Chapter 9

Model-6: AQM-RED-CPTQ

9.1 Introduction

Active Queue Management (AQM) algorithms play an important role to maintain the reliability of data packet routing in the MANET. Random Early Detection (RED) is the first successful queue management algorithm that has been suggested to apply on TCP / IP networks. RED has several tuning criteria that need to be carefully handled in order to have decent results under various network scenarios. We are proposing a new algorithm named Critical Point on Target Queue in **Model-6: (Active Queue Management in RED considering Critical Point on Target Queue: AQM-RED-CPTQ)**has greater congestion management across the network it can also preserv the value of RED and it works to enhance these criteria. Critical point on target queue and certain RED elements and their versions parameter have been added in this model. In Model-6:AQM-RED-CPTQ, we simulate the proposed algorithm using the well-known network simulator NS-2, by contrasting it with the initial RED. Simulation findings indicate that better queue size than RED is obtained by the suggested algorithm and the wait and losses are minimised.

9.2 Motivation and Objectives of Model-6: AQM-RED-CPTQ

The key points of congestion management systems is to maintain the network parameters, even though confronted with severe overload, pretty near to its rated potential. Both aims may be converted into two primary priorities, the first one is to eliminate the existence of network congestion until it happens and dissolve the congestion where it is not necessary to stop the occurrence of congestion. The second one is to provide the multiple links with a fair service, along with promoting separate internet application domains with varying criteria for quality of service (QOS). To overcome the drawback of previous proposed RED based approach with respect to packet loss, throughput and delay, AQM-RED-CPTQ has been developed.

RED's benefits are recognised by few researchers over drop tail routers. But there are some issues that decrease its efficiency. We propose to make some adjustments to the classic RED design unlike current RED improvement strategies. The majority of the original RED remains constant. The limitations of RED algorithm are as follows:

(i) 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.

(ii) The coming data packet is discarded with Pa probability even when the current queue size is empty. This is happen when the average queue size belong between the minimum and maximum value.

(iii) As communications minimize their sending percentage then window instantly decreases but the average queue size would decline slowly. If the average queue length is higher then entering data will be fall with greater probability in no congestion condition.

(iv) If congestion becomes instantly high, instant queue size will be increase 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 $minth$.

(v) RED efficiency is dependent on the number of competitors participating in flows / sources. When the load is high, the RED output is degraded.

(vi) Wild queue fluctuation is detected with RED when the traffic load is change.

(vii) RED achievement is conscious to the size of the packet.

(viii) RED output is incredibly susceptible to the configurations of its variables.

9.3 Proposed Scheme: Model-6 AQM-RED-CPTQ

We propose algorithm called AQM-RED-CPTQrelated to active queue management in RED considering Critical point on Target Queue. It aims to have greater management of congestion across the MANET while retaining the RED parameter values. As AQM-RED-CPTQ operates to render the queue more reliable, the algorithm focuses on improving the average queue size in a manner that restricts the queue size to minimise the wait and packet error rate relative to the RED queue. The calculation of the total queue size takes place as seen in the equation 9.1.

$$q_{avg} = (1 - w_q)q_{avg} + w_q \times q \tag{9.1}$$

It relies on the (w_q) queue weight parameter. The queue weight is calculated by the size and length of queue size bursts permitted at the gateway, Next to the w_q parameter, AQM-RED-CPTQ takes a new parameter named the goal queue (q_t) (i.e. the gap between the actual queue size and the highest threshold average and the minimum threshold). If the goal queue does not reach the crucial point prior to the buffer overload, AQM-RED-CPTQ will determine the following algorithm for the average queue capacity. The main key points of the proposed scheme are mentioned below:

Main feature of the Model-6 AQM-RED-CPTQ

- The weight of the queue, w_q, is calculated by the number and length of queue-size bursts authorised at the gateway.
- This technique introduced a new parameter, T_q , beside the w_q.
- T_q is the discrepancy between the scale of the present queue and the maximum and minimum threshold average.
- If the T_q does not reach the crucial point until buffer overload, according to the following algorithm, the methodology measures the average queue size

Algorithm 9.1: Algorithm of AQM-RED-CPTQ

input : Initialize the nodes
output: Congestion will be control with high throughput and lesser packet drop ratio and end to end delay

Target = (MaximumThreshold + MinimumThreshold) / 2;
for *(each arrival packet before the bufferoverflow)* **do**
    if *(AVG $<$ current-queue-size $<$ critical-point)* **then**
        Tq =AVG – Target;
        AVG = Tq (1- wq) + q * (Tq - wq) ;
    end
end

Here, Target = (MaximumThreshold + MinimumThreshold) / 2; for each arrival packet before the buffer overflow we have to check the condition. If the AVG is less than the current-queue-size which again less than critical-point then calculate $T_q = AVG–Target$; else $AVG = Tq(1 - wq) + q \times (Tq - wq)$.

9.4 Results and Comparison of Model-6: AQM-RED-CPTQ

To validate the proposed strategy, we implement it by some modifications on RED module in NS2 simulator. Then, it is run over the network and is compared with original RED algorithm in dynamic conditions of network. Data obtained after doing some experiments using ADWD-RED-IP, AQM-RED-RPL, PAQM-RS-RED, IAQM-TA-QZ, CCA-BO-RED and AQM-RED-CPTQ can be analyzed to study the performance of each model with respect to the different parameters such as End-to- End delay, Packet delivery ratio, Throughput and Goodput. By comparing their results with that of the existing RED, our proposed model AQM-RED-CPTQis the best model can be selected to improve the performance RED.

The simulation is performed with 100 nodes and the proposed path selection take random. The table 9.1 shows the performance measurement of the proposed AQM-RED-CPTQ scheme with various no of nodes from 2 to 100. In this experiments end to end delay varies from 325 to 148 Owing to the 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.

Table 9.1: *Experimental results of AQM-RED-CPTQ*

Nodes	End-to-End Delay	Packet Delivery Ratio	Throughput	Goodput
2	325.369	90.32	711.35	345.32
10	291.256	87.32	716.35	365.76
25	193.369	93.24	731.24	370.13
50	167.364	90.21	750.36	380.29
75	162.357	90.48	748.32	385.64
100	148.365	88.36	748.79	390.65

End to End Delivery: The ratio of packet received time to packet send time is termed as end to end delay. The end to end delay should be low in order to provide better performance. The Table 9.2 and graph (Fig. 9.1) represent the performance of routing end to end delay in continuous traffic pattern for RED, ADWD-RED-IP, AQM-RED-RPL, PAQM-RS-RED, IAQM-TA-QZ, CCA-BO-RED and AQM-RED-CPTQ.

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

Table 9.2: *End-to-End Delay of RED, ADWD-RED-IP, AQM-RED-RPL, PAQM-RS-RED, IAQM-TA-QZ, CCA-BO-RED and AQM-RED-CPTQ*

Nodes	RED	ADWD-RED-IP	AQM-RED-RPL	PAQM-RS-RED	IAQM-TA-QZ	CCA-BO-RED	AQM-RED-CPTQ
2	335.446	340.235	333.235	331.364	328.365	326.754	325.369
10	304.878	312.674	301.539	299.365	296.265	294.352	291.256
25	206.093	208.443	204.326	202.652	199.354	196.258	193.369
50	179.589	184.385	177.328	175.365	172.264	170.541	167.364
75	177.267	180.438	175.214	172.621	169.369	166.365	162.357
100	161.335	163.275	158.325	156.251	153.258	150.258	148.365

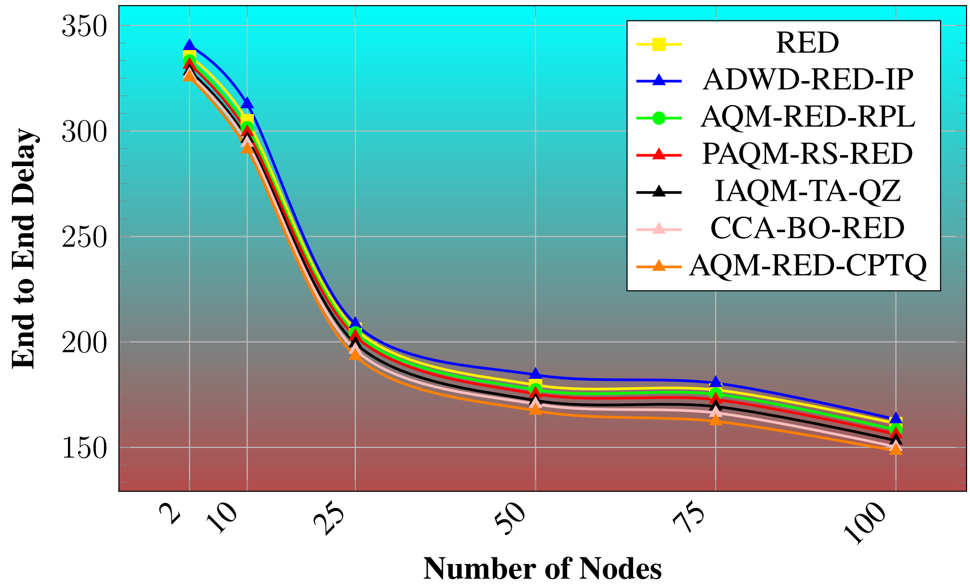

Figure 9.1: *Comparison of proposed schemes with respect to End to End Delay*

In this approach, congestion can be avoided and packet can be delivered within time while increasing number of nodes. Thus, the proposed AQM-RED-CPTQ, performs better with low end to end delay when compared with existing RED scheme.

Packet Delivery Ratio: Packet delivery ratio is defined as the ratio of the total numbers of packet send to the total number of packets received. The results are shown in Table 9.3 and the corresponding graphical representation is shown in Fig. 9.2. The Fig. 9.2 represents the routing packet delivery ratio for existing RED, ADWD-RED-IP, AQM-RED-RPL, PAQM-RS-RED, IAQM-TA-QZ, CCA-BO-RED and AQM-RED-CPTQ algorithm with respect to the number of nodes. Due to active queue management of AQM-RED-CPTQ, it is possible to receive more packet without any loss, and the proposed AQM-RED-CPTQ algorithm achieves high packet delivery ratio than RED and gives better result.

Table 9.3: *Packet Delivery Ratio of RED, ADWD-RED-IP, AQM-RED-RPL, PAQM-RS-RED, IAQM-TA-QZ, CCA-BO-RED and AQM-RED-CPTQ*

Nodes	RED	ADWD-RED-IP	AQM-RED-RPL	PAQM-RS-RED	IAQM-TA-QZ	CCA-BO-RED	AQM-RED-CPTQ
2	91.21	89.67	90.32	91.67	89.95	92.32	95.36
10	86.86	84.43	87.32	86.36	85.63	87.21	90.78
25	92.77	87.56	93.24	92.36	90.51	93.61	96.37
50	91.64	89.29	90.21	91.52	90.21	92.36	95.38
75	90.22	88.97	90.48	89.36	89.37	91.37	94.51
100	88.63	87.35	88.36	87.55	88.12	89.39	93.27

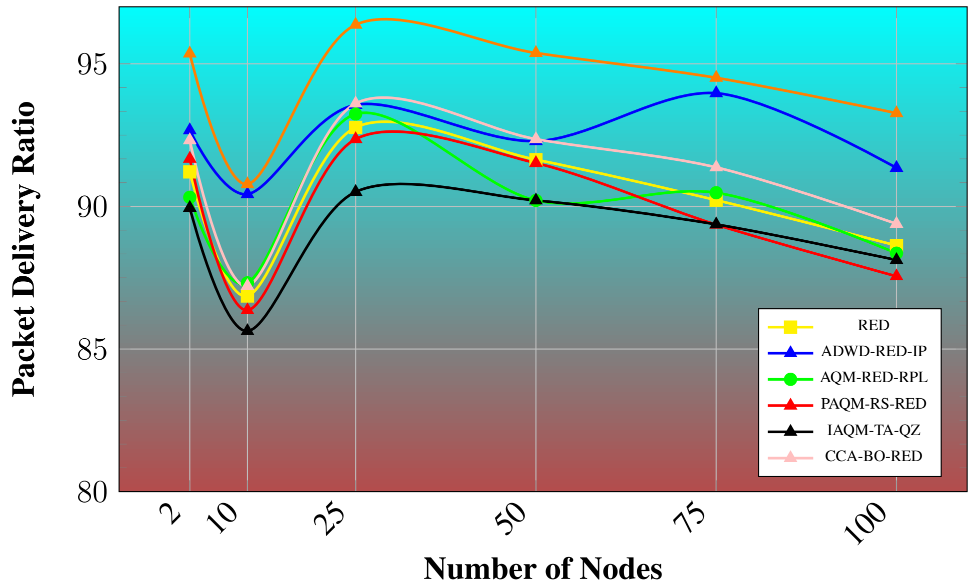

Figure 9.2: *Comparison of proposed schemes with respect to packet delivery ratio*

Throughput: Throughput is one of the important parameters for evaluating the performance.

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 shown in Table 9.4 and corresponding graph (Fig. 9.3) shown the performance comparison for RED, ADWD-RED-IP, AQM-RED-RPL, PAQM-RS-RED, IAQM-TA-QZ, CCA-BO-RED and AQM-RED-CPTQİt is analysed from the graph that, the throughput for the AQM-RED-CPTQ is gradually increasing when compared to the existing RED. Therefore, the proposed AQM-RED-CPTQ gives better throughput without loss.

Table 9.4: *Throughput of RED, ADWD-RED-IP, AQM-RED-RPL, PAQM-RS-RED, IAQM-TA-QZ, CCA-BO-RED and AQM-RED-CPTQ*

Nodes	RED	ADWD-RED-IP	AQM-RED-RPL	PAQM-RS-RED	IAQM-TA-QZ	CCA-BO-RED	AQM-RED-CPTQ
2	709.48	710.23	711.35	712.36	716.25	718.54	720.56
10	712.79	715.54	716.35	720.61	722.56	725.36	731.59
25	726.55	728.25	731.24	736.29	737.56	741.39	745.91
50	748.08	750.67	750.36	755.62	759.81	762.57	771.97
75	746.54	752.45	748.32	750.27	755.24	763.17	772.37
100	745.26	753.45	748.79	757.63	759.87	764.52	774.83

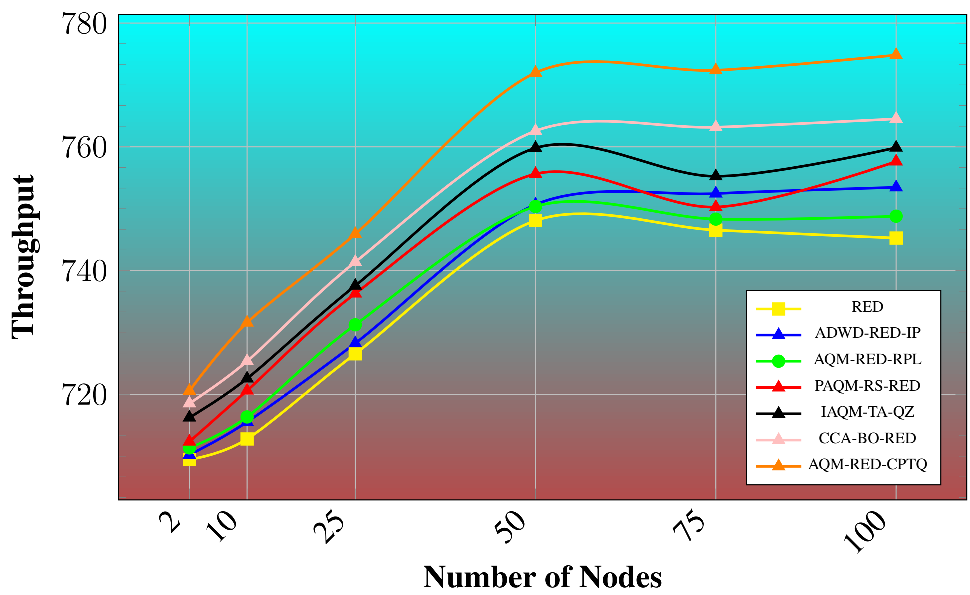

Figure 9.3: *Comparison of proposed schemes with respect to Throughput*

Goodput: The goodput is delivered as the number of useful information transmitted by the 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 9.5 shows the goodput comparison among RED, ADWD-RED-IP, AQM-RED-RPL, PAQM-RS-RED, IAQM-TA-QZ, CCA-BO-RED and AQM-RED-CPTQİn the proposed AQM-RED-CPTQ, the goodput is better than RED based algorithms because the packet drop function has been modified for the active queue management. The corresponding graph is presented in Fig. 9.4. The explanation, why AQM-RED-CPTQ has lower delay and jitter than rest algorithms is that 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-CPTQ are smaller than those of SRED, REM, BLUE and LDC algorithms. For real-time applications such as UDP in intermediate routers, the delay and jitter parameter values are lower enough to use AQM-RED-CPTQ˙

Table 9.5: *Goodput of RED, ADWD-RED-IP, AQM-RED-RPL, PAQM-RS-RED, IAQM-TA-QZ, CCA-BO-RED and AQM-RED-CPTQ*

Nodes	RED	ADWD-RED-IP	AQM-RED-RPL	PAQM-RS-RED	IAQM-TA-QZ	CCA-BO-RED	AQM-RED-CPTQ
2	337.85	388.76	345.32	401.43	406.83	411.86	417.08
10	355.04	408.32	365.76	423.77	429.01	432.61	438.42
25	367.21	431.45	370.13	456.96	464.23	469.63	473.03
50	374.04	455.39	380.29	478.39	481.57	485.76	490.71
75	377.04	462.68	385.64	479.76	482.06	487.65	493.05
100	382.18	478.49	390.65	487.29	493.04	498.46	502.67

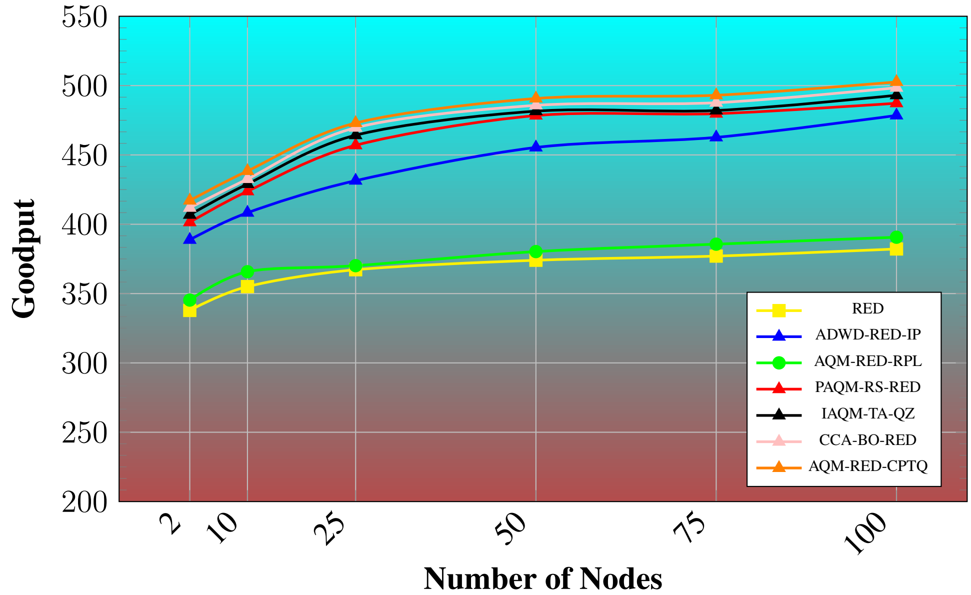

Figure 9.4: *Comparison of proposed schemes with respect to Goodput*

Table 9.6 represents the comparison of RED, ADWD-RED-IP, AQM-RED-RPL, PAQM-RS-RED, IAQM-TA-QZ, CCA-BO-RED and AQM-RED-CPTQ in terms of number of packet received, forwarded, dropped and loss rate for flows 20, 40, 60, ... 200 nodes. Here, packet loss rate is lower than RED due to active queue management with introduction of MIN_q and MAX_q parameters. The corresponding graph shows that the AQM-RED-CPTQ improves the performance of RED algorithm.

Table 9.6: *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
PAQM-RS-RED	8631.714	8428.428	203.286	0.0235	1.154119
IAQM-TA-QZ	8780.714	8579.540	200.892	0.0229	1.173448
CCA-BO-RED	8810	8719.320	190.320	0.0217	1.180437
AQM-RED-CPTQ	8900	8814.764	185.236	0.0210	1.180669

9.5 Summary of the Model-6: AQM-RED-CPTQ

This research gives an description of the process of congestion management and it is centred on the RED algorithm and its derivatives and their major roles in avoiding congestion, including our proposed algorithm. RED and other evolved RED-based algorithms are linked to the simulation performance. An improvement to the original RED algorithm named AQM-RED-CPTQ has been suggested, which does not include an end-system update. This scheme aims to decrease the RED queue's total queue capacity. The AQM-RED-CPTQ results in a limited queue capacity, resulting in less latency and a low rate of packet failure.

In this study, we have obtained result using NS-2 software by adjusting buffer size. If we change buffer size the above result will vary. Again, if we use other type of software then results may also vary. However, our process of investigation is novel one because the AQM-RED-CPTQ is showing best result with respect to less packet loss ratio and more throughput compare to existing RED.