International Journal of Engineering and Applied Sciences (IJEAS)
ISSN: 2394-3661, Volume-4, Issue-5, May 2017 (Approved by University Grants Commission, India)

Congestion control in mobile ad-hoc networks using Intelligent Priority Queueing Scheme

M.Rajeswari, Dr.N.Kasiviswanath

***Abstract*— Congestion control in mobile ad-hoc networks (MANET) is an area of interest in networking which has got a noticeable attention by researchers during the recent years. There exist different mechanismsfor providing a solution to congestion problem in MANETs, each of which concentrates on one or more key parameters of MANET such as Load, Remaining Energy, Signal Strength, Priority Queueing etc. to achieve the goal. Different proposals were made by various researchers with an intention of providing ananswer to the problem. However, based on the design approach, each of those methods carry their own set of pros and cons. In this paper, we propose a unique methodology to reduce congestion in MANET using a very intelligent queueing scheme that achieves an efficient usage of the limited network resources at the node. This mechanism can be further combined with various existing routing protocols to achieve a significant performance in mobile ad-hoc networking while attaining the core objective. As the proposed method operates at a very basic level of routing, it can be adopted easily byvarious existing congestion control protocols.**

***Index Terms*— Intelligent Priority Queueing Scheme (I.P.Q.S), Buffer Management, Congestion Control**

I. INTRODUCTION

Mobile Ad-hoc Network (MANET) is genre of Networking wherein a group of inter connectedwireless hostsconstructa network on-the-fly without requiring to have a pre-arranged infrastructure like routers, switches, etc. MANETs are useful to enable a temporary communication system at places where it is not feasible to setup the infrastructure like areas involved in emergency military operations or the places where the existing infrastructure is damaged due to natural calamities or other reasons. The mobile nodes in MANET act independently while facing various challenges like node-mobility, changing network topology, unsteady weather conditions, limited energy, less bandwidth, low memoryand less storage space etc. Congestion in the network is a major problem in MANET that degrades network performance and finally leads to data loss.

One of the key characteristics of MANET is the functionality of each node in the network to act as a router and a host as well. This is a feature MANET that overcomes the requirement of having an existing infrastructure for communication. With routing capability, each node in the network can forward traffic to the next hop on behalf of its predecessor node on a specific network path. As long as a node is within the communication range of the network, it can take part in communication while it is still allowed to move unrestrictedly in random directions. Mobility, Loss of energy and Link failure are the factors that usually cause a change in topology of the network.

M.Rajeswari, ResearchScholar, Dept. of Computer Science, Rayalaseema University

N.Kasiviswanath, Ph.D, Professor&Head, Dept. of CS & Engineering , GPREC,Kurnool.

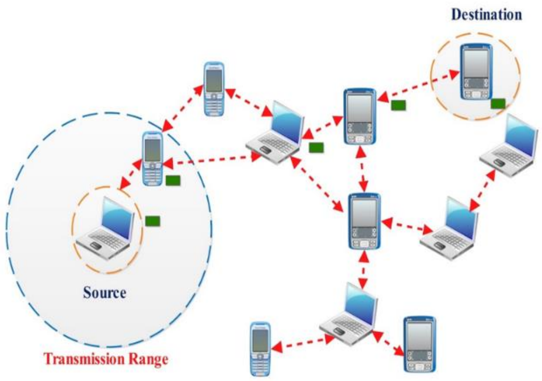

Fig 2.1 Mobile Ad-hoc Network

II. PROBLEM DEFINITION

Congestion is a situation in a network in which packet arrival rate at a routing device goes beyond its transmitting capacity. Imagine a situation where node ***N*** enters into a mode that makes it to serve three other neighbouring nodes ***N1, N2*** and ***N3*** onroutes ***R1, R2*** and ***R3*** respectively. Nodes ***N1, N2*** and ***N3*** start flooding data to node ***N***. During transmission, either if a link failure happens or the next hop stops responding on ***R2***, packets emerging from ***N2*** start to wait for more time, imposing an increased queueing delay for packets coming from other sources like ***N1*** and ***N3***too. Gradually this situation could lead to a buffer overflow. Later on, the packetsarriving at this node start to get dropped. This situation results in an increased packet loss rateand delay and decreased throughput etc.Maintaining a single queue is not at all a good idea as it makes packets with good data rate also to suffer from queueing delays. Drop Tail methodcan't help reducing the PLR (packet loss rate). Hence, a proper queue management scheme is vital in reducing packet loss rate for helping achieve better throughput.

III. EXISTING QUEUE MANAGEMENT SCHEMES

There are various ways to develop a congestion control model with a choice of implementation area being one of various stages of communication. Some sort of solutionssuch as equipping the node with a high performance processor, a high linkspeed are not feasible yet. Also, it is not recommended to equip the nodes withlarge queues as it would impose increased waiting time forthe packets waiting at the end of the queue and therefore, a proper queue management mechanism with a reasonable queue size is very important for effective network scheduling. Some of the popular queue management schemes are discussed below.

RED

Random Early Detection (RED)is a popular methodology that works on the idea that, it is better to detect the congestion possibility well in advance than to drop the packets when the buffer is full. This could not be achieved by Drop Tailmethod. Further, Drop Tail introduces a problem known as TCP global synchronization. Because all TCP connections are hold back and step forward simultaneously,it results in under-utilization of network resources. RED monitors queue length and starts to drop or mark packets with ECN (Early Congestion Notification)[1]based on statistical probabilities. As the queue length grows, the probability for dropping of an incoming packet grows too. So the probability of PLR(packet loss rate) at a host is proportional to the length of the queue. Congestion information is made available to the sender by setting ECN bit.

WRED

Classic REDhas no support for QoS. In Weighted Random Early Detection (WRED), there are different probabilities for different priorities [2]to accommodate various QoSconsiderations. In RED, several thresholds each associated with a different traffic class were maintained by a single queue. Prioritization of packets is not fair in RED as only high priority queues get service and standard traffic is not served efficiently. WRED overcomes this with a fair prioritization model for packets arriving from different traffic classes.

AQM

Active Queue Management notifies source node well in advance before the queue is getting exhausted enabling the sender to reduce the RoT (Rate of data transmission). Later, de-que and enqueue process happens between different queues until enough space is made available in the queue that it reaching its buffer size. The sender is then allowed to send more packets[3]. This model stood as a template for RED, REM and many other queueing models.

REM

Random Exponential Marking is a kind of AQM, aims at minimum loss and delays and efficient buffer usage. A variable called price is maintained by output queues to determine the marking probability.REM matches user rates against network capacity and embeds the congestion measures (sum of link prices) over all the routers on user path to the end-to-end marking possibility [4]. Price is updated asynchronously based on difference between link capacity and input rate, mismatch in rate, and variance between target and queue length etc. If sum of these mismatches is positive, price is incremented. Negative sum decreases the price. A positive number of weighted sumsignals the sender about the congestion. Sender then reduces data rate, Small source rates indicates negative mismatches and reduces marking probability and raises source ratesuntil mismatches become zero. High utilization of queue with a minimaldelay and loss are expected in this model.

AVQ

Adaptive Virtual Queue offers less delay, low loss, maximum utilization at link level. AVQ algorithm maintains a virtual queue with capacity less than the actual link capacity. Packet arrival at real queue is replicated also in the virtual queue. When the virtual buffer overflows, packets in the real queue are either dropped or marked [5]. Then, at each link, this virtual queue capacity is modified to make total flow hitting at each link utilizes a fair allocation of the link. In the absence of delays in feedback, the model is fair in maximizing the sum of utilities of all the users.

CFR

Congestion Free Routing defines a dynamic mechanism to monitor resource usage at node. Congestion is estimated at node level by calculating average queue length. Status of congestion is divided into the three zones i.e., safe zone, likely to be congested zone and congested zone). CFR makes use ofnon-congested neighbour nodes for discovering alternate non-congested route. Calculation of congestion status is done periodically by the nodes involved in transmission. The predecessor node is made aware of the congestion status to enable it to find an alternativeroute. With this methodology, CFR improved packet delivery ratio.

IV. Related Work

IyyapillaiAmbika*etal*. [2014]proposed an effective queueing methodology, to support both elastic and normal traffic. Packets from inflexible flows are stored ahead of elastic flows. If a link is loaded excessively by the inflexible traffic, it would cause more delay and some of the postponement restrictions of elastic traffic may not be eligible. Based on PID mechanism, priority dropping AQM algorithm (PID_PD) [6]provides service for the different layers based on their priority. Simulation outcomes shows that the proposed model offers good delivery ratio, better fairness and reduced postponement.

Pham and Perreau*et al*. [2003]proposed a mechanism based on load balance. This mechanism senses the traffic from a centre of the network, using a routing metric that takes considers degree of importance of the node for both proactive and reactive routing protocols. Load distribution is progressed in their proposed method which enhances the performance of the network in terms of dependability and average end to end delay. Their method used single path routing[7]. High node mobility and frequent route breaks could cause extra overhead which is observed in the proposed method.

Muhammad Aamir*et al*.[2013] propounds a new buffer management scheme in which it was proposed to provide efficient buffer management by employing a central interactive node[8]. Through an AQM strategy, buffer space is assigned dynamically to the nodes based on the packet arrival rate from neighbours. This mechanism is to have more control on packet drop probabilities. The suggested algorithm is run on the occurrence of a chosen incident, and the allocation is adjusted dynamically according to the share of neighbours in the buffer of the node gap buffer space allotted and the space occupied.

Mr. A. Chandra *etal*. [2014] introduced a chore packet mechanism to send the feedback to the sender. It observed that traffic overhead is more based on certainadditional actions involved in the methodology. Also, a considerable amount of buffer space[9] gets wasted to maintain virtual queue.

K.Dinesh Kumar, I.Ramya&M.RobertsMasillamani proposed Predictive Queue Management for MANETs usingPAQMAN, the AQM scheme which the author describes as a lightweightscheme that requires 6 multiplications every 0.02 seconds, proactive (queue is managed by anticipating the future). No prior knowledge of traffic model is required. Therefore, this suits ideally for MANETs. PAQMAN [10]reduces PLR and increases efficiency of transmission, with computational overhead being negligible. This predictive model uses Sampling Interval (SI) and Prediction Interval (PI). Average sampled queue length is calculated at the beginning of each PI, which is then used to predict to the average queue length using RLS (Recursive Least Squares). These values are used to compute PDP (packet drop probability). PDP decides whether to drop an incoming packet. Performance metrics in terms of PLR, Retransmit fraction and PDR are increased in this method.

Dinesh Gupta, *etal*. [2015] proposed a methodology called Dynamic Queue and TCP based Multipath Congestion Control Scheme [11]to minimize PDR by the selection of base rate through estimation of delay in acknowledgement by using a dynamic queue. Wired and wireless communication parameters are set initially and the best possible multiple path from source to destination is selected for data transmission. TCP New Reno updated technique is applied to calculate difference of delay in acknowledgement. Once data size is then set, Dynamic Queue Scheme is applied if incoming rate is faster than the outgoing rate to minimize PLR. Sender is alerted to lower the data rate via TCP New Reno. But this model is aimed at controlling the congestion in a wired network.

EssamNatsheh*et al*.[2007] proposed Fuzzy Active Queue Management for Congestion Control in Wireless Ad-hocNetworks wherein current queue size is used to estimate the probability of dropping of incoming packets. PDP (packet drop probability) is calculated based on estimated load factor and propagation delay. This calculation is embedded with fuzzy logic. Different fuzzy systems are constructed using Fuzzification, Inference and Defuzzification,[12]processes among which Mamdani method is used in the proposed method. A novel AQM algorithm (Fuzzy-AQM) is finally suggested to achieve high queue utilization and low PLR. From simulation results, Fuzzy AQM announces less routing overhead, low delay and less average packet loss ratios compared to other AQM policies.

PROPOSED MODEL

The proposed model (I.P.Q.S) is based on the idea derived by an in-depth analysis of the design approach and performance metrics of various existing congestion control models. It is observed that the design approach is the factor which drives the flow of the model and is the reason for the fact that it favours only a subset of performance metrics due to its approach but not the entire set thereby leaving a scope for further research works on the topic.

I.P.Q.S aims at achieving the optimum results in terms of all the major performance metrics like network throughput, packet loss rate, energy efficiency, minimal delay, best transmission rate, optimal utilization of network resources like link capacity etc. Many of the existing protocols tend to compete for resources available in the shortest path that are found during its route discovery process. A drawback here emerges out of the fact that it is an inherent characteristic of packet switched networks that the data is bursty in nature and is found to be the major factor contributing to 60% of congestion occurring scenarios while link failure and other reasons may constitute the remaining 40% of congestion events. In a typical transmission plan, the next hop of a particular node is flooded with data at an exponential growth until source reaches the knee point of its transmission rate. This phenomenon alone, doesn't seem to be a scenario that could lead to a congestion if all the neighbouring nodes are idle without transmitting any data. But this is not a valid scenario in MANETs. Because it is a productivethought to expect that at any given point of time, at least 50% of nodes have some data to transmit either getting originating from or is being routed through each node. This makes it clear that there is a fair amount of probability that the next hop for any node always renders its services for multiple nodes and not only to a single node.

I.P.Q.S overcomes this situation with its efficient queueing algorithm for which the key parameter is Average RoT (Rate of Transmission). Figure 5.1 below shows a typical IPQM design in high level.

Architecture of I.P.Q.S

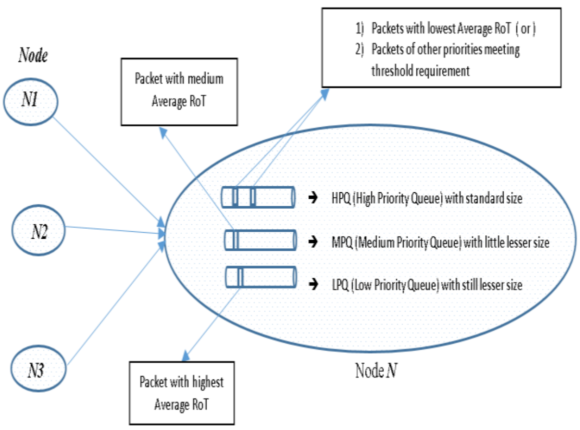

Figure 5.1Architecture of I.P.Q.S

The figure 5.1 abovedepicts the queueing scheme of Node *N* in the perspective of its routing functionality. Node *N* contains three queues with priorities High, Medium and Low which are intended to hold the packets with rate of transmission being low, medium and high respectively.I.P.Q.S algorithm is run usually for each packet arrived whilst periodicity of its each run is configurable. At each interval, this algorithm re-calculates the current rate of transmission of the route and determines the queue which the packet should get inserted into. The idea behind recalculating the RoT for each packet arrival is that,transmission rate on a particular route is not guaranteed to be constant due to a lot of inherent constraints in MANET environment like link failure, node failure, fluctuations in signal, noise,congestion at the next hop, and channel collisions due to drastically changing weather conditions, etc. These are some of the valid reasons to imagine an inconstant RoT on any route in MANET environment. As the main objective of I.P.Q.S is to reduce the congestion, it is achieved by not holding packets in the queue that really don't require more waiting time while retaining the packets that carry a comparatively less current RoT. This helps keeping usable buffer space to the maximum extent possible for accommodating subsequent packets arriving at the node thereby increasing the PDR to achieve overall network throughput with reduced congestion.A pseudo code which gives an overview of the algorithm is given below.

A pseudo code implementing I.P.Q.S

foreach packet arrived

```
if ( GetAverageRoT(RouteOfThePacket) = "High" )
{
if (IsSpaceAvailableInHighPriorityQueue == true )
{
   push_the_packet_into_HighPriorityQueue();
   If ( BufferUsageReachingQueueCapacity == true )
      alertSenderToReducePacketSendingRate();
}
elseif (IsSpaceAvailableInMediumPriorityQueue == true )
   push_the_packet_into_MediumPriorityQueue();
elseif (IsSpaceAvailableInLowPriorityQueue == true )
   push_the_packet_into_LowPriorityQueue();
else
   dropCurrentPacket();
}
elseif ( GetAverageRoT(RouteOfThePacket) = "Medium" )
{
if (IsMediumToHighThresholdReached == false )
push_the_packet_into_HighPriorityQueue();
elseif (IsSpaceAvailableInMediumPriorityQueue == true )
  {
push_the_packet_into_MediumPriorityQueue;
   If ( BufferUsageReachingQueueCapacity == true )
      alertSenderToReducePacketSendingRate();
  }
elseif (IsSpaceAvailableInLowPriorityQueue == true )
push_the_packet_into_LowPriorityQueue();
elseif (IsSpaceAvailableInHighPriorityQueue)
push_the_packet_into_HighPriorityQueue();
else
   dropCurrentPacket();
}
elseif ( GetAverageRoT(RouteOfThePacket) = "Low" )
{
if (LowTo_HighThresholdReached == false )
push_the_packet_into_HighPriorityQueue();
if (LowTo_MediumThresholdReached == false )
push_the_packet_into_MediumPriorityQueue();
elseif (IsSpaceAvailableInLowPriorityQueue == true )
  {
push_the_packet_into_LowPriorityQueue();
   If ( BufferUsageReachingQueueCapacity == true )
      alertSenderToReducePacketSendingRate();\
  }
elseif (IsSpaceAvailableInHighPriorityQueue == true )
push_the_packet_into_HighPriorityQueue();
elseif  (IsSpaceAvailableInMediumPriorityQueue == true )
         push_the_packet_into_MediumPriorityQueue();
else
         dropCurrentPacket();
}
elseif ( GetAverageRoT(RouteOfThePacket) = '\0' )  /* Null */
{
if (IsSpaceAvailableInHighPriorityQueue == true )
   push_the_packet_into_HighPriorityQueue();
elseif (IsSpaceAvailableInMediumPriorityQueue == true )
push_the_packet_into_MediumPriorityQueue();
else (IsSpaceAvailableInLowPriorityQueue == true )
   push_the_packet_into_LowPriorityQueue();
else
dropCurrentPacket();
}
```

How does the above Queueing Model help?

The efficiency of the algorithm lies in its fair queueing methodology wherein each queue is utilized to the best extent possible while not restricting medium and low priority packets to always be bound to the respective queues. Even a medium priority packet gets its turn to enter into HPQ and a lower priority packet also gets its turn to take advantage of MPQ and HPQ based on certain pre-defined threshold values for the movement of packets from *medium-to-high, low-to-high and low-to-medium* as illustrated below.

High Priority Packet

A high priority packet is first tested for the possibility of getting inserted into HPQ (High Priority Queue). If the space is available in HPQ, it gets inserted. If HPQ is full, the packet is not dropped. Instead, if MPQ has space, it gets into MPQ. If MPQ (Medium Priority Queue) is also full, and LPQ (Low Priority Queue) has space, packet is inserted into LPQ. If all the queues are full, the packet is dropped.

Medium Priority Packet

When a low priority packet is arrived at a node, it is not directly moved to MPQ. Instead, usage of HPQ is tested and if found to be below the threshold value of medium-to-high, packet gets its room in HPQ. If usage of HPQ is beyond this threshold value and space is available in MPQ, then the packet is inserted into MPQ. If MPQ is also full and LPQ has space, the packet is inserted into LPQ. In the case LPQ is also full, the packet is not dropped. Instead, if HPQ has space though beyond medium-to-high threshold value, still the packet gets inserted into HPQ. If all the queues are full, then the packet is dropped.

Low Priority Packet

When a low priority packet is arrived, it doesn't directly get into LPQ. Instead, if usage of HPQ is below *low-to-high* threshold value, packet is inserted into HPQ. If usage is beyond this threshold and usage of MPQ is below low-to-medium, the packet enters MPQ. If usage of MPQ is beyond this threshold value and LPQ has space, the packet enters into LPQ. If LPQ is full and HPQ has space irrespective of any threshold value, the packet gets into HPQ. If HPQ has no space and MPQ has space even beyond any threshold value, still the packet goes into MPQ. If all the queues are full, the packet is dropped.

A noticeable point here is that, while queue priorities are maintained, still the packets of all categories are allowed to gradually change their priority in anticipation of benefiting further improvements if any in their respective links thereby permanently change their priority with improved subsequent RoTs. Also, if a link speed graduallygoes down due to any reason, the route's priority is automatically changed by an algorithm which runs at a regular interval. This approach makes use of all kind of queues very efficiently and minimizes the wastage of vital resources at node level thereby minimizing the PLR, end-to-end delay and increases the overall network throughput.

Advantages of the model

1) This queueing model can be easily integrated with many of the existing communication protocols with very minimal changes.

2) A generic model with a high performance algorithm which is fully customizable and can be used to introduce queues with other categories to accommodate application level priorities.

3) Congestion due to flooded data and Link failure need not be addressed separately. For instance, a Lowest Priority Queue can be introduced with a small size and packets of which are made never to enter queues of other category. Later, examining the data transmission rate at this queue helps detecting link failures and with an explicit message, energy involved in re-transmitting the packets at the source can be minimized.

4) Minimized end-to-end delay would have direct impact on increased throughput which in turn helps reducing PLR and finally achieves reduced congestion.

5) This model also answers the question 'why can't buffer size at the node be increased to reduce PLR'. The straight answer for this question is that, an increased buffer impacts the waiting time of the packets staying at the end of the queue. This would result in time outs for those packets and also trigger retransmission of such packets. But adding more buffer size is achievable in the I.P.Q.Sby way of distributing this extra buffer among queues with various other priorities.

Sizing of the Queue to alleviate Packet Loss Rate

We shall now see the advantage of slight difference found in sizes of the queues. Though the difference between Small-To-Medium and Medium-To-Large is very small, the purpose behind the concept is to gain the a little more efficiency in the queueing model. This is based on the fact that packets belonging to the lowest RoT category usually do not need a longer waiting time and hence a queue with comparatively a little bigger size could still be utilized to accommodate more packets whereas a small queue is ideal for packets with less RoT because lengthy queues impose an additional waiting time for the packets which are already carrying a less priority. Further, a provisionis made for the packets of all categories to improve their RoT by utilizing HPQ and MPQ during the times when threshold limits are satisfied. As the sendersfalling in all categories are informed well in advance to reduce transmission rates, packet loss ratio is tried to be reduced to the maximum extent possible.

Average RoT & Cross Queue Thresholds

How does an intermediate node get RoT info of the packets flowing on a particular route? This is achieved by making use of some bits in packet header to provide RoT info. It could be derived using different mathematical equations. A basic equation may look like equation number given shown and can be easily extended to consider various other parameters that can impact the network scheduling at the node.

1)
$$\text{Average RoT} = \frac{\text{RTT of the Route}}{\text{Number of hops from Source to Destination}}$$

2)
$$\text{Average RoT} = \frac{\text{RTT of the Route}}{\text{No. of hops on the Route}} \times \frac{\text{Bandwidth Standard}}{\text{Average Bandwidth of Route}}$$

3)
$$\text{Average RoT} = \frac{\text{RTT of the Route}}{\text{No. of hops}} \times \frac{\text{Bandwidth Standard}}{\text{Average Bandwidth}} - \text{Average Queuing Delay}$$

Normalization of variations in Bandwidth

While calculating average bandwidth, it is a possible scenario that link speed may vary between different pairs of hops on a particular route as seen in the figure 5.0.

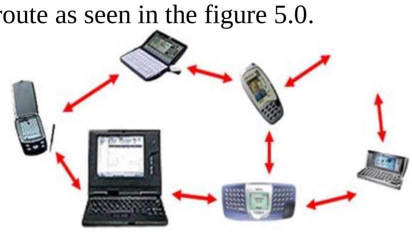

Figure 5.0. Devices with different capacities

In this case, to calculate RoT correctly, these varying speeds must be generalized into a uniform measure of transmission rate.

In the series of the above equations it is apparent that the equation could be easily extended to include other factors that may affect RTT of a packet such as Signal Strength, Signal-to-noise-Ratio and Total distance from source to destination (as number of hops is considered) etc. And finally, the choice of selecting the formula could be made available as a configurable parameter of the node to meet different kind of QoS policies.

Periodicity of refreshing priorities

A QoS Policy may drive the periodicity or interval of the refreshing activity happening at node level to recalculate RoT of the routes which is usually calculated for each packetarrival. For example, A QoS policy itself could be influenced by different factors like considering priorities based on type of application, weather conditions involved etc.

V. SIMULATIONRESULTS

5.1 Simulation Environment

The proposed model I.P.Q.Sis simulated using NS2 with channel capacity of mobile nodes as 2 Mbps. Wireless standard used is IEEE 802.11. In the simulation, mobile nodes move in a 750 meter x 500 meter region for 60 seconds of time. The number of mobile nodes range from 50 to 350. It is assumed that each independent node moves in random direction with varying speeds but average speed being the same Transmission range of all nodes is 300 meters with simulation speed being 20m/s with Constant Bit Rate. Pause Time of node is set at 20-120 sec. Queue sizes we employed is 220 packets for HPQ, 200 for MPQ and 180 for LPQ. Cross Queue thresholds being 50% for medium-to-High, 30% for low-to-high Threshold of Medium and 50% for low-to-medium.

5.2. Performance Metrics

The simulation statistics are collected to evaluate the performance of the proposed model. The results plotted belowshow the network throughput, packet delivery ratio, packet loss rate, average queueing delay etc.

5.2.1 Network Throughput

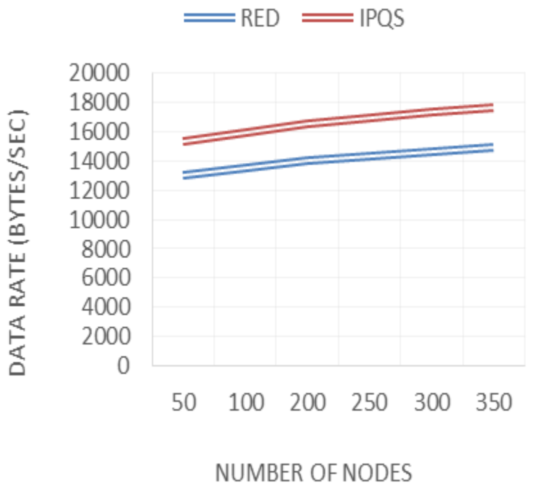

Fig.5.1 Throughput for varying number of Nodes

Throughput tells the number of packets received successfully at the destination which is usually measured asBytes/Sec. The throughput for RED and I.P.Q.Sfor varying number of nodes is plotted and compared in figure 5.1 above and the I.P.Q.S is found to have performed better. Increment in the number of nodes is leads to increased flow of packets to the nodes. From the figure 5.1, it can be observed that I.P.Q.Sis able to provide more throughput when compared to RED.

5.2.2 Packet Delivery Ratio

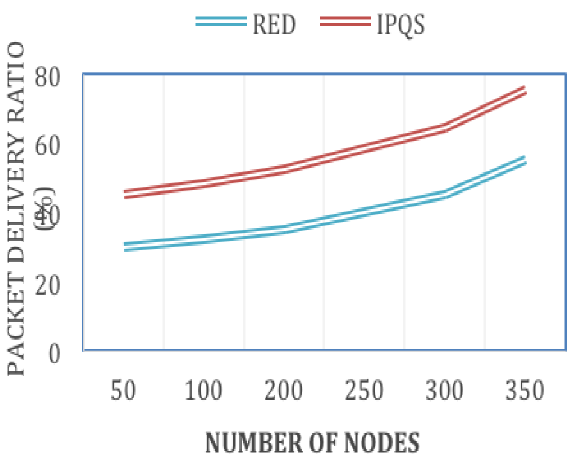

Fig.5.2 Packet Delivery Ratio Vs Number of Nodes

Packet delivery ratio is derived by dividing the number of packets received at the destinationwith the number of packets transmitted at the source. The percentage of successful packet delivery rate reflects the throughput of the network and is inversely proportional to the packet loss rate. It also reflects the efficiency of the chosen methodology for controlling the congestion. PDR for RED and I.P.Q.S, is plotted above in figure.5.1, for the increasing number of nodes.

5.2.3 Average Delay Analysis

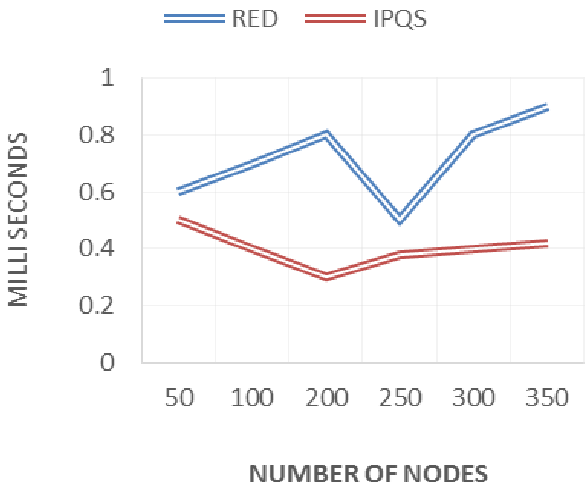

Fig.5.3 Average Delay

Figure 5.3 above shows the noticeable decrement in average delay when compared to the RED. Despite the increment in the number of nodes, the response time in the I.P.Q.S is very minimal which reinstates the efficiency of the algorithm implementing the queueing model with queues of different priorities but with a functionality of still allowing packets of all the categories when threshold limits are met.

5.2.4 Packet Loss Analysis

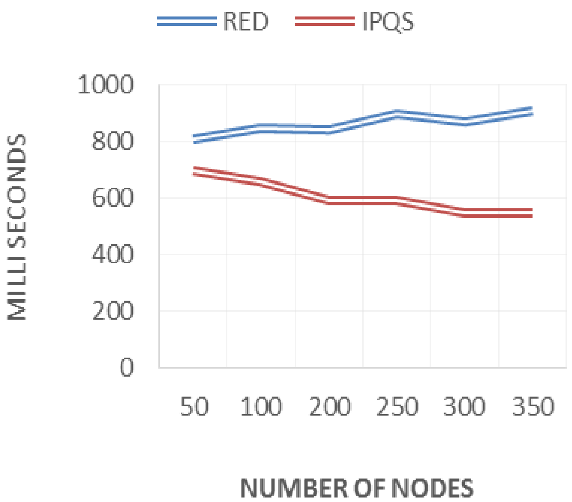

Fig.5.4 Packet Loss Ratio

The performance of the algorithm is clearly seen in the figure 5.4 plotted above. Unlike DropTail and other algorithms, I.P.Q.Sinvents a new way of reducing packet loss ratio. This is made possible by two ways firstly by maintaining cross queue thresholds to provide a possibility for low priority packets to consume high priority queues thereby slowly increasing their RoT and secondly by alerting senders on a certain queue threshold points to reduce the transmission rate to help reduce the PLR.

VI. Conclusion

Due to the fact that nodes in MANET operate under various constraints like less energy, limited buffer, less bandwidth, node mobility and thereby changing network topology, no single congestion control model is good enough to completely avoid the congestion. Hence, it is necessary that congestion must be avoided from very basic level of communication system. In this paper, we presented a very efficient queueing scheme which is for controlling the congestion at each node. The algorithm is very generic and robust. Simulation results shows less PLR, good Data Rate and less end-to-end Delay compared to RED, FIFO, WFQ and Drop Tail. This queue management scheme can also be used along with many other existing protocols for getting added advantage, high performance and to implement various QoS policies.

References

[1] K. R. S. Floyd and D. Black, The addition of explicit congestion notification (ECN) to IP, RFC 3168, Sep.2001.

[2]Arsh Arora1, Lekha Bhambhu2, Jan Nayak Ch., Sirsa Haryana-India, Evaluation of Active Queue Management Algorithms, 4, Jul-Aug 2014.

[3] L. Chrost and A. Chydzinski, The evaluation of the active queue management mechanisms, in Proc. 1st Int. Conf. On Evolving Internet (INTERNET), 2009, pp. 113-118.

[4] SanjeewaAthuraliya Victor H. Li Steven H. Low Qinghe Yin, REM: Active Queue Management, January 15, 2001.

[5] S. Kunniyur and R. Srikant. A time-scale decomposition approach to adaptive ECN marking. In To be published in the Proceedings of INFOCOM 2001, Alaska,Anchorage,April 2001.

[6] Ambika, I., Sadasivam, V.P. & Eswaran, An effective queueing architecture for elastic and inelastic traffic with different dropping precedence in MANET.

[7] Prachi Jain, Vijay Prakash, Architecture Of New Buffer Management Scheme For Controlling Congestion In MANET, ISSN: 2277-9655, [Jain*, 4.(10): October, 2015].

[8] M. Aamir, M. Zaidi, and H. Mansoor, Performance analysis of Diffserv based quality of service in amultimedia wired network and VPN effect using OPNET, International Journal of Computer Science Issues, Vol. 9, no. 3, pp. 368-376, 2012.

[9] Mr. A. Chandra, Ms. T. Kavitha “Adaptive Virtual Queue with Choke Packets for Congestion Control in MANETs” International Journal of Computer Networks and Wireless Communications (IJCNWC), ISSN: 2250-3501 Vol.4, No2, April 2014.

[10] Mr. Dinesh Guptam and Mr. Deepak Singh Tomar, Dynamic Queue and TCP based Multipath Congestion Control Scheme for Wired Network, International Journal of Computer Applications (0975 – 8887) Volume 123 – No.10, August 2015.

[11]K.Dinesh Kumar, I.Ramya&M.RobertsMasillamani, 2010 IEEE/ACM International Conference on Green Computing and Communications & 2010 IEEE/ACM International Conference on Cyber, Physical and Social Computing , Queue Management In Mobile Ad-hoc Networks(Manets).

[12] EssamNatsheh, Adznan B. Jantan, SabiraKhatun, and ShamalaSubramaniam, Fuzzy Active Queue Management for Congestion Control in Wireless Ad-Hoc, The International Arab Journal of Information Technology, Vol. 4, No. 1, January 2007.

[13] Y.-F. Guo, G.-S. Kuo, A packet scheduling framework for multipath routing in mobile ad hoc networks, in Vehicular Technology Conference (IEEE, 2007), pp.

[14] S.T. Hasson, E. Fadil, Queueing approach to model the MANETs performance. Br. J. Sci. 6, 18–24 (2012).

[15] Qingming Ma, K. K. Ramakrishnan, Queue Management for Explicit Rate Based Congestion Control.

Authors' Profile

M.Rajeswari received Msc(C.S) degree from S.K.University. Presently pursing Doctorate Degree (Ph.D) in Computer Science from Rayalaseema University, Kurnool and working as a Teaching Assistant in Computer Science Department in Rayalaseema University.Research interest includes computer networks and MANETs.

Dr.N.Kasiviswanath received his B.E.degree in Computer Science & Engineering from MarthwadaUniversity,M.S. degree in Computer Science & Engineering From BITS,Pilani and Ph.D degree from RayalaseemaUniversity,Kurnool.He has 24 years of teaching experience. He has published 30 research papers in National/International journals/conferences. At present, he is working as Professor and Head of Computer Science & Engineering Department, at GPREC, Kurnool.