Chapter 8

Model-5: CCA-BO-RED

8.1 Introduction

RED is the most widely deployed AQM based algorithms suggested for implementation on the MANET by the Internet Engineering Task Force (IETF). While RED promises low average queuing delay and high throughput at the very same time, RED performance is highly responsive to the setting of RED parametric variable. It is not an optimal approach to set RED parameters with defined values because system requirements differ significantly. For dynamically tuning RED parameters, we propose a new **Model-5: A Nobel Congestion Control Algorithm Using Buffer Occupancy RED (CCA-BO-RED)**. We measure the rate at which the queue is filled and treat it as a congestion parameter that will be expected whenever the queue is crowded. This parameter is being used to configure RED parameters dynamically. The simulation findings indicate the feasibility of the approach suggested. According to the observations, in complex network environments, we obtain considerably greater utilisation and less packet loss relative to the initial RED algorithm.

Congestion control is a vital problem in MANET. Congestion control has been given due importance for last many years by the researchers to ensure the good utilization of through put and performance. It is sub divided mainly into two parts: TCP and AQM. Transaction Control protocol (TCP) is not adequate for congestion control of Internet. Among the several approaches, RED, as one of the Active Queue Management (AQM) approach most widely used which are recommended by the Internet engineering Task Force (IETF) for deployment on the MANET, RED offers low average queuing delay and fast throughput at the same time. But RED efficiency is very responsive to setting of parameters. As the condition of network changes frequently, so the values of the parameters are dynamic. Therefore tuning of the parameters is required to obtain best results.

In this Model-5 (CCA-BO-RED), an attempt has been made to evaluate the effectiveness of RED by using dynamic buffer size parameter. For this algorithm (CCA-BO-RED) we have developed a new approach and four modes of performance parameters tested with respect to End to End delay, packet delivery ratio, Throughput and Goodput. The simulation results have been compared with that of RED. In this work, to rectify the shortcoming of the original RED algorithm, we are suggesting a new hierarchical tuning system for RED parameters dependent on network conditions. For this purpose, we introduce a new measure that is total buffer size and occupied cell (BOC)of buffer in each interval. If BOC is zero or less than zero, then BOC is a 15 percent of the total buffer size and else if BOC is higher than zero, than it varies from 0 to 14 percent of total buffer size according to BOC value. The proposed algorithm has been implemented in NS2 simulator. The effects of the simulation demonstrate the efficacy of the suggested algorithm, which addresses the issue of setting parameters and greatly increases the efficiency of RED.

8.2 Motivation and Objectives of Model-5: CCA-BO-RED

The main objective in this study is to develop an algorithm to overcome the packet loss so that the performance of the existing RED can be improved. The main motive of this study is to achieve higher utilization of Internet so that congestion control will be effective.

The weakness 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 coming data packet is discarded with P_a probability even when the current queue size is empty. This is happen when the average queue size belong between the minimum and maximum value.
- 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.
- 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 $minth$.

- 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 change.
- RED achievement is conscious to the size of the packet.
- RED output is incredibly susceptible to the configurations of its variables.

8.3 Proposed Scheme: Model-5: CCA-BO-RED

In order to design efficient RED based routing mechanism, its control parameters should be set to proper values according to the condition of the network (congestion level, the number of active connections, link bandwidth, and so on). However, as noted by other researchers [33], [4], it is infeasible to gain one parameter set to make the RED algorithm work effectively in dynamic condition of network. If the control parameters of RED are not configured correctly, then its throughput is often under that of drop-tail routers. To overcome the limitations of RED algorithm, we introduce a new variable which indicates network condition, and then, control RED's parameters which need to be configured dynamically based on it. Our new metric, that called BOC (Buffer Occupancy), infers network conditions from rapidity of the buffer occupancy in the router. To do this, we measure speed of change in the quantity of occupied cells of buffer. If the quantity of occupied cells is increased, then the BOC will have a positive value proportional to the network congestion level, meaning that offered load to the network is increasing. If the quantity of occupied cells is decreased, the quantity of free cells is increased, then the BOC will have a negative value, meaning that offered load to the network is decreasing. So, with BOC metric that indicates network dynamic conditions, we can dynamically adjust the RED's control parameters proportional to network dynamic conditions. In our algorithm CCA-BO-RED, the aggressiveness of RED is tuned by updating of only MAX_{th} and MIN_{th} based on BOC value. The main key points of this Model is given below:

- A new parameter, BOC, is introduced here. It indicates effective network condition.
- The other RED's control parameters are configured dynamically based on it. The speed of change is measured in the quantity of occupied cells of buffer.

- If the quantity is increased, then the BOC will have a positive value proportional to the congestion level.
- If the quantity is decreased, then the BOC will have a negative value.

8.3.1 Algorithm: Model-5: CCA-BO-RED

The proposed algorithm to dynamically RED's parameter setting are given below, where *Total Buffer Size* and *Occupied Cell* of buffer in each interval is considered. In CCA-BO-RED algorithm, the value of MIN_{th} is 5 plus C so that C is proportional to BOC value. If BOC is zero or less than zero, then BOC is a 15% total buffer size and else if BOC is higher than zero, it varies from 0% to 14% of total buffer size according to BOC value.

Algorithm 8.1: Algorithm of CCA-BO-RED

input : Initialize the nodes

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

```
for (each interval i separated by function of 2 cells in the buffer) do
    BOC is calculated;
    if (BOC ¡= 0) then
        C = 0.15 * (Total Buffer Size - Occupied Cell );
    else
        C = 1 / (BOC * (Total Buffer Size - Occupied Cell));
        MinimumThreshold = 5 + C;
        MaximumThreshold = 3 * MinimumThreshold;
    end
end
```

8.4 Results and Comparison of Model-5: CCA-BO-RED

In order to evaluate the proposed approach, we implement it by some modifications on RED module in NS2 simulator. Then, it is run over the network and is then compared with original

RED algorithm in dynamic conditions of nodes. Data obtained after doing some experiments using ADWD-RED-IP, AQM-RED-RPL, PAQM-RS-RED, IAQM-TA-QZ and CCA-BO-RED it has been analyzed to study the performance of each model with respect to the following parameters: End-to- End delay, Packet delivery ratio, Throughput and Goodput.

The simulation is performed with 100 nodes and the proposed paths are selected randomly. The table 8.1 shows the performance measurement of the proposed CCA-BO-RED scheme with various number of nodes from 2 to 100. End to End delay varies from 326 to 150. It depends on the number of nodes used. 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 8.1: *Experimental results of CCA-BO-RED*

Nodes	End-to-End Delay	Packet Delivery Ratio	Throughput	Goodput
2	326.754	90.32	711.35	345.32
10	294.352	87.32	716.35	365.76
25	196.258	93.24	731.24	370.13
50	170.541	90.21	750.36	380.29
75	166.365	90.48	748.32	385.64
100	150.258	88.36	748.79	390.65

End to End Delay: 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 8.2 and Graph as shown in Fig. 8.1 describes the performance of routing end to end delay in continuous traffic pattern for ADWD-RED-IP, AQM-RED-RPL, PAQM-RS-RED, IAQM-TA-QZ, CCA-BO-RED and RED.

In the proposed CCA-BO-RED, due to the introduction of BOC value, the packet moves smoothly with a little bit data loss. Here congestion can be avoidable and packet can be delivered within time while increasing mode. Thus the proposed CCA-BO-RED performed good with low end to end delay when compared with existing scheme.

Table 8.2: *End-to-End Delay of RED, ADWD-RED-IP, AQM-RED-RPL, PAQM-RS-RED, IAQM-TA-QZ and CCA-BO-RED*

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

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

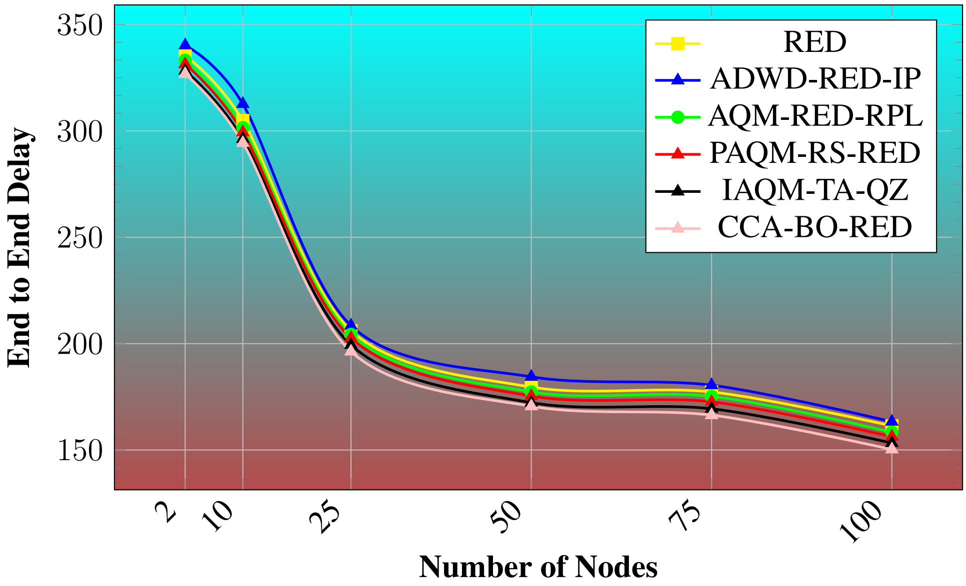

Figure 8.1: *Comparison of proposed schemes with respect to End to End Delay*

Packet Delivery Ratio: Packet delivery ratio is termed as the ratio between total numbers of packet send to the total number of packets received. The results are listed in Table 8.3 and the corresponding graphical representation is shown in Fig. 8.2. The Fig. 8.2 represents the routing packet delivery ratio for existing RED, ADWD-RED-IP, AQM-RED-RPL, PAQM-RS-RED, IAQM-TA-QZ, and CCA-BO-RED algorithm with respect to the number of nodes. Due to introduction BOC value of CCA-BO-RED, it is possible to receive more packet without any loss, and the proposed CCA-BO-RED algorithm achieves high packet delivery ratio than RED and gives better result.

Table 8.3: *Packet Delivery Ratio of RED, ADWD-RED-IP, AQM-RED-RPL, PAQM-RS-RED, IAQM-TA-QZ and CCA-BO-RED*

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

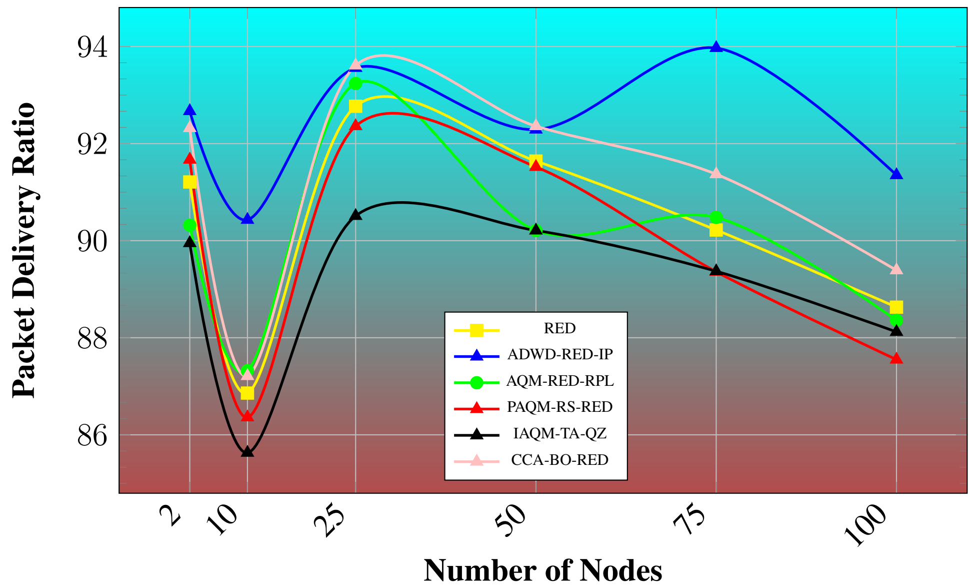

Figure 8.2: *Comparison of proposed schemes with respect to packet delivery ratio*

Throughput: Throughput is one of the important parameter for evaluating the performance. The throughput is calculated based on number of bits transmitted per second. In order to provided better performance of the network the system throughput must be high. The simulated results are shown in Table 8.4 and corresponding Graph (Fig. 8.3) is shown the performance comparison for RED, ADWD-RED-IP, AQM-RED-RPL, PAQM-RS-RED, IAQM-TA-QZ and CCA-BO-RED. It is analysed from the graph that, the throughput for the CCA-BO-RED is gradually increasing when it is compared to the existing RED. Therefore, the proposed CCA-BO-RED gives better throughput without loss than RED.

Table 8.4: *Throughput of RED, ADWD-RED-IP, AQM-RED-RPL, PAQM-RS-RED, IAQM-TA-QZ and CCA-BO-RED*

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

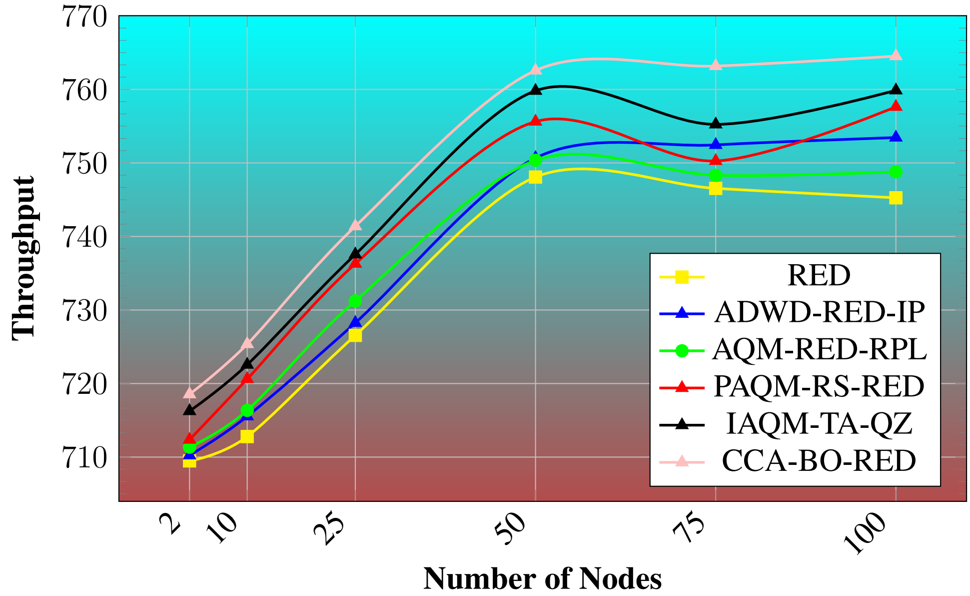

Figure 8.3: *Comparison of proposed schemes with respect to Throughput*

Goodput: The goodput is defined 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 8.5 shows the goodput comparison among RED, ADWD-RED-IP, AQM-RED-RPL, PAQM-RS-RED, IAQM-TA-QZ, and CCA-BO-RED. In the proposed CCA-BO-RED, 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. 8.4. The BOC value is taking the responsibility for the fast forwarding by managing the queue buffer. The jitter values of CCA-BO-RED are smaller than those of SRED, REM, BLUE and LDC algorithms. The delay values are lower enough to use CCA-BO-RED in intermediate routers for real-time applications such as UDP.

Table 8.5: *Goodput of RED, ADWD-RED-IP, AQM-RED-RPL, PAQM-RS-RED, IAQM-TA-QZ, and CCA-BO-RED*

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

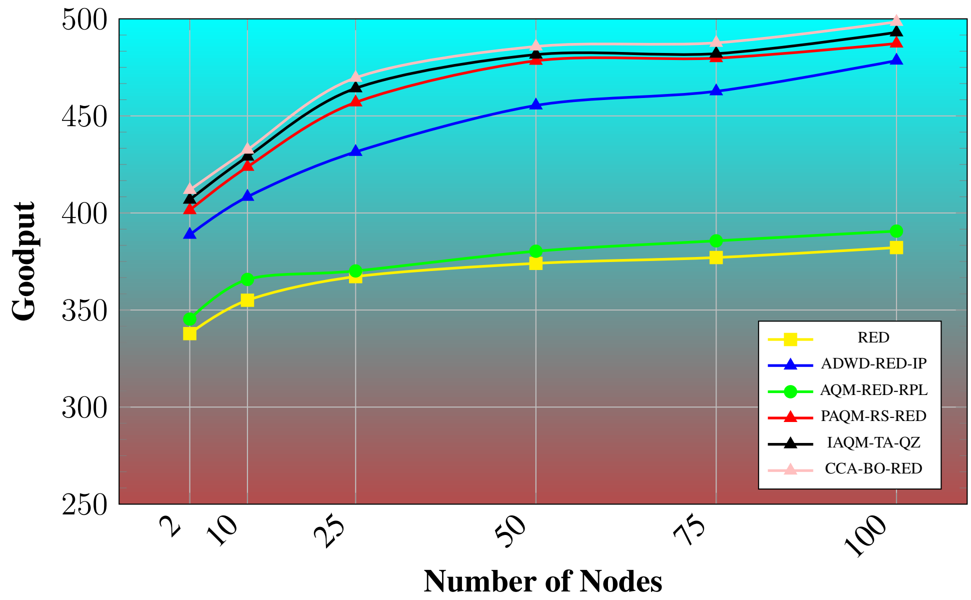

Figure 8.4: *Comparison of proposed schemes with respect to Goodput*

Table 8.6 presented the comparison of RED, ADWD-RED-IP, AQM-RED-RPL, PAQM-RS-RED, IAQM-TA-QZ, and CCA-BO-RED 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 then RED due to active queue management with introducing MIN_q and MAX_q parameters. The corresponding graph shows that the CCA-BO-RED improved the performance of RED algorithm.

Table 8.6: *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

8.5 Summary of Model-5: CCA-BO-RED

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. However, our approach of investigation is novel one because the CCA-BO-RED is showing best result with respect to less packet loss ratio and more throughput compare to existing RED. This work proposes a new method to improve the RED algorithm performance. We have introduced a new metric called BOC to detect any change in the network congestion status. Then, we have configured RED's control parameters (MAX_{th} and MIN_{th}) based on the BOC. The simulation findings indicate the feasibility of the approach suggested. We have obtained considerably higher usage and less packet loss relative to the initial RED algorithm in complex network environments, according to the findings. It is important to note that the proposed algorithm solves the parameters setting problem without making additional overhead to the original algorithm. It is notable that BOC is reliable to apply in other problems that need predicting network load status without making additional overhead.