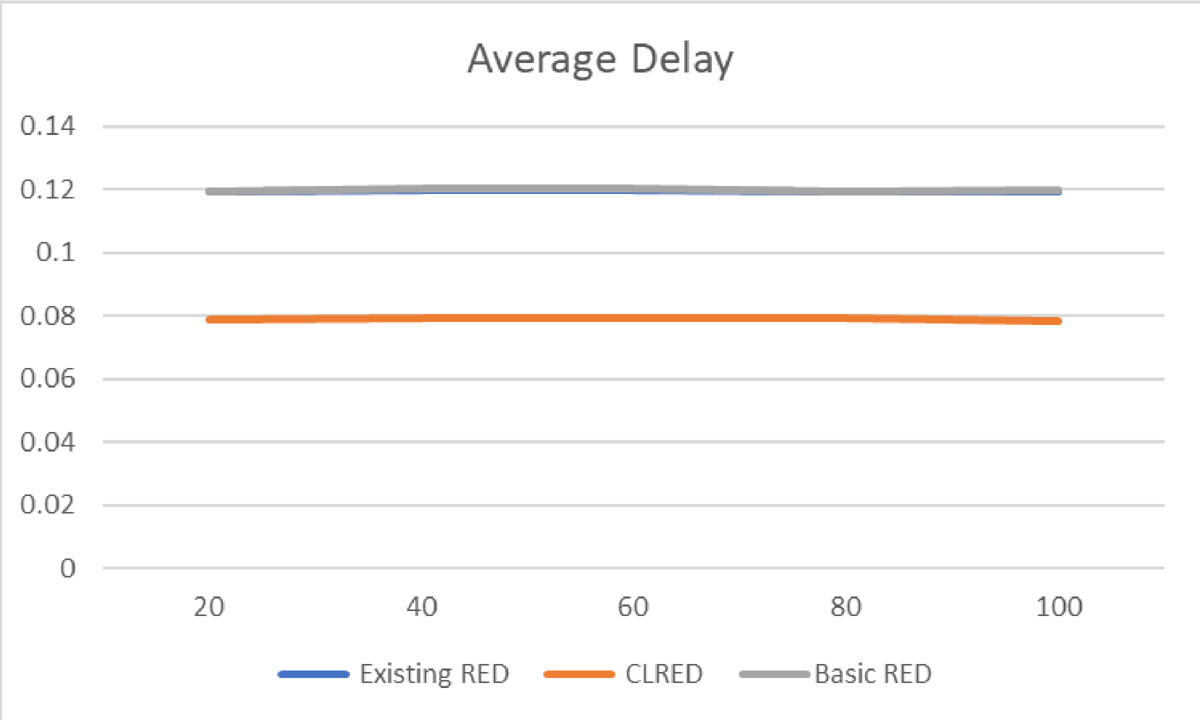

Fig. Average Delay vs No. of Nodes in Wired topology

Remarks: CLRED performs better than other two mechanisms for the average delay vs no. of nodes in Wired topology. The average delay for the other mechanism is in the 0.11-0.12 range, whereas that of CLRED belongs to the 0.078-0.079 group.

3. Delivery Ratio vs No. of Nodes in Wired topology

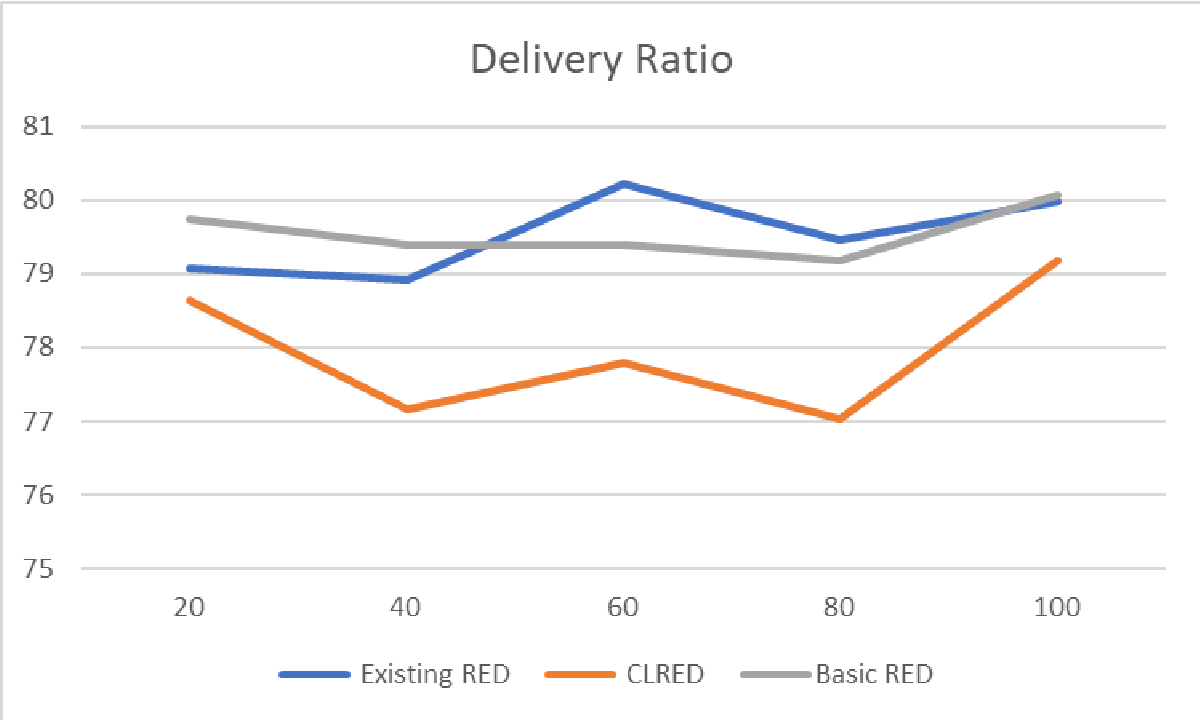

Fig. Delivery Ratio vs No. of Nodes in Wired topology

Remarks: Other two mechanisms perform slightly better than CLRED in case of delivery ratio vs no. of nodes in Wired topology. But the difference is marginal and they can be regarded as performances of the same range.

4. Drop Ratio vs No. of Nodes in Wired topology

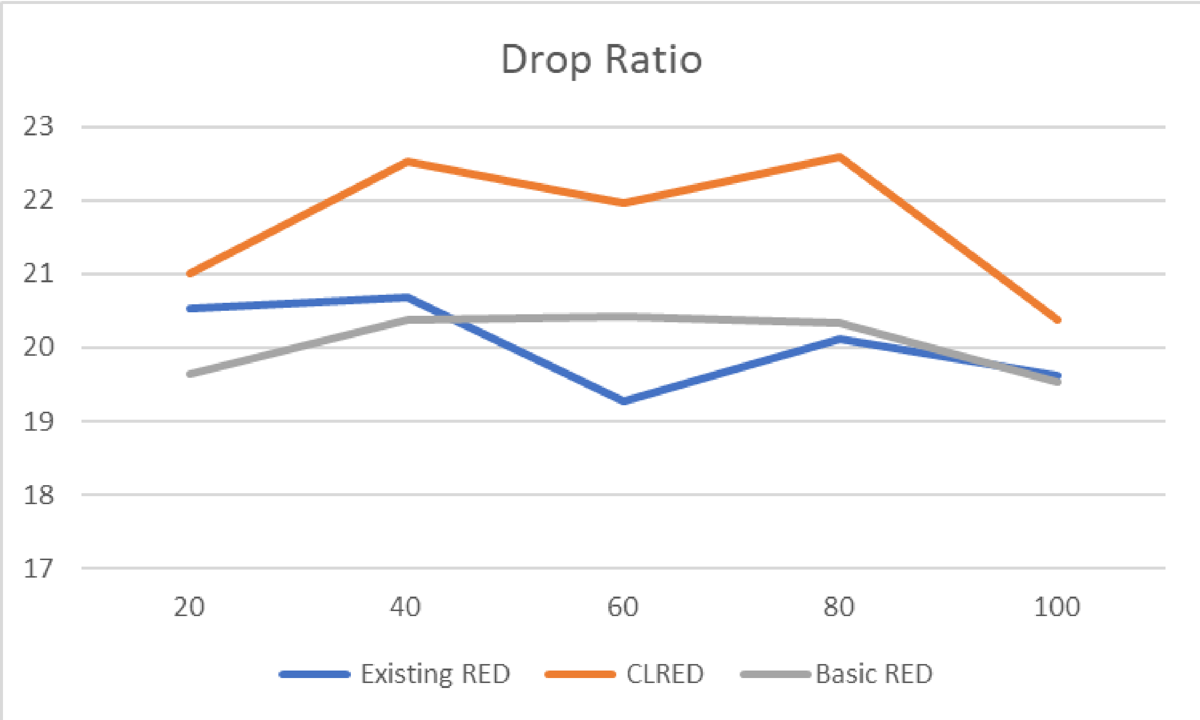

Fig. Drop Ratio vs No. of Nodes in Wired topology

Remarks: Other two mechanisms also perform slightly better than CLRED in case of drop ratio vs no. of nodes in Wired topology. But, like delivery ratio vs no. of nodes, the difference is marginal and they can be regarded as performances of the same range.

5. Throughput vs No. of Flows in Wired topology

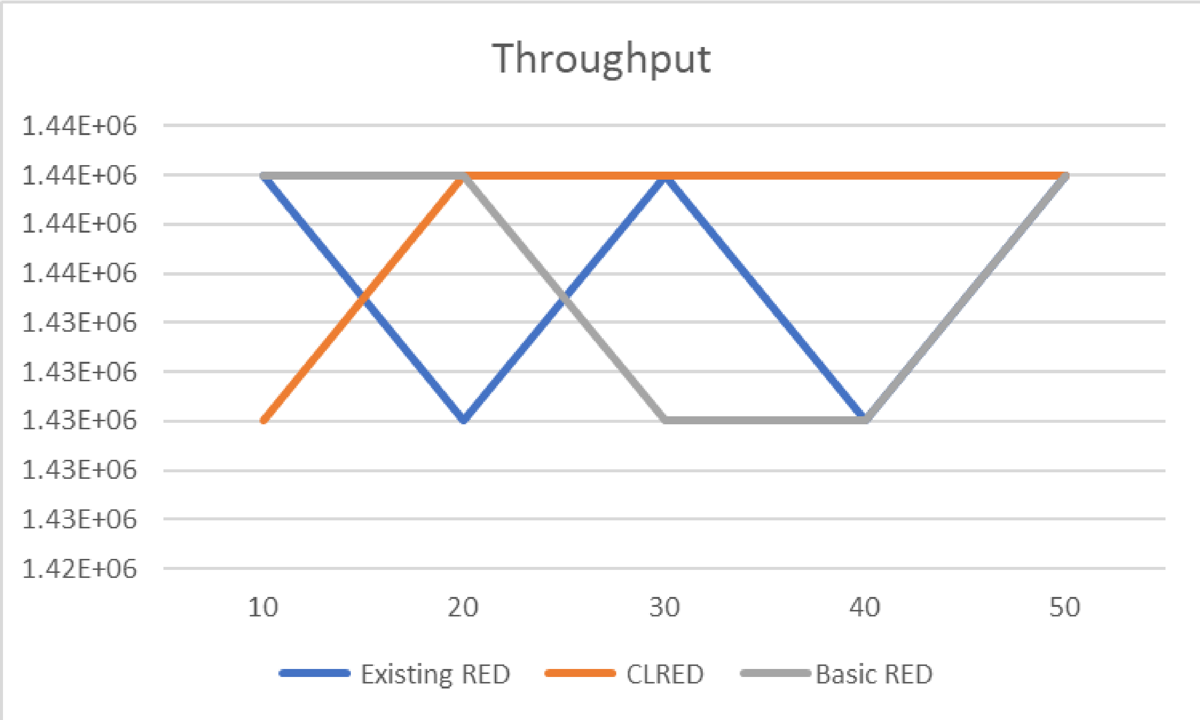

Fig. Throughput vs No. of Flows in Wired topology

Remarks: All three RED mechanisms perform almost similarly for throughputs vs no. of flows in Wired topology.

6. Average Delay vs No. of Flows in Wired topology

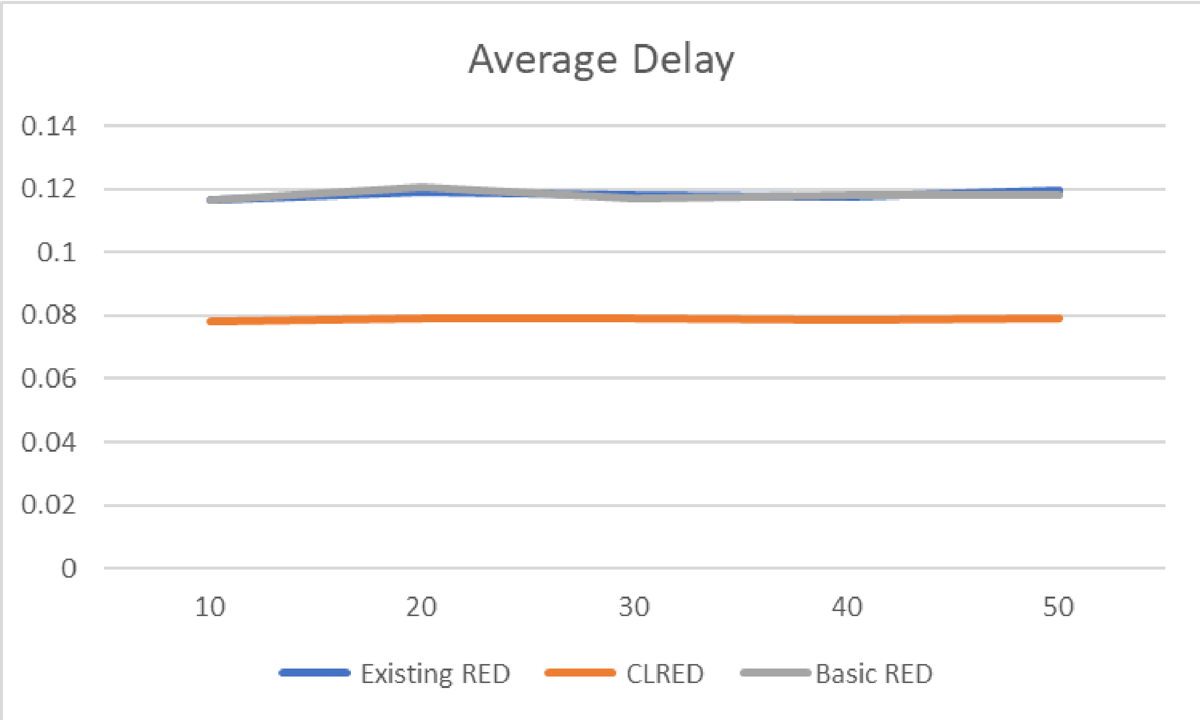

Fig. Average Delay vs No. of Flows in Wired topology

Remarks: CLRED performs better than other two mechanisms for the average delay vs no. of flows in Wired topology. The average delay for the other mechanism is in the 0.11-0.12 range, whereas that of CLRED belongs to the 0.078-0.079 group.

7. Delivery Ratio vs No. of Flows in Wired topology

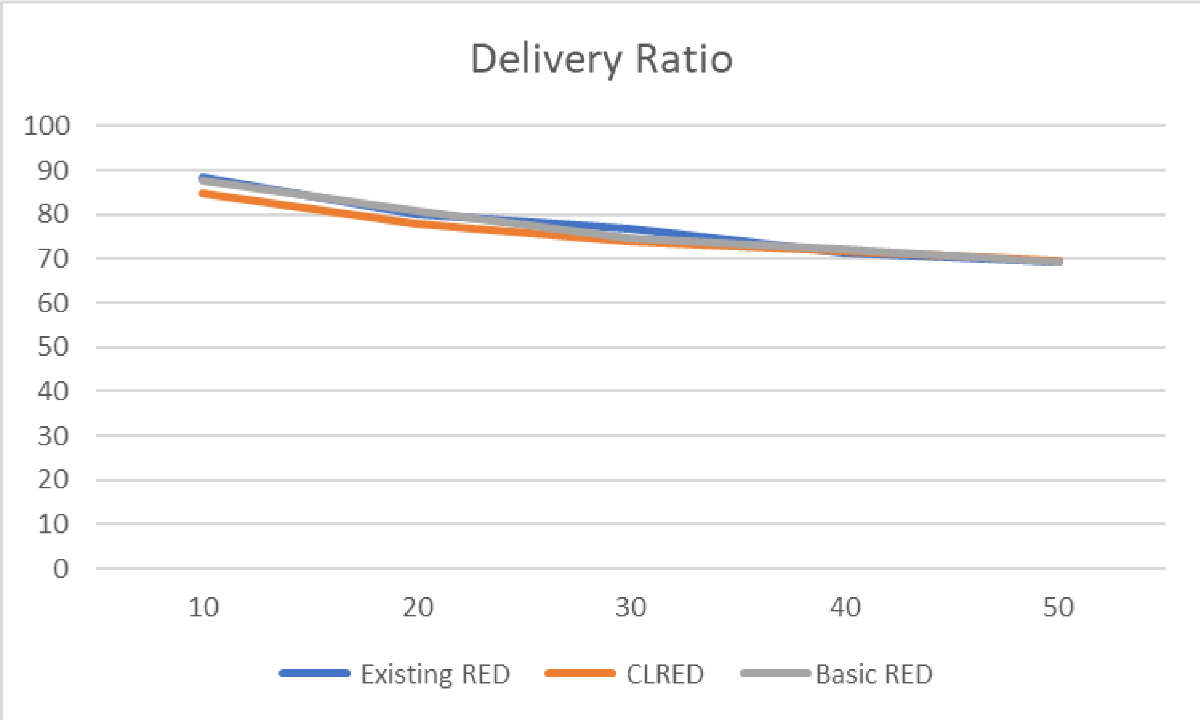

Fig. Delivery Ratio vs No. of Flows in Wired topology

Remarks: All three mechanisms almost similarly for delivery ratio vs no. of flows in Wired topology.

8. Drop Ratio vs No. of Flows in Wired topology

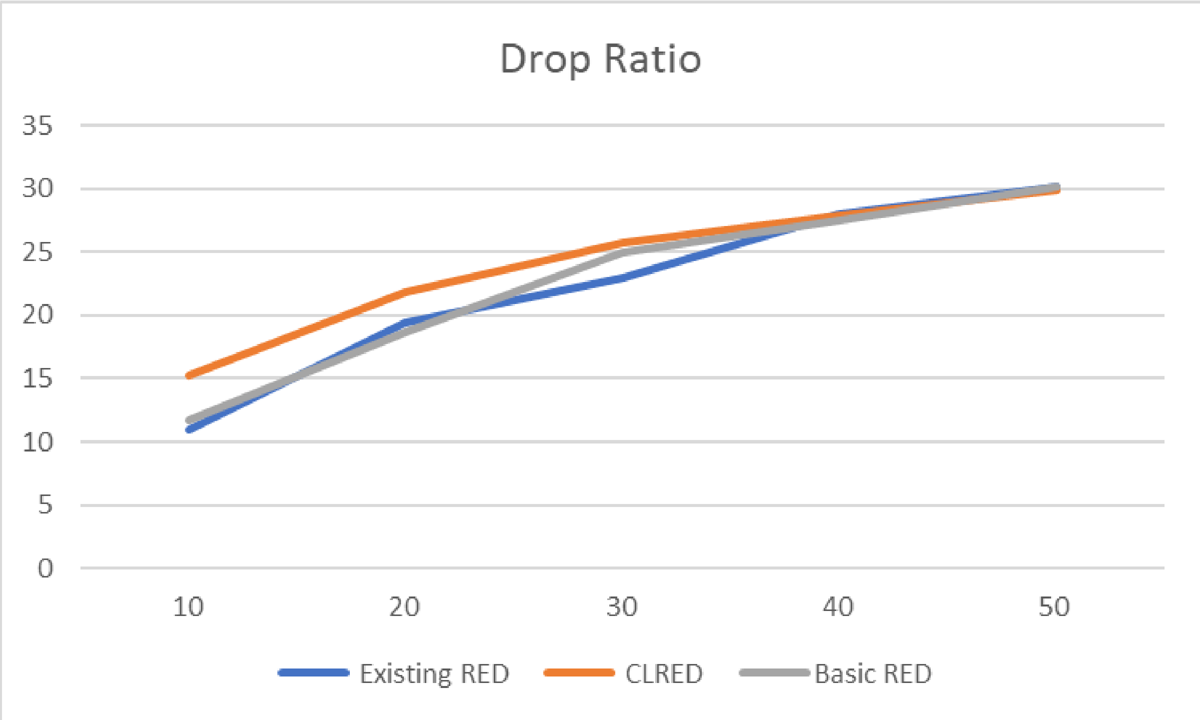

Fig. Drop Ratio vs No. of Flows in Wired topology

Remarks: All three mechanisms almost similarly for drop ratio vs no. of flows in Wired topology.

9. Throughput vs Packets per second in Wired topology

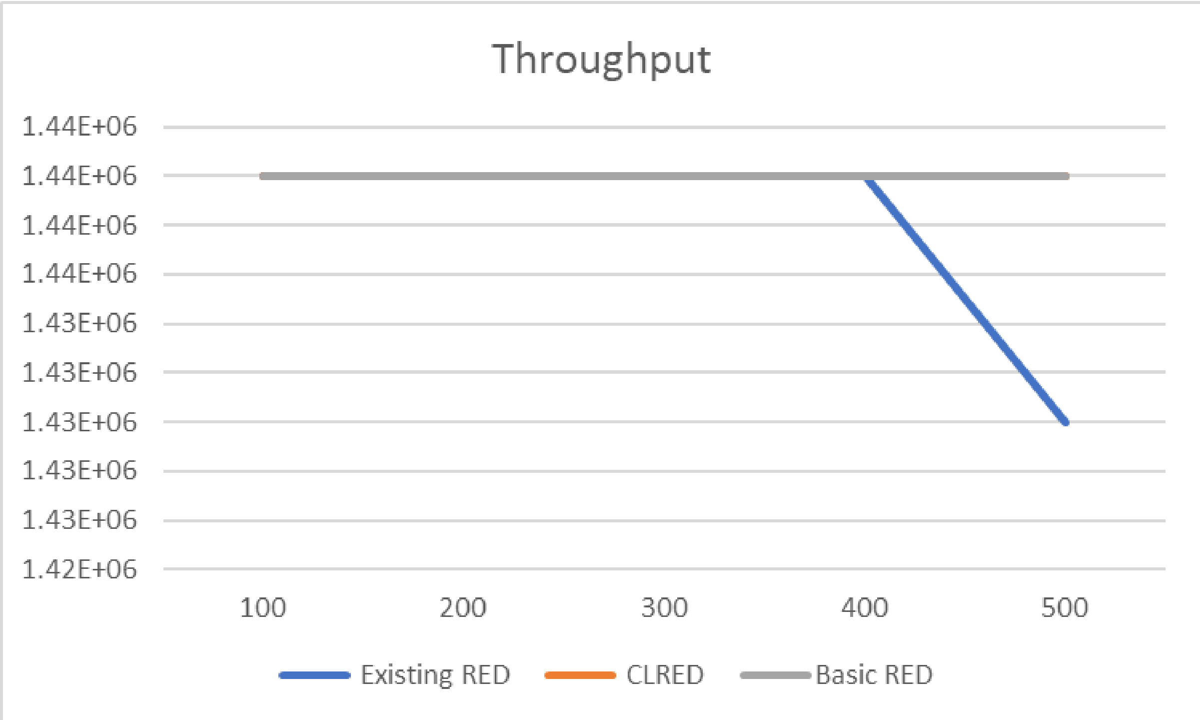

Fig. Throughput vs Packets per second in Wired topology

Remarks: All three mechanisms almost similarly for throughput vs packets per second in Wired topology.

10. Average Delay vs Packets per second in Wired topology

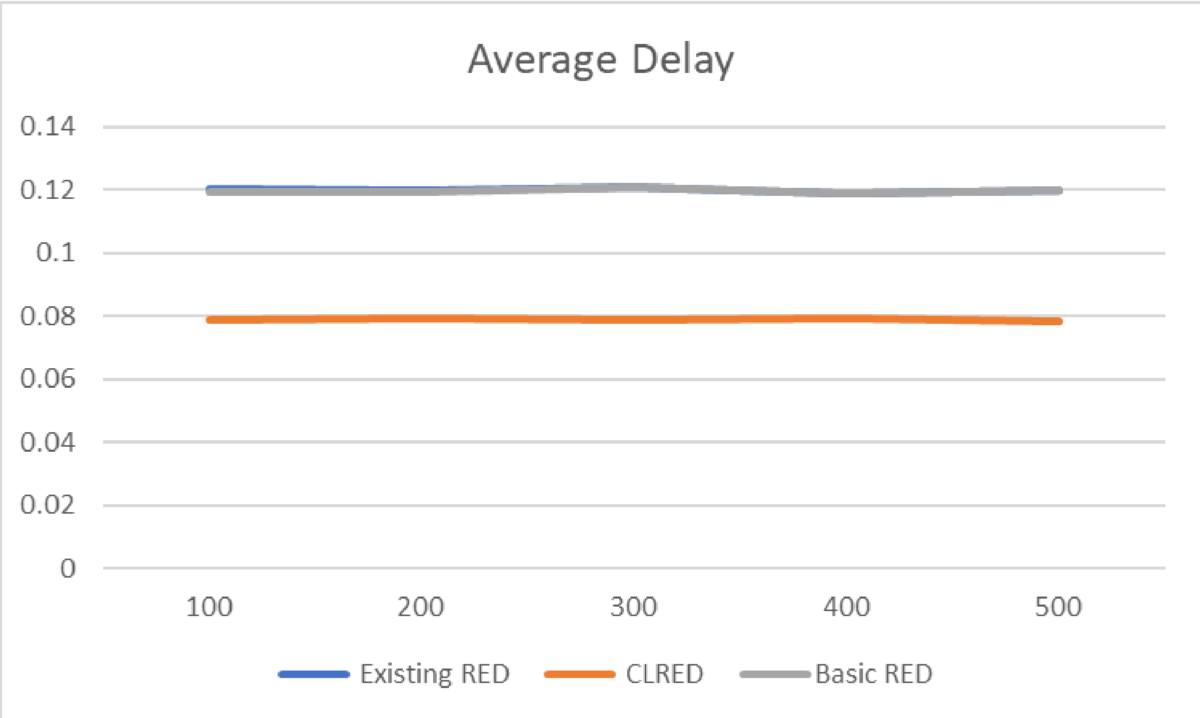

Fig. Delivery Ratio vs Packets per second in Wired topology

Remarks: CLRED performs better than other two mechanisms for the average delay vs packets per second in Wired topology. The average delay for the other mechanism is in the 0.11-0.12 range, whereas that of CLRED belongs to the 0.078-0.079 group.

11. Delivery Ratio vs Packets per second in Wired topology

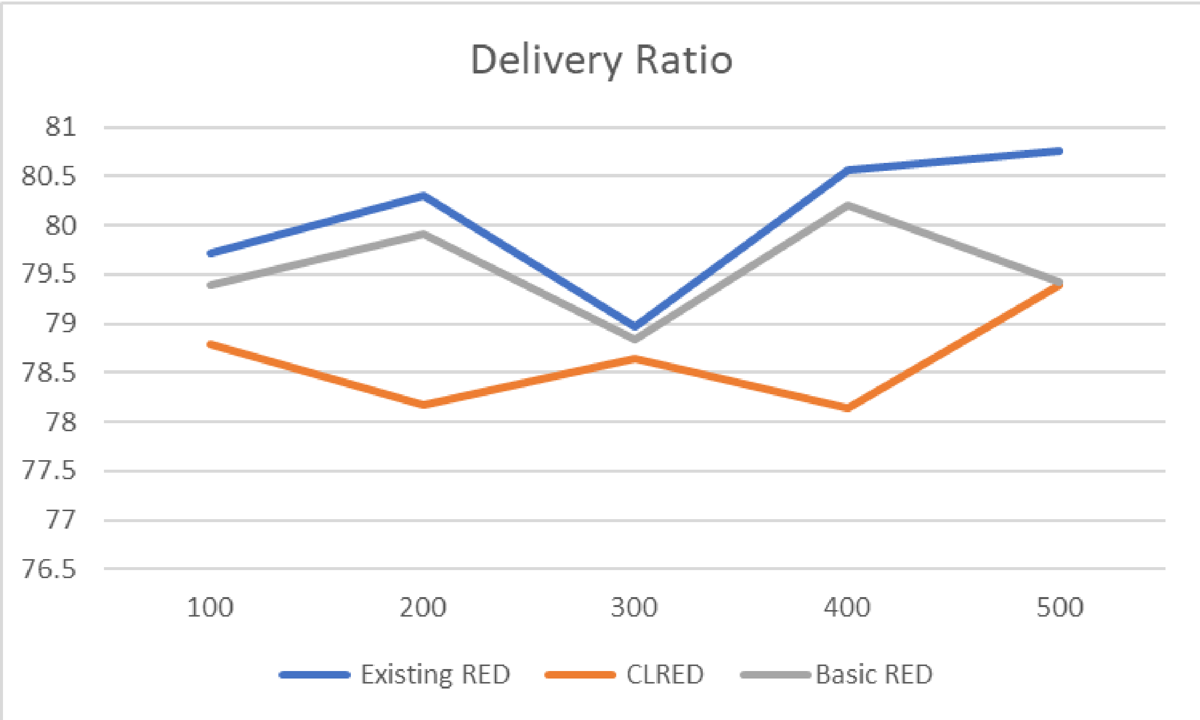

Fig. Delivery Ratio vs Packets per second in Wired topology

Remarks: Other two mechanisms perform slightly better than CLRED in case of delivery ratio vs packets per second in Wired topology.

12. Drop Ratio vs Packets per second in Wired topology

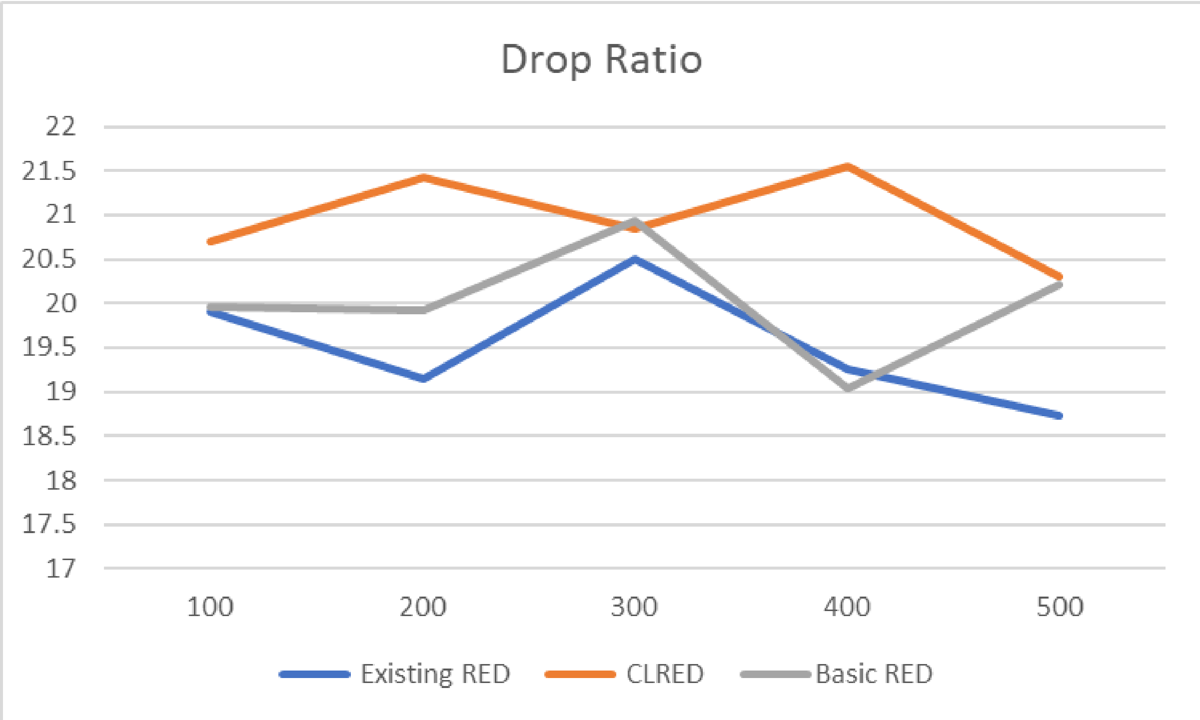

Fig. Drop Ratio vs Packets per second in Wired topology

Remarks: Other two mechanisms also perform slightly better than CLRED in case of drop ratio vs packets per second in Wired topology.

Wireless Topology

1. Throughput vs Area in Wireless topology

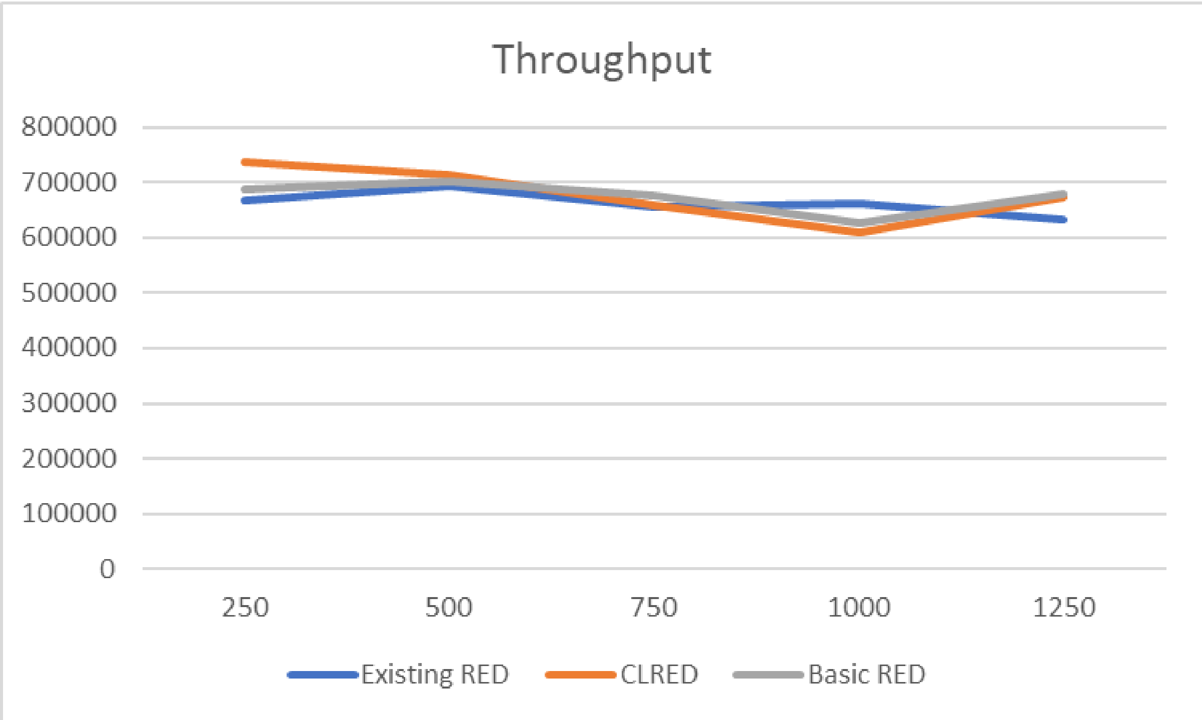

Fig. Throughput vs Area in Wireless topology

Remarks: CLRED performs better than the other two mechanisms in throughput vs area in Wireless topology.

2. Average Delay vs Area in Wireless topology

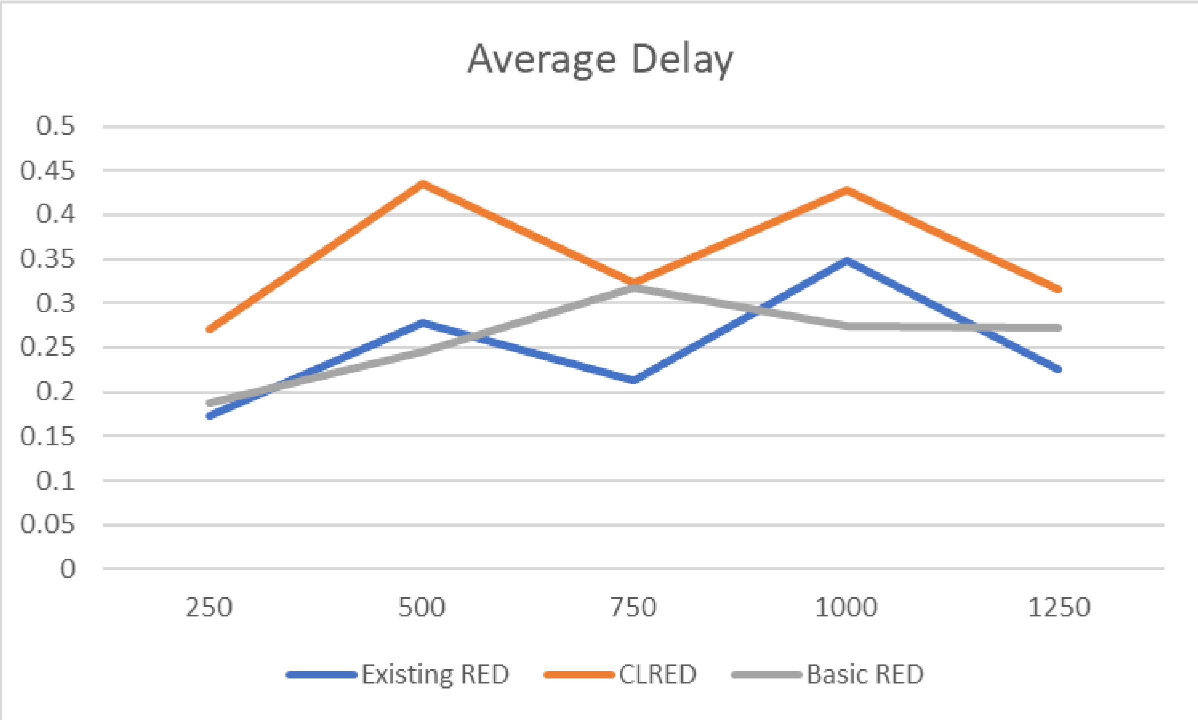

Fig. Average Delay vs Area in Wireless topology

Remarks: basic RED performs slightly better than the other two mechanisms for average delay vs area in Wireless topology.

3. Delivery Ratio vs Area in Wireless topology

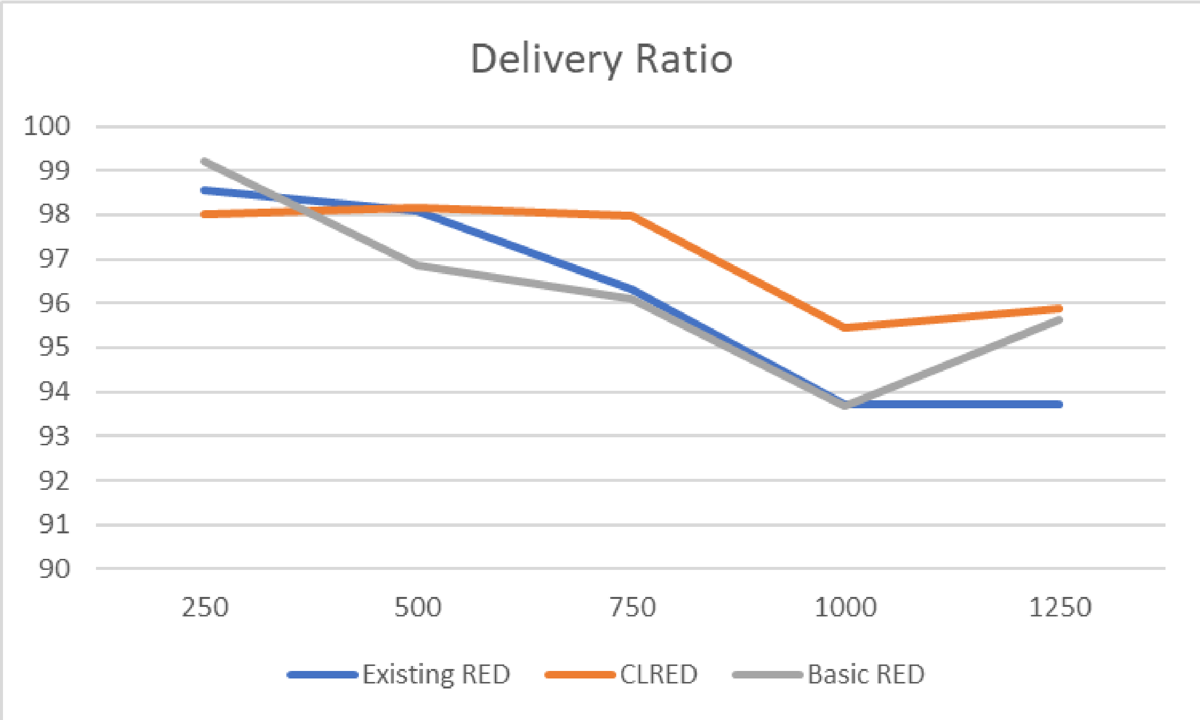

Fig. Delivery Ratio vs Area in Wireless topology

Remarks: As the area increases delivery ratio starts to decrease over all three schemes. However, for smaller areas, CLRED appears to be more stable than the other two.

4. Drop Ratio vs Area in Wireless topology

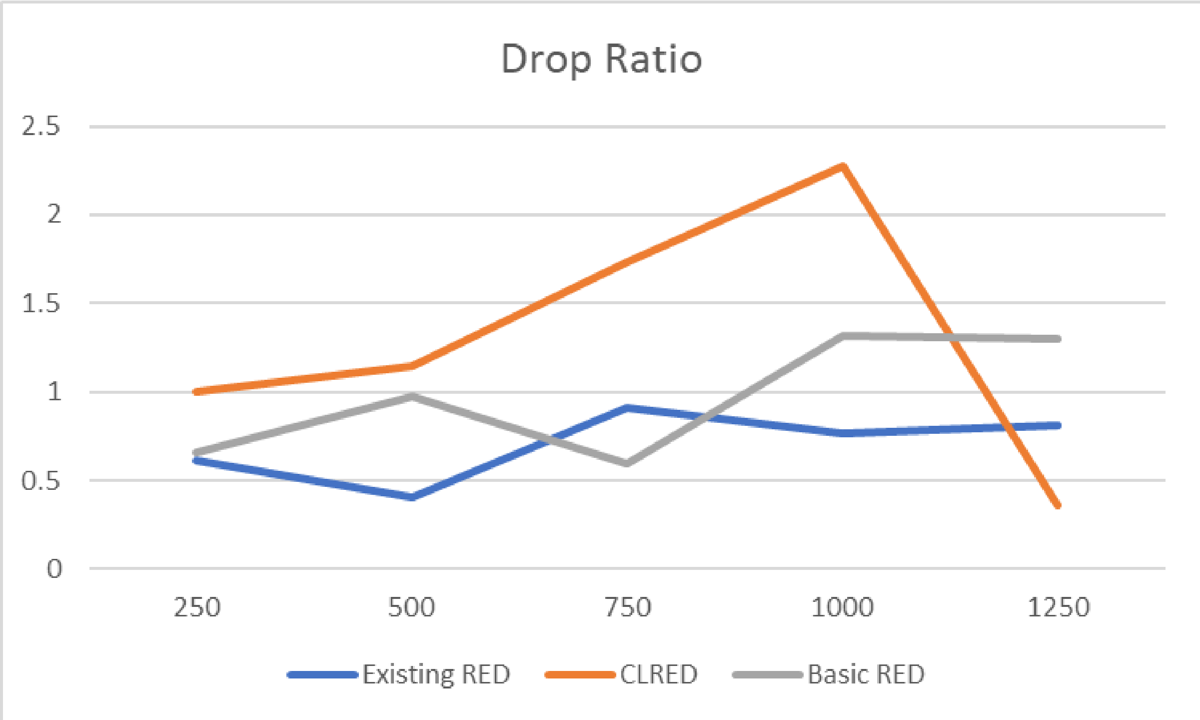

Fig. Drop Ratio vs Area in Wireless topology

Remarks: The existing RED in the VM performs slightly better than CLRED and basic RED for drop ratio vs area in Wireless topology. CLRED could not outperform the other two mechanisms here.

5. Energy Consumption vs Area in Wireless topology

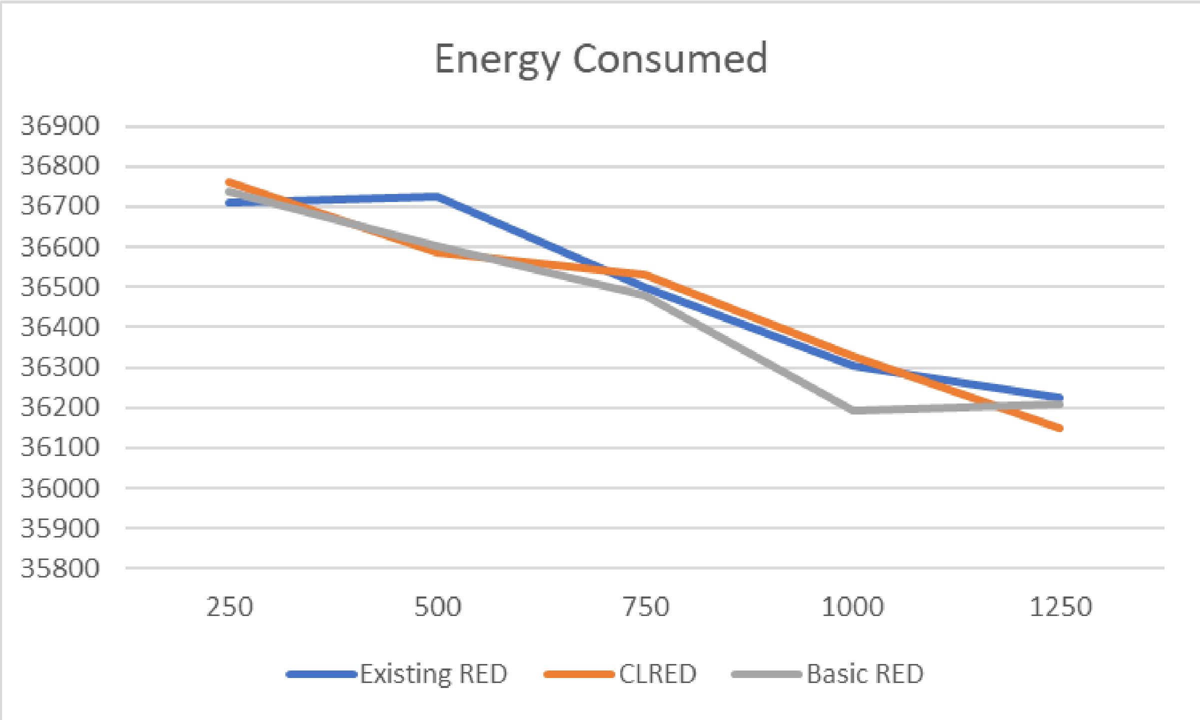

Fig. Energy Consumption vs Area in Wireless topology

Remarks: All three mechanisms consume energies of similar range for growing area size in Wireless topology.

6. Throughput vs No. of Nodes in Wireless topology

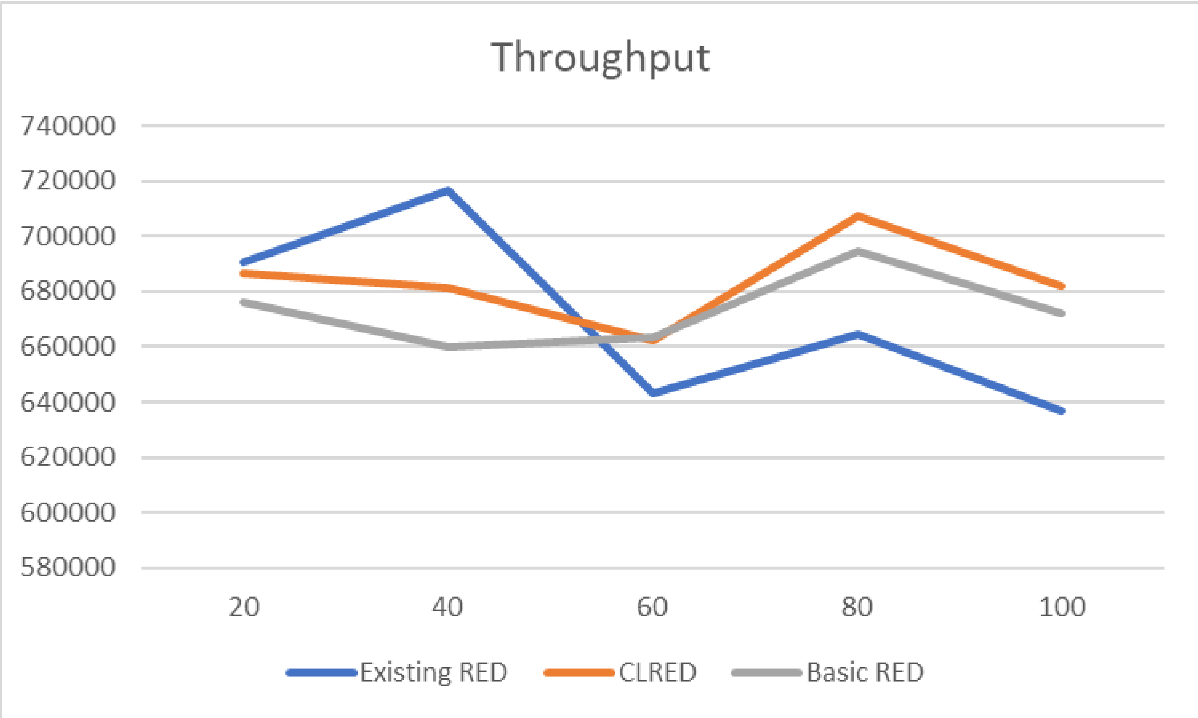

Fig. Throughput vs No. of Nodes in Wireless topology

Remarks: All three mechanisms almost similarly for throughput vs no. of nodes in Wireless topology.

7. Average Delay vs No. of Nodes in Wireless topology

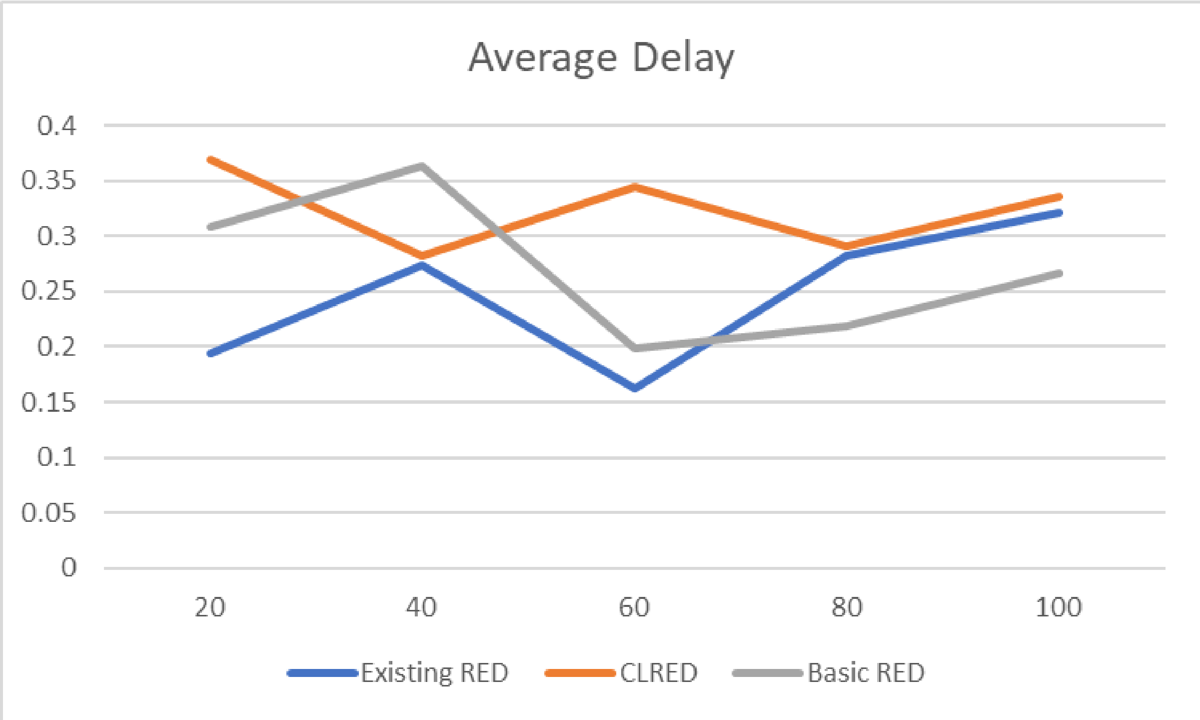

Fig. Average Delay vs No. of Nodes in Wireless topology

Remarks: All three mechanisms almost similarly for average delay vs no. of nodes in Wireless topology.

8. Delivery Ratio vs No. of Nodes in Wireless topology

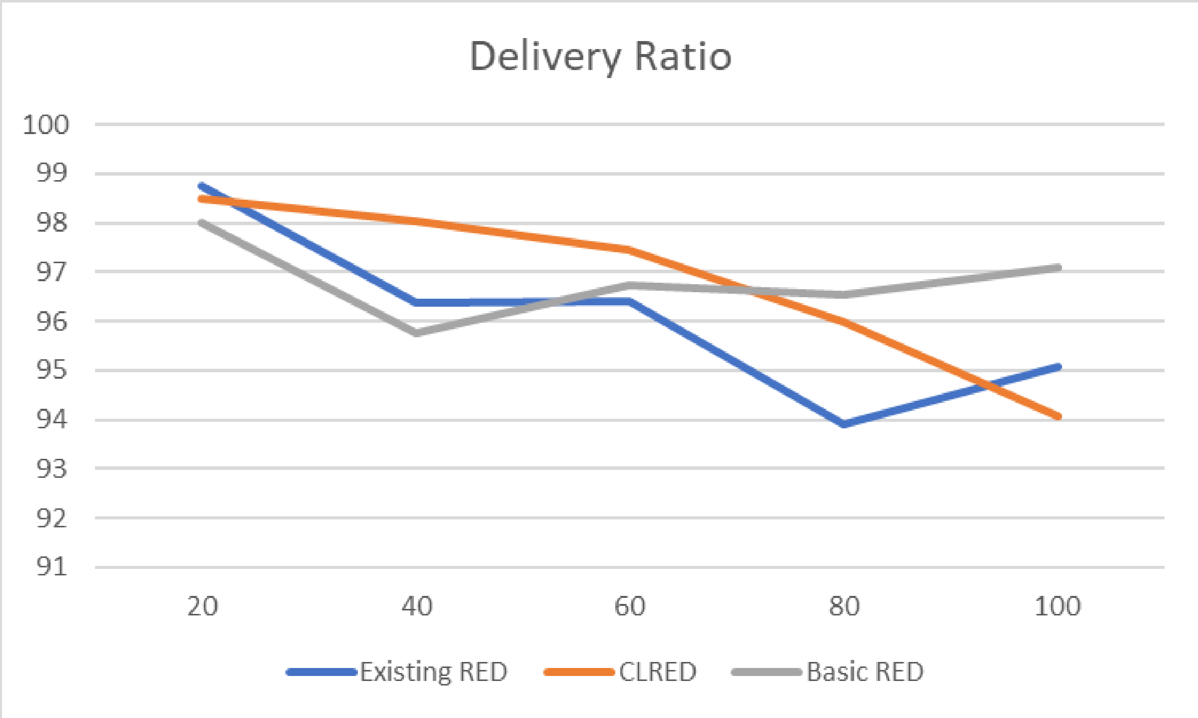

Fig. Delivery Ratio vs No. of Nodes in Wireless topology

Remarks: With increasing no. of nodes in Wireless topology, delivery ratio starts to decrease over all three schemes. However, the change of CLRED is more gradual and stable than the other two.

9. Drop Ratio vs No. of Nodes in Wireless topology

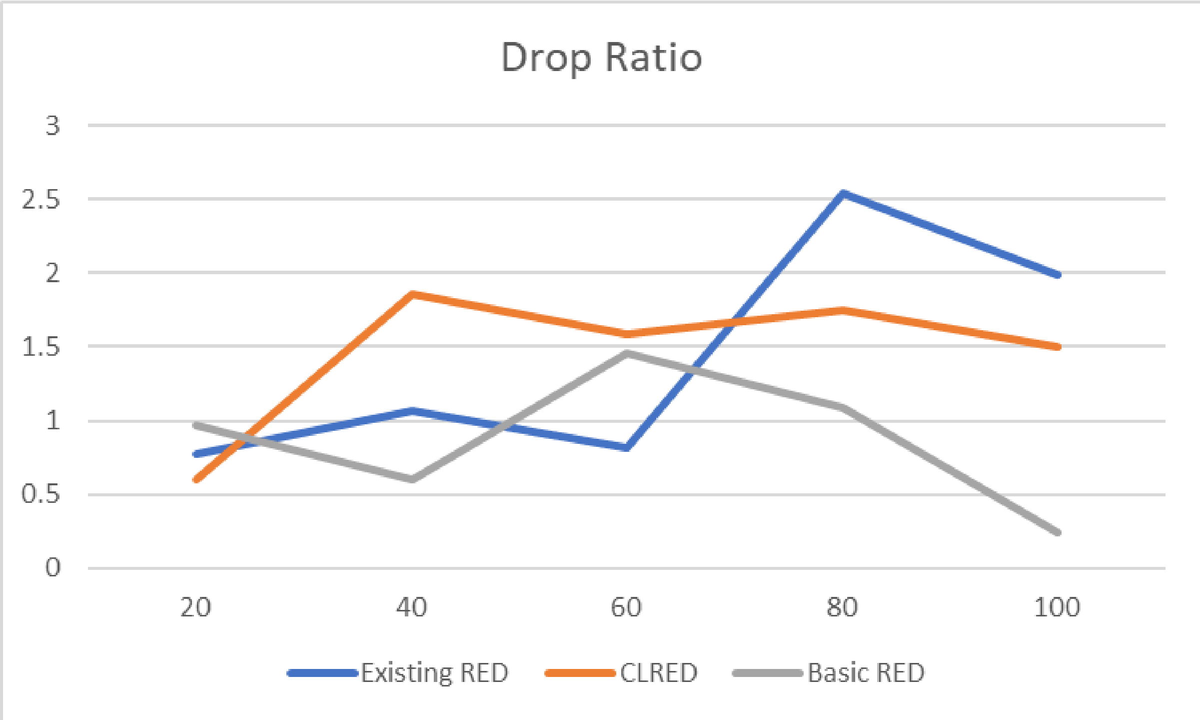

Fig. Drop Ratio vs No. of Nodes in Wireless topology

Remarks: For drop ratio vs no. of nodes in Wireless topology, basic RED performs the best, followed by CLRED.

10. Energy Consumption vs No. of Nodes in Wireless topology

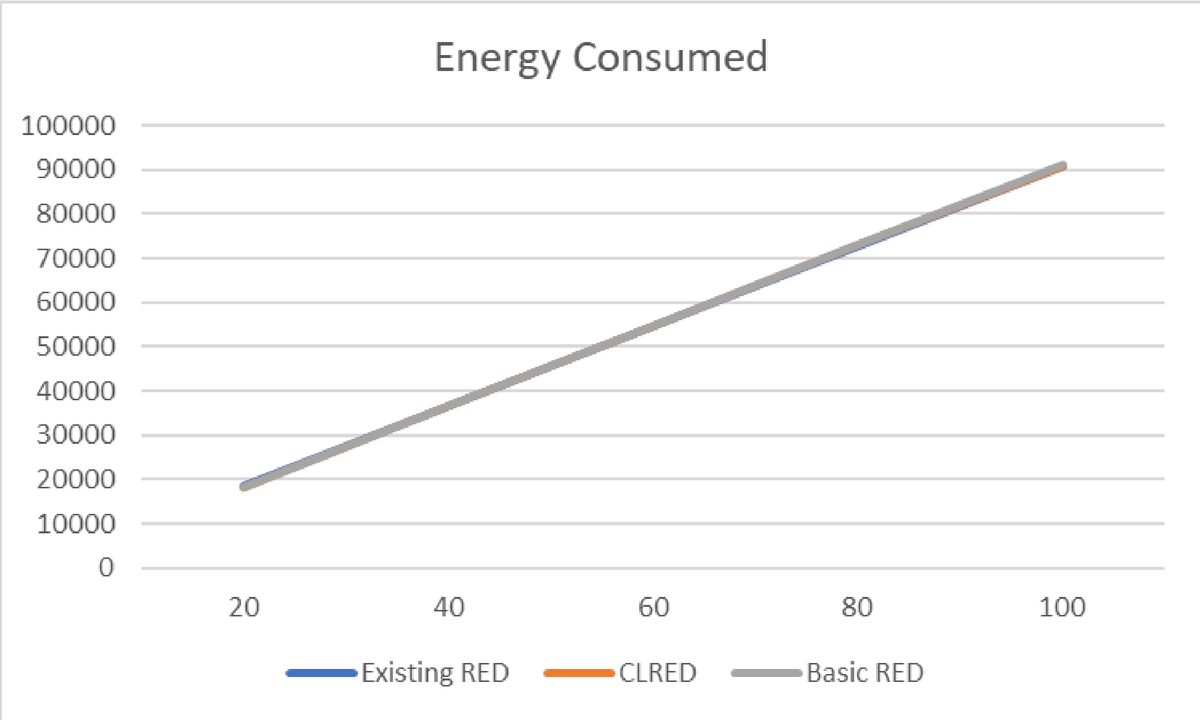

Fig. Energy Consumption vs No. of Nodes in Wireless topology

Remarks: Surprisingly, all three mechanisms perform almost exactly in a linear manner for energy consumption vs no. of nodes in Wireless topology.

11. Throughput vs No. of flows in Wireless topology

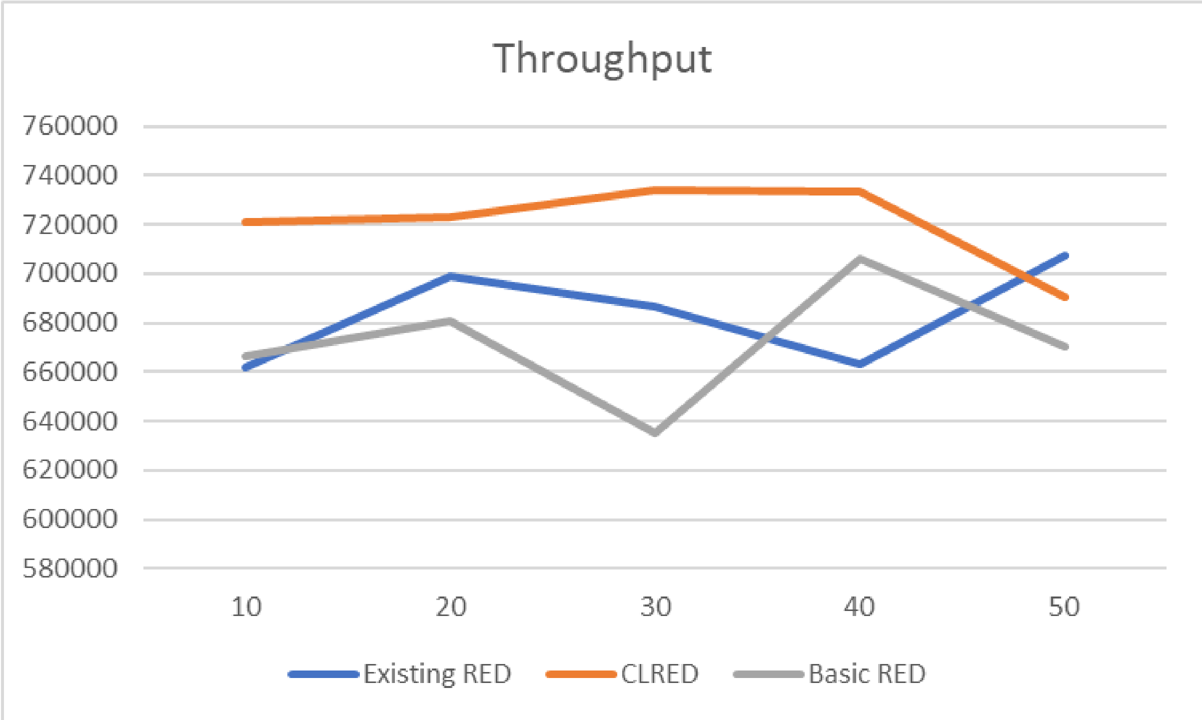

Fig. Throughput vs No. of Flows in Wireless topology

Remarks: All three mechanisms almost similarly for throughput vs no. of Flows in Wireless topology with CLRED slightly better than the other two.

12. Average Delay vs No. of flows in Wireless topology

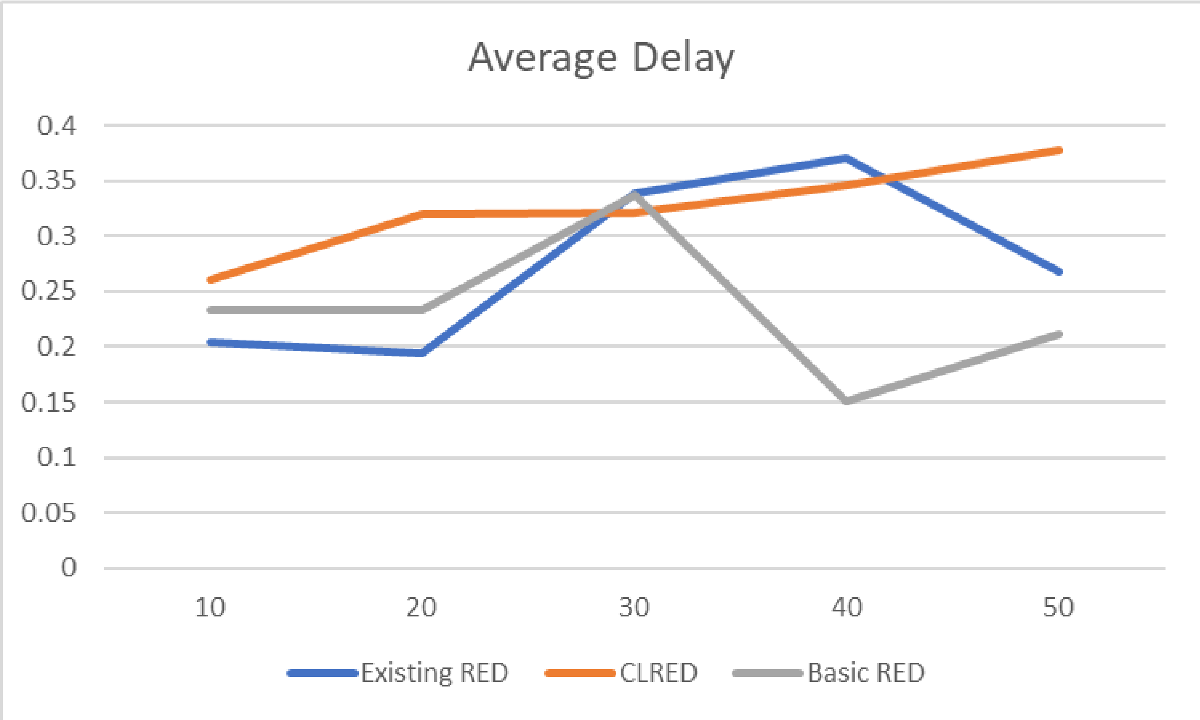

Fig. Average Delay vs No. of Flows in Wireless topology

Remarks: All three mechanisms almost similarly for average delay vs no. of flows in Wireless topology.

13. Delivery Ratio vs No. of flows in Wireless topology

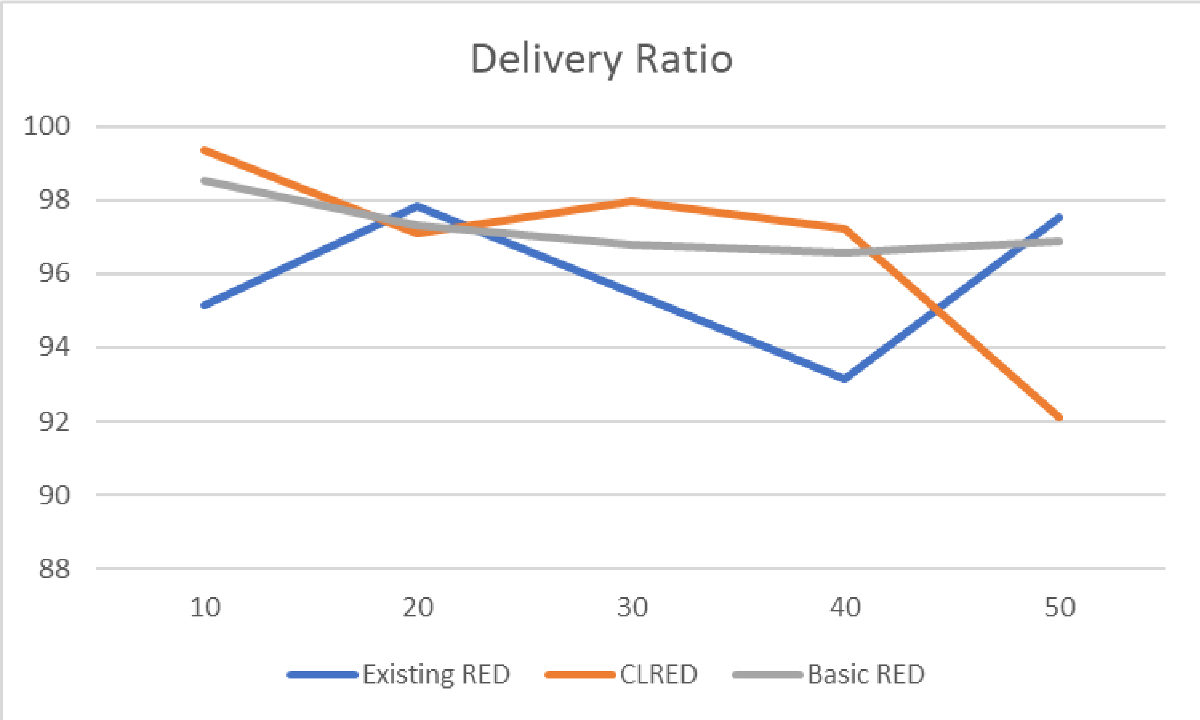

Fig. Delivery Ratio vs No. of Flows in Wireless topology

Remarks: The result of delivery ratio vs no. of flows in Wireless topology is almost similar to that of delivery ratio vs no. of nodes. With increasing no. of flows, performance declines, but the change of CLRED is slightly better than the other two.

14. Drop Ratio vs No. of flows in Wireless topology

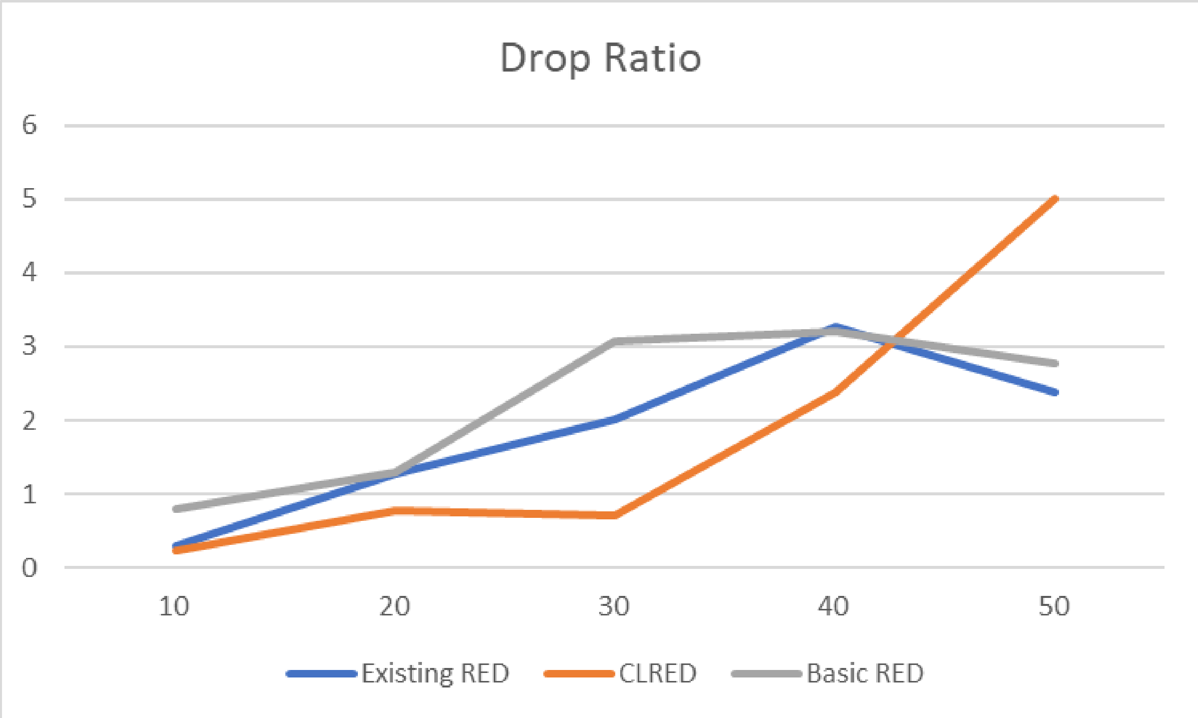

Fig. Drop Ratio vs No. of Flows in Wireless topology

Remarks: In Wireless topology, with growing no. of flows all three schemes start to drop more packets. Other two schemes perform better than CLRED in this metric.

15. Energy Consumption vs No. of flows in Wireless topology

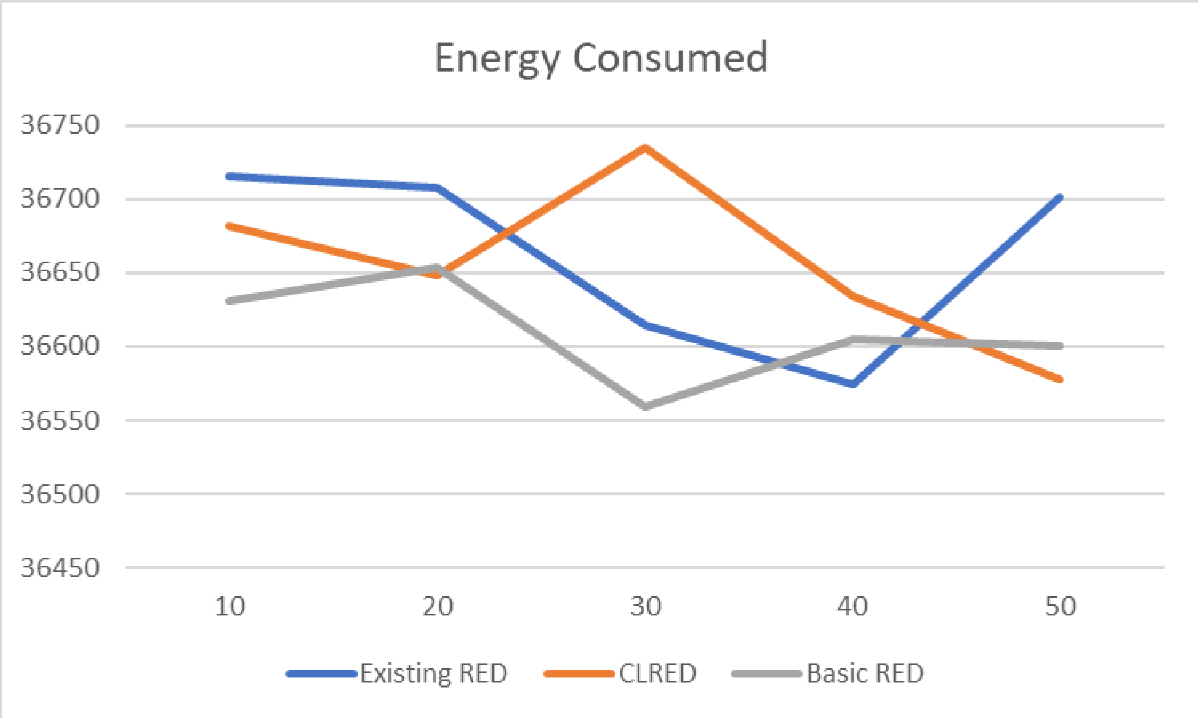

Fig. Energy Consumption vs No. of Flows in Wireless topology

Remarks: All three mechanisms consume energies of similar range for growing no. of flows in Wireless topology.

16. Throughput vs Packets per second in Wireless topology

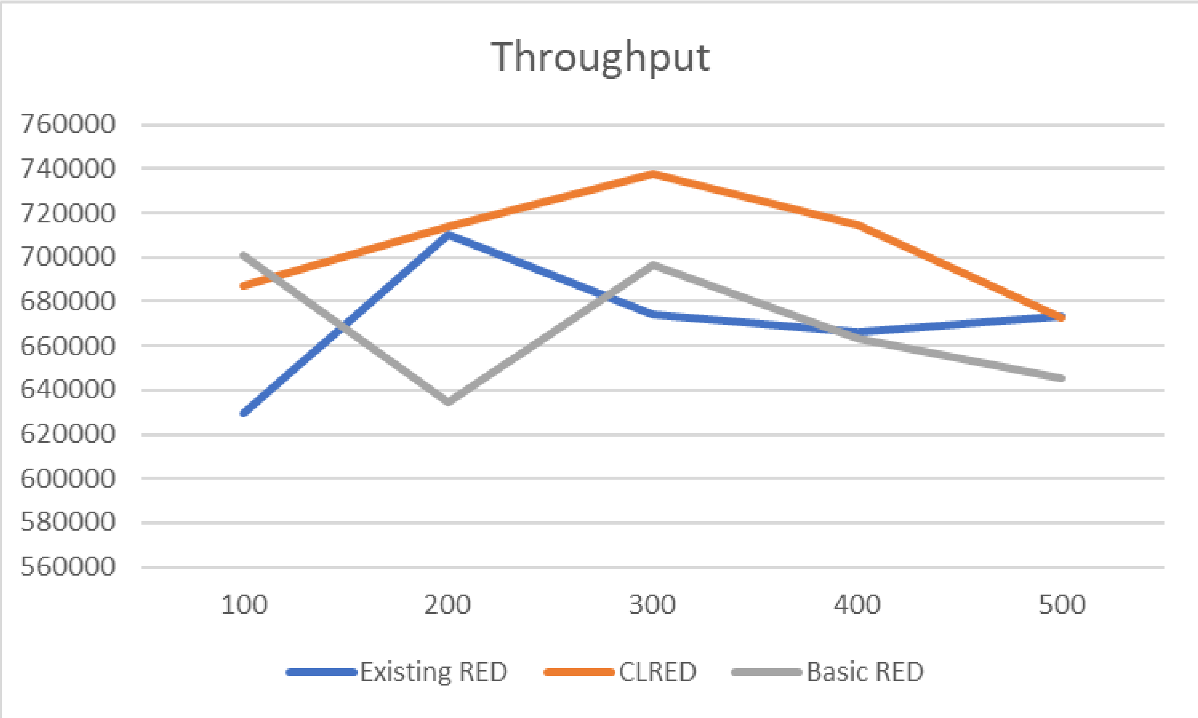

Fig. Throughput vs Packets per second in Wireless topology

Remarks: All three mechanisms almost similarly for throughput vs no. of packets in Wireless topology.

17. Average Delay vs Packets per second in Wireless topology

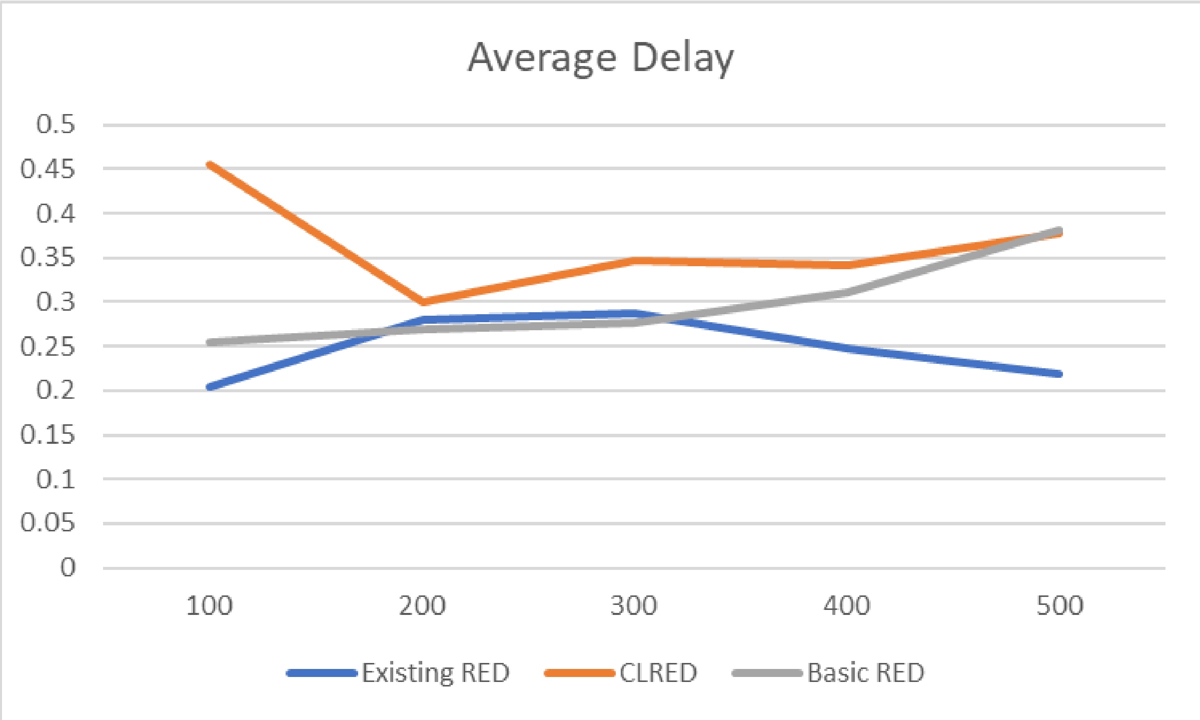

Fig. Average Delay vs Packets per second in Wireless topology

Remarks: Other two mechanisms perform slightly better in case of average delay vs packets per second in Wireless topology.

18. Delivery Ratio vs Packets per second in Wireless topology

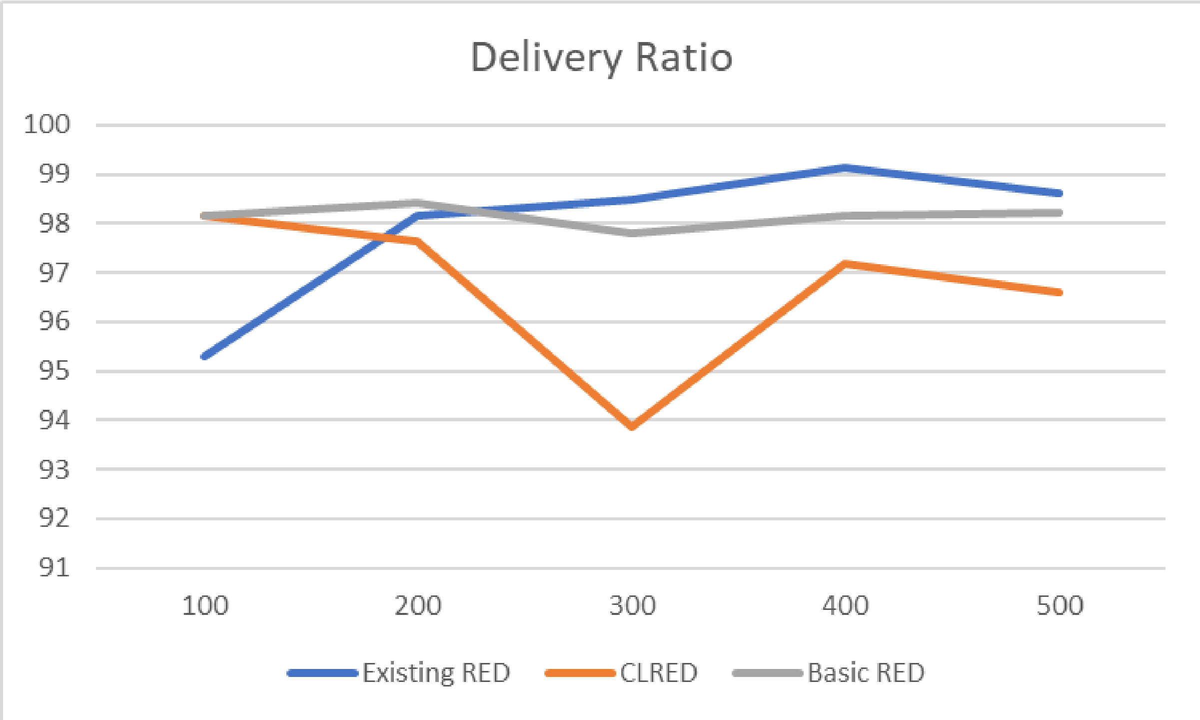

Fig. Delivery Ratio vs Packets per second in Wireless topology

Remarks: The existing RED in the VM performs slightly better than CLRED and basic RED for delivery ratio vs packets per seconds in Wireless topology. The former shows a bit of a stable result.

19. Drop Ratio vs Packets per second in Wireless topology

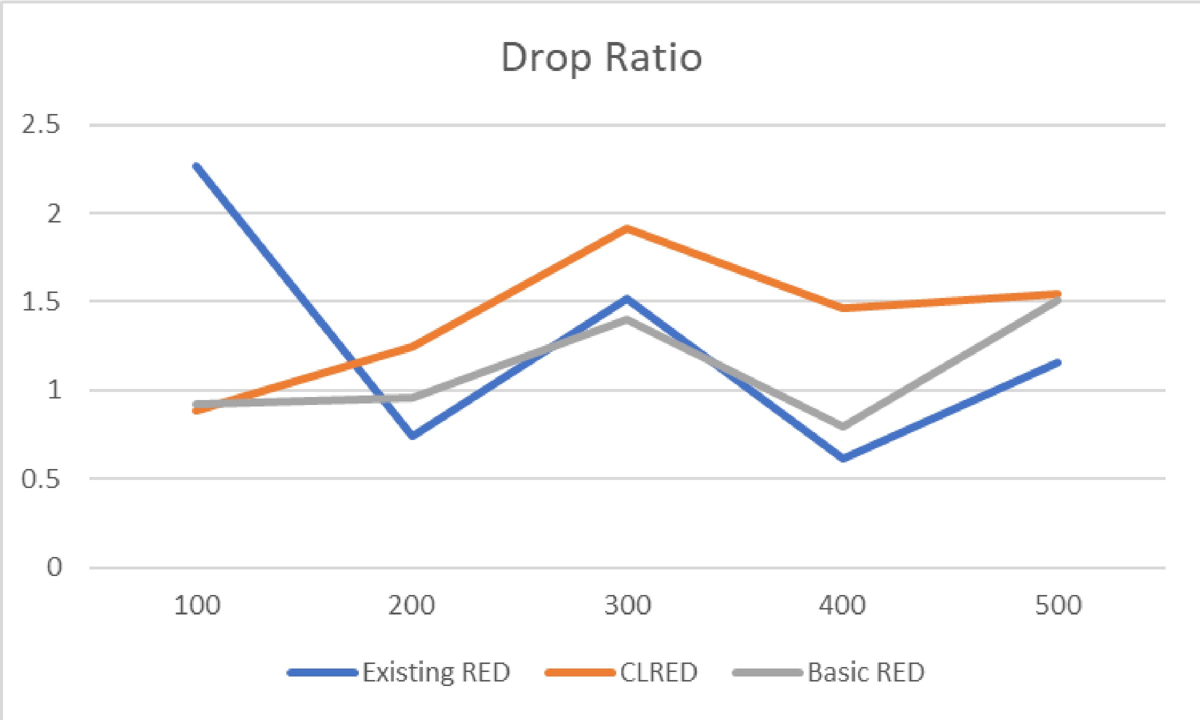

Fig. Drop Ratio vs Packets per second in Wireless topology

Remarks: All three mechanisms perform almost similarly for drop ratio vs no. of flows in Wireless topology.

20. Energy Consumption vs Packets per second in Wireless topology

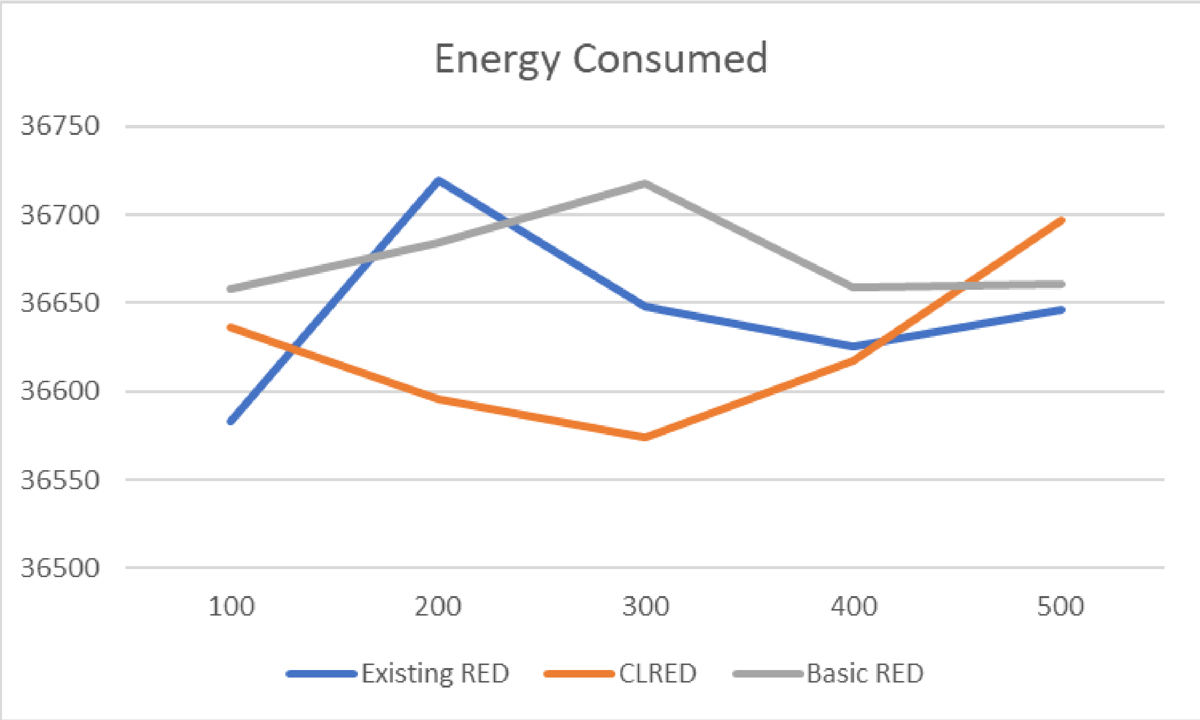

Fig. Energy Consumption vs Packets per second in Wireless topology

Remarks: Although the graphs for all three schemes are different for energy consumption vs packets per second in Wireless topology, they somewhat fall in the same range.

Summary

The performance of a RED algorithm depends solely on the drop probability function. With a better formulation, the performance of RED can be enhanced. CLRED was introduced with the same motivation to improve the drop probability function so that congestion can be implicitly better tackled by dropping packets early.

In the original paper where CLRED was introduced, the authors compared CLRED with basic RED in NS3. They used a network topology with a bottleneck node that connects all the nodes and a sink. In their *Wired* topology, their result showed, the average delay of CLRED is 75% of that of basic RED.

For this project, two network topologies were used: *Wired* and *Wireless*. The simulations were run in NS2. Three different RED algorithm was used:

i. The one that already existed in the VM supplied to us
ii. CLRED
iii. Basic RED

For the *Wired* topology, two bottlenecks were used, which made it similar to that of the paper. The results were also similar. The improved average delay by using CLRED was 60-70% of that obtained from basic RED. In other metrics such as throughput, delivery ratio and drop ratio, all three algorithms performed in the same range. And for the *Wireless* topology, a grid was used. Although CLRED performed neck to neck with the other two algorithms, in few cases, basic RED slightly outperformed CLRED. But to differentiate between the performance of CLRED and other two schemes in *Wireless* topology, one has to be very critical with minimal room for deflection.

CLRED offered very little improvement in a *Wireless* topology. But as the authors claimed, CLRED proved to be truly outperforming basic RED algorithms in a *Wired* topology where bottleneck nodes are present. So, in cases where the network is wired and bottlenecks are involved, CLRED can be a great alternative to existing RED algorithms for a better early detection of congestion.