Chapter 5

Model-2: AQM-RED-RPL

5.1 Introduction

Random Early Detection (RED) is also an efficient method to manage congestion that operates mostly on intermediary gateways. We have defined a modern active queue management framework aimed at growing the rate of packet loss in an easy and scalable way. We also have made a few improvements to the existing RED scheme's packet drop feature. The majority of the initial RED is now intact. We accomplish this by making many improvements and monitoring both the overall queue size and the instant queue size of the packet dropping feature. Simulations reveal that **Model-2: RED Active Queue Control for Packet Loss Reduction (AQM-RED-RPL)** achieves the maximum throughput and lowest packet drops relative to RED , Blue, REM, FRED, LDC and SRED. As this is completely compliant with RED, this solution can be quickly upgraded / replaced by current RED implementations.

There are some drawbacks of RED which affect its performance for effective congestion control in mobile ad-hoc networks. RED performance is sensitive to packet size. There are various mechanisms by which the performance of RED can be improved by reducing packet loss. Active Queue Management (AQM)is an important mechanism in RED to reduce packet loss so that its performance can be improved. This could be achieved by monitoring both the average queue size and the instant queue size packet dropping feature. In this **Model-2: Active Queue Management in RED to Reduce Packet Loss (AQM-RED-RPL)**, an attempt has been made to develop mew model using a modified RED algorithm.

We are recommending certain refinements to the standard RED framework, unlike the latest RED improvement projects. Rest part of the original RED remains same. We call this new scheme Active Queue Management (AQM) in RED AQM-RED-RPLto reduce packet loss. During light traffic, when the total size of the queue approaches the Maximum threshold (maximum), RED will drop all packets even if current queue size is either low, or empty. When charging is getting heavy and the current size of the queue soon approaches the wait limit – an indication that the duration of the wait can soon be exited monitor but the average size of the queue is not big enough to allow random drops; ERED permits more systematic falling of packets go back away from it.

5.2 Motivation and Objectives of Model-2: AQM-RED-RPL

The main motivation and objectives of designing this model is to improve the performance of original RED algorithm by reducing packet loss using new Active Queue Management (AQM) technique: RED to reduce packet loss. The weaknesses of the RED algorithms are mentioned below:

- If congestion is too high, it is impossible for the gateway to controlling the average queue size by numbering a fraction atmost packet max_p so the average size of the queue will surpass. The max_{th} and the gateway mark each packet before each packet is picked.
- The received packets is discarded with probability P_a even when the current queue size is empty. It happens when the average queue size lies between the minimum and maximum values.
- As communications minimize their sending percentage, show window instantly decreases but the average queue size would decline slowly. If the average queue length is higher then entering data will fall with greater probability in no congestion condition.
- If congestion becomes instantly high, then queue size will be increased immediately and the limits of queue will be raised and exceeded but no packets will be randomly dropped because the average size of the queue is less than $minth$.
- RED efficiency is dependent on the number of competitors participating in flows / sources. When the load is high, the RED output is degraded.
- Wild queue fluctuation is detected with RED when the traffic load is changed.
- RED achievement is conscious to the size of the packet.
- RED output is incredibly susceptible to the configurations of its variables.

5.3 Proposed Scheme: Model-2 AQM-RED-RPL

Queue based algorithms such as RED conducted relatively good though these are difficult to configure. Load based algorithms such as REM decreases wait in queuing, but it has poor site efficiency where traffic is used without ECN. Using both loading and queuing delay congestion indicators reduce response time and delay; But sometimes it has an adverse effect on the results when data gramme protocol (UDP) is used for both single and multiple-bottleneck situations. In addition, the LDC maintains the actual queuing wait.

The main feature of this Model - 2 AQM-RED-RPLare mentioned below:

- To achieve better results and overcome the drawback of the previous model, Model-2 (AQM-RED-RPL) has been proposed with some modification without modifying the Queue Weight parameter in the initial RED algorithm.
- The MinimumThreshold and MaximumThreshold parameters are modified as shown below:
  MaximumThreshold = 2 $\times$ MaximumThreshold
  MinimumThreshold = (2 $\times$ MaximumThreshold + 3 $\times$ MinimumThreshold) / 5 + MinimumThreshold
- The current queue size is controlled together with average queue size.
- if (MinimumThreshold $<$ AVG $<$ MaximumThreshold and Queue-Length $>$ MinimumThreshold)
  Each arriving packet is dropped with probability P_a

5.3.1 Algorithm of Model-2: AQM-RED-RPL

This algorithm is based on active queue management and it is able to control the congestion:

Here, new average queue size AVG is calculated when queue is not empty as $AVG = (1-W(q)AVG+Wqq)$ otherwise, $AVG = (1-W(q)m \times AVG)$ where $m = f(time-qtime)$. Then depending on the AVG value, we set the probability value $Pb = MAXp(AVG - MinimumThreshold)/(MaximumThreshold - MinimumThreshold)$ and $Pa = Pb/(1 - Pb \times count)$. Here we consider three new variable M1, M2 and M3 as
$M1 = (MaximumThreshold - MinimumThreshold)/2 + MinimumThreshold$,
$M2 = (MaximumThreshold + MinimumThreshold)/2 + MaximumThreshold$ and

$M3 = 2 \times (MaximumThreshold + MinimumThreshold)/3 + MinimumThreshold$ to control the size of queue in different situation. The algorithm describe below of the proposed model AQM-RED-RPL.

5.4 Results and Comparison of Model-2 : AQM-RED-RPL

In this section, the results of the simulated text have been presented to make a comparison of performance of Model-II:AQM-RED-RPL- based routing algorithm with that of existing RED-based algorithm. It has been seen that the proposed Model-II is very effective with respect to the parameters:throughput, goodput, end to end delay and packet delivery ratio.

Simulation has been performed with 100 nodes and the proposed path selection take random. The Table 5.1 shows the performance measurement of the proposed AQM-RED-RPLscheme with various number of nodes from 2 to 100. Here, in this experiments, end to end delay varies from 333 to 158 with the variation of number of nodes. When number of node increases in the network then end to end delay decreases. Packet delivery ratio also decreases, but throughput and goodput increases depending on the increase of input node.

End to End Delay: The ratio of packet received time to packet send time is termed as end to end delay. That end to end delay is supposed to be low in order to provide better performance. The Table 5.2 and graph Fig. 5.1 show the performance of routing end to end delay in continuous traffic pattern for ADWD-RED-IP, and AQM-RED-RPL and RED.

In the proposed AQM-RED-RPL, due to active queue management the packet moves smoothly with a little bit loss. Here, congestion can be avoidable and packet can be delivered within time while increasing mode. Thus, the proposed AQM-RED-RPL performed better with low end to end delay when compared with existing scheme.

In this approach, congestion can be avoided and packet can be delivered within time while increasing number of nodes. Thus, the proposed AQM-RED-RPL, performance is better com-

Algorithm 5.1: Algorithm of AQM-RED-RPL

input : Initialize the nodes.

output: Congestion will be control with high throughput and low packet drop ratio and end to end delay.

```
Algorithm Transform():
    Let AVG=0, count=-1;
    for (each packet arrival) do
        // new average queue size is calculated
        if (queue is not empty) then
            AVG=(1-W(q)AVG + Wqq);
        else
            m = f(time - q_time);
            AVG = (1-W(q)m × AVG);
        end
    end
    if ((MinimumThreshold <= AVG < MaximumThreshold)) and (queue_length>=MinimumThreshold) then
        count++;
        Pb = MAXp(AVG- MinimumThreshold)/(MaximumThreshold - MinimumThreshold);
        Pa = Pb/(1 - Pb × count);
        // mark the arriving;
        count=0;
    else if (queue-length > 1.75 × MaximumThreshold ) then
        Pb = MAXp;
        Pa = Pb / (1- count × Pb);
        // mark the arriving packet;
        count = 0;
    else if (AVG >= MaximumThreshold) then
        count = 0;
    else
        count = -1;
        // when queue becomes empty;

        q_time = time;
    end
    for (each packet departure) do
        if (AVG>M1) and (queue_length < M2) then
            AVG = M3;
        end
        Here, M1 = (MaximumThreshold - MinimumThreshold)/2 + MinimumThreshold;
        M2 = (MaximumThreshold + MinimumThreshold)/2 + MaximumThreshold;
        M3 = 2 × (MaximumThreshold + MinimumThreshold)/3 + MinimumThreshold;
    end
```

pared with existing RED scheme.

Packet Delivery Ratio: Packet delivery ratio is outlined as the ratio between total numbers

Table 5.1: *Experimental results of AQM-RED-RPL*

Nodes	End-to-End Delay	Packet Delivery Ratio	Throughput	Goodput
2	333.235	90.32	711.35	345.32
10	301.539	87.32	716.35	365.76
25	204.326	93.24	731.24	370.13
50	177.328	90.21	750.36	380.29
75	175.214	90.48	748.32	385.64
100	158.325	88.36	748.79	390.65

Table 5.2: *End-to-End Delay of RED, ADWD-RED-IP and AQM-RED-RPL*

Nodes	End-to-End Delay (RED)	ADWD-RED-IP	AQM-RED-RPL
2	335.446	340.235	333.235
10	304.878	312.674	301.539
25	206.093	208.443	204.326
50	179.589	184.385	177.328
75	177.267	180.438	175.214
100	161.335	163.275	158.325

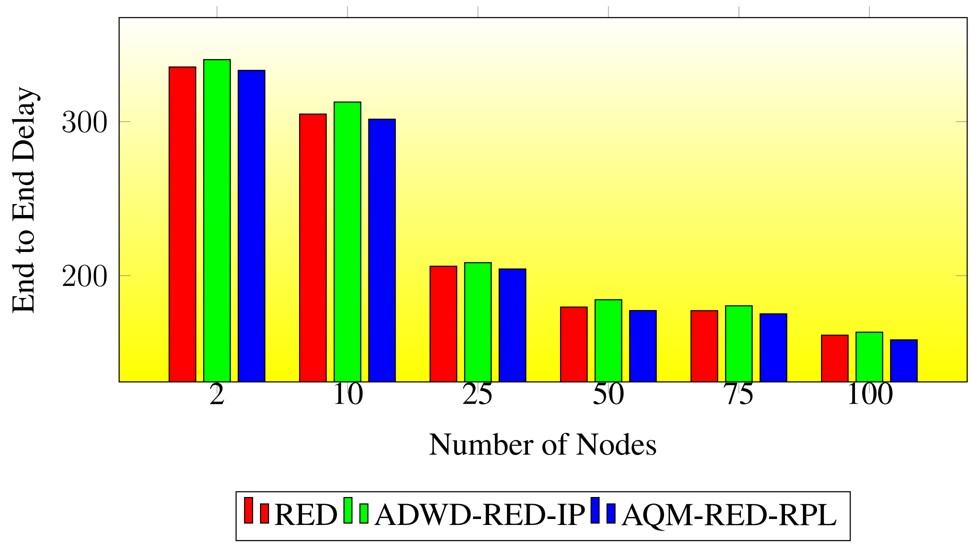

Figure 5.1: *Comparison of proposed schemes with respect to End to End Delay*

of packet send to the total number of packets received. The results are presented in Table 5.3 and the corresponding graphical representation is depicted in Fig. 5.2. The Fig. 5.2 represents the routing packet delivery ratio for existing RED, ADWD-RED-IP and AQM-RED-RPL algorithm with respect to the number of nodes. Due to active queue management of AQM-RED-RPL, it is possible to receive more packet without any loss, and the proposed AQM-RED-RPL algorithm achieves high packet delivery ratio than RED and gives better result.

Table 5.3: *Packet Delivery Ratio of RED, ADWD-RED-IP, and AQM-RED-RPL*

Nodes	RED	ADWD-RED-IP	AQM-RED-RPL
2	91.21	89.67	90.32
10	86.86	84.43	87.32
25	92.77	87.56	93.24
50	91.64	89.29	90.21
75	90.22	88.97	90.48
100	88.63	87.35	88.36

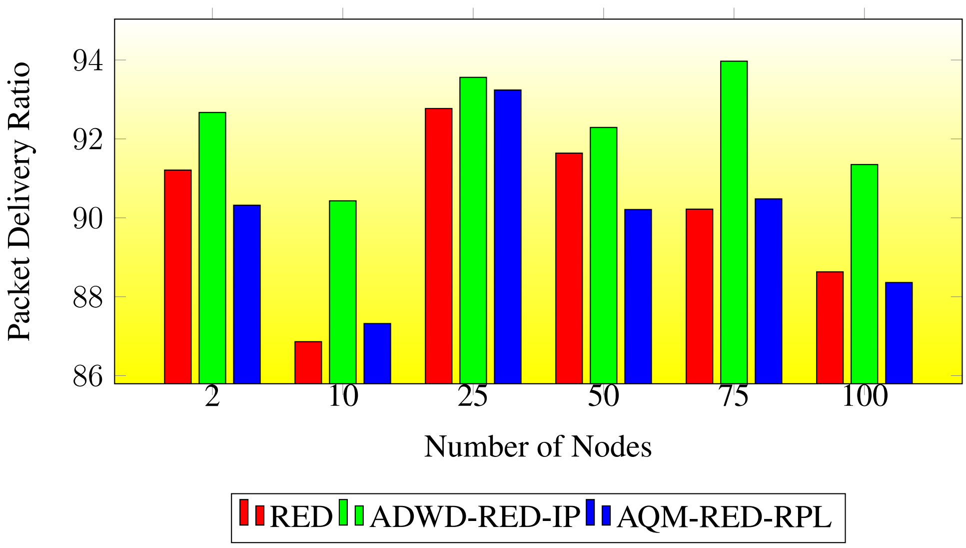

Figure 5.2: *Comparison of proposed schemes with respect to packet delivery ratio*

Throughput: Throughput is one of the important parameter for evaluating the performance of wireless Ad-Hoc network. The throughput is calculated based on number of bits transmitted per second. In order to provide better performance of the network, the system throughput must be high. The simulated result are shown in Table 5.4 and corresponding graph (Fig. 5.3) displayed the performance comparison for RED, ADWD-RED-IP and AQM-RED-RPL. It is analysed from the graph that, the throughput for the AQM-RED-RPL is gradually increasing more compared to the existing RED. Therefore, the proposed AQM-RED-RPL gives better throughput without loss.

Goodput: In the MANET, goodput is the number of useful information delivered by the

Table 5.4: *Throughput of RED, ADWD-RED-IP, and AQM-RED-RPL*

Nodes	RED	AQM-RED-RPL	AQM-RED-RPL
2	709.48	710.23	711.35
10	712.79	715.54	716.35
25	726.55	728.25	731.24
50	748.08	750.67	750.36
75	746.54	752.45	748.32
100	745.26	753.45	748.79

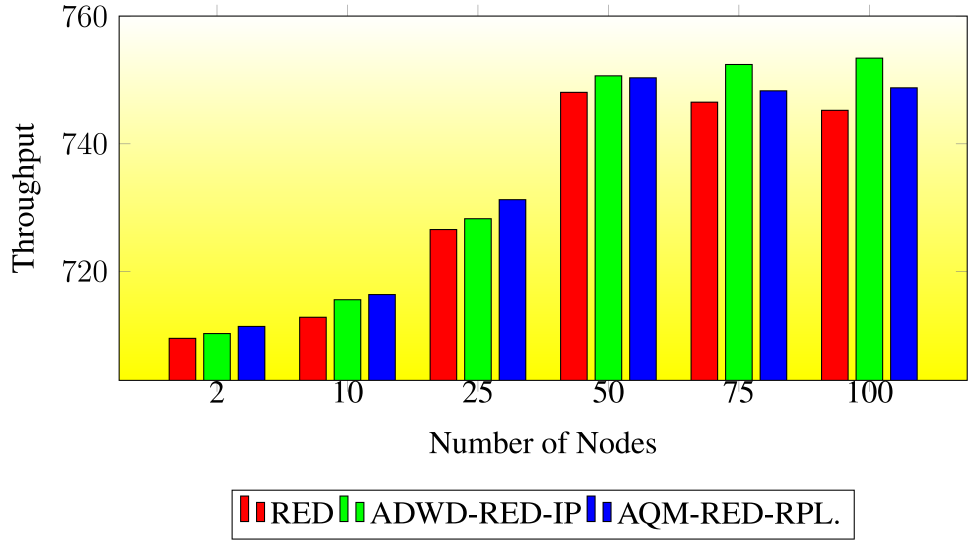

Figure 5.3: *Comparison of proposed schemes with respect to Throughput*

network to a certain node per unit of time. The goodput is always lower than the throughput due to overhead and lost or dropped packet for congestion. Table 5.5 shows the goodput comparison among RED, ADWD-RED-IP and AQM-RED-RPL. In the proposed AQM-RED-RPL, the goodput is better than RED based algorithms because the packet drop function has been changed which has been used for the active queue management. The corresponding graph is presented in Fig. 5.4. The explanation why ADWD-RED-IP and AQM-RED-RPL have a lower latency and jitter than other algorithms, because it is appropriate to forward or drop a packet that enters the router buffer without waiting in the router buffer anymore. The latency and jitter values of AQM-RED-RPL are smaller than those of the SRED, REM, BLUE and LDC algorithms. For real-time applications such as UDP in intermediate routers, the Latency and Jitter parameter values are lower enough to use AQM-RED-RPL.

Table 5.5: *Goodput of RED, ADWD-RED-IP, and AQM-RED-RPL*

Nodes	RED	ADWD-RED-IP	AQM-RED-RPL
2	337.85	388.76	345.32
10	355.04	408.32	365.76
25	367.21	431.45	370.13
50	374.04	455.39	380.29
75	377.04	462.68	385.64
100	382.18	478.49	390.65

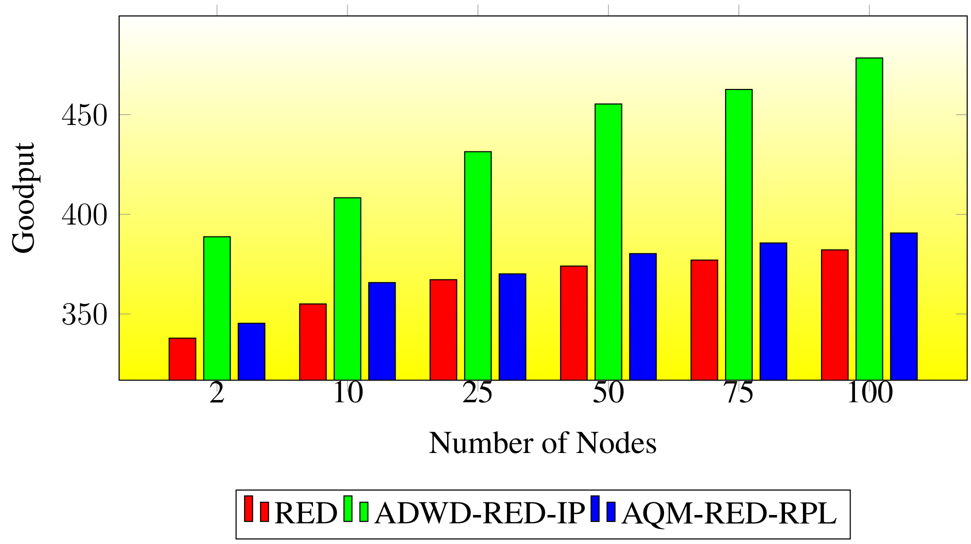

Figure 5.4: *Comparison of proposed schemes with respect to Goodput*

Table 5.6 represents the comparison of RED, ADWD-RED-IP and AQM-RED-RPL in terms of number of packet received, forwarded, dropped and loss rate for flows 20, 40, 60, ... 200 nodes. In this case packet loss rate is lower then RED due to active queue management with introducing MIN_q and MAX_q parameters. The corresponding graph shows that the AQM-RED-RPL improvess the performance of RED algorithm.

Table 5.6: *Analysis of the proposed scheme in terms of the number of packets received, forwarded, dropped, and packet loss rate for flows 20,40,60, ..., 200*

Algorithms	Packets received	Packets sent	Packets dropped	Packet Loss Rate	Throughput
RED	8487	8018.643	463.4286	0.0612	1.170581
ADWD-RED-IP	8357.231	8565.617	208.386	0.0243	1.124935
AQM-RED-RPL	8370.143	8155.071	206.3571	0.0235	1.192354

5.5 Summary of the Model-2: AQM-RED-RPL

We have proposed a priority queue based AQM scheme called Active Queue Management in RED to Reduce Packet Loss (AQM-RED-RPL), In order to determine the likelihood of falling and labelling a packet to reduce the impact of network congestion, it utilizes its packet arrival rate and queue size. Comparative study with the previous AQM based schemes shows that our AQM-RED-RPLalgorithm not just to outshines the other strategies by reducing overall packet loss rate and higher goodput, and is more robust to keeping a secure queue with complex workloads. Queue reliability is a good function of an AQM policy as it tends to lower the risk of packet failure. The simulation results indicate the better performance of the proposed algorithm by reducing delay. In addition, high packet delivery ratio is achieved while not increasing the overhead significantly. For this analysis, we have used the NS-2 simulation application. The NS-2 simulator produces a rather comparable traffic load to the actual network. We believe that if we understand this analysis in the real world, we will get better results. We have planed to equate AQM-RED-RPL with more recently developed AQM algorithms with specific data parameters extracted from real environments in future work. Evaluating AQM algorithms under varying circumstances, such as multi-hop networks, cellular networks, or gigabit networks, will be very useful for further study.