

Performance Improvement by Dynamic Queue Management in Mobile Ad hoc Network

Kajal Yadav
Department of CSE
RKDF Institute of Science &
Technology, Bhopal

Prof. Gaurav Shrivastava
Department of CSE
RKDF Institute of Science &
Technology, Bhopal

ABSTRACT

Mobile Ad hoc network [MANET] consists of mobile nodes without any fixed infrastructure, communicating through wireless medium. In MANET, Congestion occurs in any intermediate nodes when data packets travel from source to destination and they incur higher packet loss rate and long delay, which cause the degradations in performance of a network. Congestion can be controlled by Active Queue Management [AQM] like Random Early Detection [RED]. But RED is very sensitive to traffic load and parameter configuration, and its average queue length varies greatly with the congestion degree and parameter settings. To resolve the above problems, this paper proposes Performance Improved RED (named PIREN) by using auto-tuning RED Queue thresholds parameters and packet dropping probability. An adaptive mechanism is designed to adjust the packet dropping probability and RED Queue thresholds parameters for stable average queue length. The key idea is that when the traffic load changes and therefore the queue length deviates from the target value, the algorithm adjusted RED Queue thresholds parameters and packet dropping probability to improve the network performance over the conventional RED.

Keywords

Mobile Ad hoc Network, Active Queue Management, Average Queue length

1. INTRODUCTION

Mobile Ad hoc Network [MANET] [1] is self-organizing network of mobile devices which does not rely in any fixed infrastructure. Mobile devices in MANET can take part in the communication if they are within the range of network, and can move freely within transmission range of network. Mobile devices which are outside the transmission range of network cannot take part in communication. The dynamic nature of MANET with limited resources, can vary with time such as battery power, storage space bandwidth makes QoS provisioning, a challenging problem.

Congestion within the network happens when the aggregated demands exceed available resources. But merely increasing network resource is unable to address the congestion problem. Various congestion control methods have been proposed to solve this issue.

The mobile network nodes these days use Drop tail queue management [2] where packets are discarded when the queue is full and multiple packets are dropped due to queue overflow, multiple transmission control protocol flows (to

which these packets belong to) back off. However, as there is a considerable amount of delay between the packet drop at the queue and the notification at the sender, large number of packets may be dropped as the sender continues sending at its current rate. Multiple flows back off results in under utilization of the queue (no congestion).

2. EXISTING SYSTEM

Floyds et al proposed Random Early Detection (RED) [3] in 1993. The basic idea of this mechanism is that the router can detect incipient congestion by monitoring the average queue length. If congestion is detected, router selects the source terminal to notify the congestion. So that source terminal can reduce the data transmission rate before the queue overflow, and try to alleviate the network congestion.

RED algorithm consists of two steps in first step it calculates the average queue length, and in the second step it calculate the packet drop probability. Packet drop probability is used to decide whether to drop the packet or not, packet drop is treated as the signal of congestion.

RED suffers from several shortcomings, such as sensitivity to parameter configurations and traffic load. The average queue length of RED highly depends on the number of traffic flows and RED's parameters, especially the maximum packet dropping probability (p_{max}).

Random Early Detection algorithm [3][4], initialize the threshold parameters (Min_{th} , Max_{th}) to the fixed value before simulation starts, these values remains fixed during the simulation and maximum packet dropping probability (p_{max}) increases linearly with increasing of network congestion. In MANET, due to quick diversity of changing the network congestion status, fixed threshold parameters are not appropriate because performance of RED algorithms is much sensitive to parameter setting.

RED is an effective approach in the sense that it can push back the traffic such that the average incoming rate is equal to the network capacity [5]. On the other hand, RED has the innate characteristics of oscillation in queue size, due to varying round trip time, congestion levels, etc, resulting in sensitivity of parameter setting, under-utilization of bandwidth, and potential instability [6], [7] By adaptively tuning the parameters, RED achieves improved stability with reduced oscillation and higher throughput [6]. Nevertheless, the oscillation characteristic, mainly due to round trip delay in the control loop, is hard to prevent as it is complicated to have

the RED attempt to dynamically adjust its parameters optimally as a function of average queue size. In addition, when pushing back the traffic, subject to the equilibrium of incoming rate and link capacity, a large dropping probability may occur, this is undesirable. Proposed algorithm combines the RED and window adjustment methods [9], leveraging the advantages from both while mitigating the drawbacks. Specifically, the algorithm estimates the average queue size to estimate congestion level in the network. It employs RED when the congestion level is exceeding beyond certain thresholds. In the meanwhile, the packet dropping probability is monitored. The algorithm applies window adjustment to mitigate the packet dropping probability and traffic oscillation. When the dropping rate is lower than a threshold, the window is increased; otherwise, it is decreased. As a result, the proposed approach can perform TCP flow control with fast response and improved stability.

3. PROPOSED SYSTEM

Proposed system is based on dynamic setting of RED parameters according to the changing of network congestion status. RED parameters are dynamically adjusted with the changing of network congestion to improve the network performance such as packet delivery ratio, network throughput and end to end delay without changing the basic functionality of RED algorithm.

3.1 PROPOSED ALGORITHM

Proposed algorithm Performance Improved RED (named PIREd) is using the basic idea of Random Early Detection algorithm with some modification in working is mentioned below:

To utilize full queue space with minimum delay PIREd gradually adjust the maximum threshold value (Max_{th}) to maximum available queue size, when average queue size is exceeding to maximum threshold value, this utilization available queue space without additional delay.

Random Early Detection uses the fixed window size due to fixed thresholds parameters (Min_{th} and Max_{th}). Proposed performance improvement mechanism uses variable window size which is adjusting both threshold parameters automatically to keep small window size, because small window size responds quickly on each incoming packet. Calculation of packet drop probability is similar as in random early detection algorithm.

To understand algorithm, it is divided into three steps according to its working. In step 1 comparison of average queue size (Avg) with Max_{th} and if Avg is exceeding from Max_{th} then Max_{th} is increased by one each time to utilize available buffer. Steps 2 adjust both threshold values if condition is fulfilled. Step 3 shift lower Min_{th} by one if condition is fulfilled given in that step. Calculation of probability (calculate_p) is same as Random Early Detection mechanism.

Step 1:

```
If ( Avg > Maxth )
  { If ( Maxth < Queue_size )
    { Maxth = Maxth + 1 ;
      P = calculate_p ;
    }
    else
      P = 1 ;
  }
```

Step 2:

```
else if ( Minth < Avg <= Maxth )
  { If ( Thdiff >= 0.5 * Queue_size )
    { Mid = ( Minth + Maxth ) / 2 ;
      If ( Avg < Mid )
        { Maxth -- ; Minth -- ; }
      else { Minth ++ ; Minth ++ ; }
      P = calculate_p ;
    }
  }
```

Step 3:

```
else ( Avg <= Minth )
  { If ( Thdiff >= 0.5 * Queue_size )
    { Maxth = Maxth - 1 ;
      P = calculate_p ;
    }
    P = 0 ;
  }
```

4. SIMULATION AND RESULTS

Proposed algorithm simulated by using the network simulator NS2 (version NS-2.34) algorithm investigate the implications of using RED and PIREd in MANETs. To calculate performance of the network, following parameters average end-to-end delay, network throughput, Packet loss ratio has been used. These parameters are used for calculation of result of existing mechanism and also for proposed solution approach of performance improvement mechanism.

4.1 PERFORMANCE METRICS

4.1.1 Average end to end Delay

The average end-to-end delay of data packets is the interval between the data packet generation time and the time when the last bit arrives at the destination. End-to-end delay generally includes all delays, along the path from source to destination. This includes the transmission delay, queuing delay, propagation delay, processing delay experienced at every node in the route.

4.1.2 Network Throughput

Network throughput is the average rate of successful message delivery over a communication channel. This data may be delivered over a logical or physical link, or may pass through a certain network node. The throughput is generally measured in bits per second (bit/s or bps), and sometimes in data packets per time slot or data packets per second.

4.1.3 Packet loss ratio

Packet loss occurs when one or more packets of data travelling across a computer network fail to reach their destination. Packet loss is calculated as total lost packet to the total number of transmitted packets.

4.2 Simulation Scenarios

To calculate the results network simulator-2 has been used. Simulation consists of 20 nodes within the area of 500*500 sq meters. Average node movement speed is set to 5 meter/sec and queue length is 60 packets.

4.3 RESULTS

4.3.1 Delay V/s Time xgraph

In this graph, simulation time is plotted in x-axis and delay (in ms) of each packet is plotted in y-axis. In this graph packet transmission starts at 10 second and ends at 55 second, thus total packet transmission time in simulation is 45 seconds. In this graph delay has been plotted against their respective packet transmission time.

In both cases aggregate delay is proportional to the number of packet received at the destination, if packet reception is higher, then higher delay will occur, which is verified by numerical analysis of simulation result. In PIRED due to higher packet delivery ratio and higher throughput delay is higher than the convention RED.

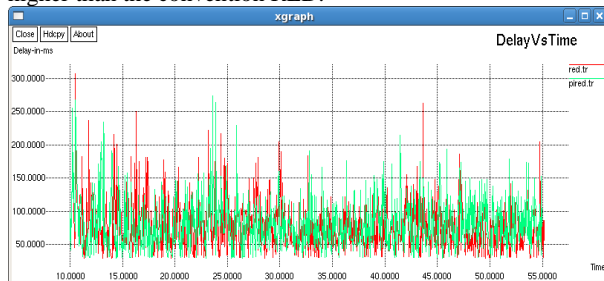


Figure 1: Packet Delay V/s Simulation Time xgraph

4.3.2 Packet Delivery Ratio V/S No of TCP Flows xgraph

In this graph, x-axis plots number of active transmission control flows and y-axis plots respective packet delivery ratio (PDR) of both RED and PIRED. Simulation is performed by using 20 mobile nodes within area of 500*500 sq meters. Queue length is 60 packets and average packet size is 1000 bytes.

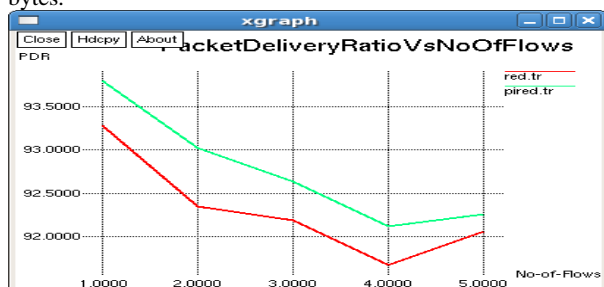


Figure 2: Packet Delivery Ratio V/s No of Flows xgraph

Figure 2 shows that packet delivery ratio of PIRED is higher than RED. Numerical analysis is done in Table 1. Numerical analysis shows that PDR is consistently higher in PIRED than RED. Numerical result also shows that PDR has been dropped in both mechanisms by increasing number of flow due to increasing of network congestion.

Table 1: Packet Delivery Ratio

S. No.	RED	PIRED
1	93.28	93.8
2	92.35	93.02
3	92.19	92.64
4	91.67	92.12
5	92.06	92.26

4.3.3 Throughput V/s No of TCP Flows xgraph

In this graph, x-axis plots number of active transmission control flows and y-axis plots network throughput. Network throughput is plotted against their respective number of TCP flows.

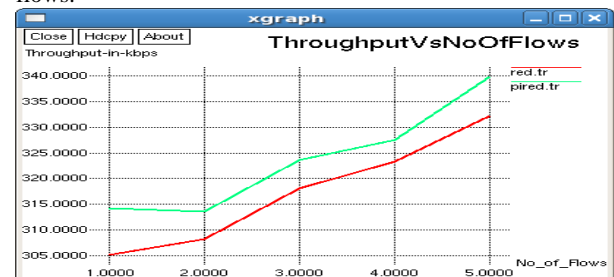


Figure 3: Throughput V/s No of Flows xgraph

Figure 3 shows that throughput of PIRED is higher than RED. Numerical analysis is done in Table 2. Numerical analysis shows that throughput is consistently higher in PIRED than RED. Numerical result also shows that throughput has been increased by increasing number of flow due to increasing of multiple TCP flows.

Table 2: Network throughput

S. No.	RED	PIRED
1	305.10	314.28
2	308.28	313.65
3	318.11	323.62
4	323.35	327.46
5	332.17	339.9

5. CONCLUSION

An algorithm for active queue management in Mobile Ad hoc network is proposed in this paper called performance improvement RED (PIRED). A dynamic adjustment policy is designed to regulate the maximum packet dropping probability and threshold parameters adjustment using average queue length. The proposed algorithm was compared with RED by using NS2 simulations. The results demonstrate that PIRED outperforms RED in terms of network throughput

and packet delivery ratio. PIRED stabilizes the queue length and is no longer sensitive to traffic loads, even in the dynamic network with time-varying traffic flows. Moreover, PIRED does not introduce extra system variable or too much overhead.

6. REFERENCES

- [1] Patel j. k., Dubey j., "Mobile Ad Hoc Network Performance Improvement Using Strategical RED", Ninth International Conference on Wireless and Optical Communications Networks WOCN2012, 2012, IEEE.
- [2] K.Dinesh Kumar, I.Ramya & M.Roberts Masillamani, "Queue Management in Mobile Adhoc Networks (Manets)" 2010 IEEE.
- [3] S. Floyd and V. Jacobson. "Random early detection gateways for congestion avoidance", IEEE/ACM Transactions on Networking, 1(4):397–413, 1993.
- [4] V. Firoiu and M. Borden, "A study of active queue management for congestion control", In Proceedings of the IEEE Infocom, pages 1435–1444, Tel Aviv, Mar 2000.
- [5] L. Wang, L. Cai, X. Liu and X. Shen, "Bounds estimation and practical stability of AIMD/RED systems with time delays," Computer Networks, 2009.
- [6] W. Feng, D. Kandlur, D. Saha and K. Shin, "A Self-Configuring RED Gateway, " Proceedings of IEEE Infocom'99, Vol. 3, No. 1, pp. 1320-1328, March 1999.
- [7] P. Ranjan, E. Abed and R. La, "Nonlinear Instabilities in TCP-RED," IEEE/ACM Trans. on Networking, Vol. 12, No. 6, pp. 1079-1091, December 2004.
- [8] The network simulator-ns-2, <http://www.isi.edu/nsnam/ns/>
- [9] Rob Torres, John Border, George Choquette, Jun Xu, and Je-Hong Jong, "Congestion Control using RED and TCP Window Adjustment" 978-1-4673, 2013 IEEE.
- [10] Zhenyu Na and Qing Guo "An Improved AQM Scheme with Adaptive Reference Queue Threshold" 978-1-4577, 2011 IEEE.
- [11] Tolaimate Ichrak and Elalami Nourredine "Basic control theory Applied to the Active Queue Management model" 978-1-4673 1520, 2012 IEEE.