Congestion Control in Wireless Networks Using AQM Algorithms

C.MerlinLagnes¹, J.Benimol², Mrs. D.C. Joy Winnie Wise³

¹,² U.G Student, B.E CSE, Alpha college of Engg, Chennai, T.N, India.
³Assoc. prof, Dept of CSE, Alpha college of Engg, Chennai, T.N, India.

Abstract—In a wireless environment packet loss does not always mean congestion, it can be due to sudden drop in channel quality. TCPs congestion control becomes insufficient in wireless environment, the work of TCP ends in an unnecessary reduction of end-to-end throughput which results in suboptimal performance. Hence the sender has to be made aware that some of the losses occurred is not due to congestion, through some feedback mechanism. The Active Queue Management algorithms (AQM) are used to reduce congestion, Here we analyze two AQM algorithms, Both make use of feedback mechanisms. Queue Management Congestion Control Algorithm (QMCCA) and its advance version Extended Queue Management Congestion Control Algorithm (EQMCCA). QMCCA uses ISQ notification and CE bit when the average queue size crosses minimum threshold value. The reverse ISQ traffic is reduced using EQMCCA, by introducing a configurable intermediate threshold value IntThres between minimum and maximum threshold values. The ISQ signals are generated only if the congestion crosses the intermediate threshold value. The feedback will reach the source with out any loss, since CE bit is also set in the packets once congestion occurs. Here we compare in terms of Gain in Goodput, Loss percentage and ISQ traffic in wireless environment. The performance of EQMCCA is almost equal to that of QMCCA and the ISQ traffic in the reverse direction is reduced.

Index Terms— Active Queue Management, Congestion Control, EQMCCA, ISQ Reverse Traffic, QMCCA

1. INTRODUCTION

The Wireless networks has led to heavy utilization and congestion because of its growing popularity. In heavily utilized wireless networks, the wireless portion of the network is the major performance bottleneck. Due to high density of nodes with in a single collision domain of IEEE 802.11 Wireless network can result in congestion. Thereby it causes severe performance bottleneck. Drastic drops in network throughput, unacceptable packet delays, and session disruptions are the effects of congestion. Communication over wireless links is characterized by sporadic high bit rates, failure of nodes, links and intermittent loss of connectivity due to handoffs, significant throughput degradation and very high interactive delays. In wireless networks TCPs congestion control mechanism is deficient, It is because packet loss does not always means congestion but it can be due to sudden drop in channel quality. The reduction in congestion window, which occurs quite frequently in a wireless network can result in poor performance in terms of throughput and delay. When packets are lost for reasons other than congestion, these measures results in unnecessary reduction in end-to-end throughput and in suboptimal performance. Therefore the sender has to be made aware that some of the loss that occurred are not due to congestion. The quality of service (QOS) has become very important part in networking. There is a great demand of high bandwidth and low latency traffic from sender to receiver, to provide such kind of service is not always possible because of low speed links at the wireless fusion point of the interconnection that passes the packets. It becomes very important to note how we put different traffic in queue in different situation its because some traffic needs to pass quickly and some needs greater bandwidth. The main difficulty is to send large amount of data from one end to other end in a network when congestion occurs [1]. The competition of flows for obtaining network resources increases in case of congestion [2]. Due to poorly designed protocol, a network may be very much congested so that no data could effectively be transferred, or a communication link that uses only a tiny fraction of bandwidth maybe available at the bottleneck node. For avoiding similar pitfalls in protocol design extensive analysis and testing plays a crucial role. Internet is designed to

ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 2, Issue 3, March 2013
All Rights Reserved © 2013 IJARECE
allow for communication at reasonable rates. If resources are over utilized congestion will occur. Congestion occurs when there is more incoming traffic than network nodes can handle. The result of congestion is full queues, long delays, and packet drops have an adverse effect in throughput. If there is a reduction at the rate at which packets are send throughput may be increased [3]. When the demand on the network resources is greater than the available resources and due to increasing mismatch in link speeds caused by intermixing of heterogeneous network technologies, congestion control problem occurs. With in a large buffer space, the congestion problem cannot be solved. The window will time out, if packets stay too long in the buffers this results in congestion and tcp has to retransmit the packet. It is clear that too much traffic will lead to a buffer overflow, high packet loss and large queueing delay. The high speed link connected via the high speed switch with the low speed links will cause congestion at the wireless fusion point of interconnection. Hence the congestion problem cannot be solved by high-speed links or with high Speed processor. The high link utilization is lead by the usage of queue management mechanism in an efficient way and it avoids congestion collapse [4]. This paper is organized as follows, session 2 deals with Works of AQM in wireless networks. Two congestion control algorithms QMCCA and EQMCCA based on QOS are discussed in brief in session 3. Parameters used, simulator, simulation scenario and topology are discussed in session 4. Simulation results of the two algorithms are given in session 5. Session 6 concludes the paper.

2. WORKS OF AQM IN WIRELESS NETWORKS

In wired networks Active Queue Management(AQM) is most commonly used, principally in backbone routers where packet loss is due to network congestion [5]. In providing Quality of Service (QOS) the queue management algorithm applied to router plays a major role [6]. Queue management algorithms in IP routers determine which packet has to be deleted, when necessary [7]. The use of AQM in wireless networks is studied in [8]. The design of an AQM ensuring the stability of the congestion phenomenon at the router is proposed in [9]. It was recognized that TCP with an AQM router can be considered as the feedback control system. Thus the control theoretic models of AQM have been developed in [10], [11], [12] this has helped in better understanding of router queue dynamics in TCP networks. This has led to systematic and scalable techniques for controller design refer [13] and [14]. With respect to AQM in wireless links is studied in [15]. Wired link has fixed capacity were as a wireless link has capacity that is time varying due to multipath fading and mobility. The access point in wireless networks acts as a bridge between wireless and wired networks. The actual available bandwidth in wireless network is much smaller than in wired networks, there is a bandwidth disparity in channel capacity that makes the access point a significant network congestion point. Future heterogeneous wireless environments will characterize the coexistence of a large variety of wireless access technologies with different protocol stacks and supporting a number of applications and services with different QOS requirements to be provisioned to terminals with different modes of multimode capabilities to access the available networks [16]. "Mark Front" which can lead to a fast congestion feedback was proposed by Liu and Jain [17], a router will mark the packet which is at the head of the queue, when incipient congestion is detected. It is because such a packet is to be send out first, the feedback time of the mark front strategy is considerably shorter than that of the mark tail method [18]. An algorithm called “New ECN ”, was suggested by Hamann and Walrand [19] which works by preventing a fast connection from opening its congestion window too quickly, but instead enabling a slow connection to open its window more aggressively. The congestion window size is decreased also the rate of congestion window increased is modified, when the marked packet is echoed to the sender. In [20], Xu et al presented an AQM scheme for WLAN, the performance analysis of the proposed AQM scheme is limited, its because the authors only consider the delay from the wired network to WLAN. On the other hand fairness and percentage of loss are accepted as more important metrics in evaluating an AQM performance. The comparative analysis of different versions of TCPs, particularly TCP Veno and TCP Reno under RED and Tail-Drop(TD) queue is focused by Pang et al in [21]. In [22] TCP enhancement for wireless networks is done. Random Early Detection [23] follows the concept of Early Random Drop by introducing the measure of average queue size and also dynamically changing the drop probability. The use of dynamic drop probability ensures that the gateway reacts differently to different level of congestion expectation. The instantaneous queue sizes are not a correct indicator of congestion due to burst nature of internet traffic. Average queue size measured over all times is used by RED. The RED gateway usually finds the average queue size as exponentially weighted moving average of instantaneous queue size. The average queue size is
compared with minimum threshold and maximum threshold. When the average queue size is between minimum threshold and minimum threshold all the arriving packets are marked. When the mechanism ensures that the marked packets are dropped, or if all source nodes are co-operative, then the average queue size does not exceed the maximum threshold. The WECN algorithm proposed by Peng et al[24]. Which is taken for analysis. In WECN, due to buffer overflow in the router when the first packet loss is noted, the CE bit in the header of the passing packets in the queue are marked. By this way the sender is indicated that the packet loss has occurred due to network congestion and not because of random loss due to degraded channel quality. This mechanism helps in clearly analyzing that the packet loss has occurred due to congestion only. The TCP responds by reducing the congestion window accordingly. For reasons other than congestion the packets are not marked, the TCP times out and retransmits the packets. It reduces the window but the recovery time will be much faster compared to that of a loss that occurred due to congestion. The unnecessary reduction of window due to packet loss for reasons other than congestion is avoided here.

3. ANALYSIS OF ALGORITHMS

This session describes about QMCCA and EQMCCA algorithms that are taken for analysis.

3.1.QMCCA

Queue Management Congestion Control Algorithm (QMCCA) [25] is an alternate approach to WECN algorithm. It makes use of existing Internet Control Message Protocol(ICMP) Source Quench(ISQ) signaling mechanism. The congestion notification is kept at the IP level. From the point where the congestion occurs the ISQ signal is generated and send to the source. The reaction time to congestion is reduced by this approach. The information about the severity of the congestion is included in the ISQ message. The end host responds by reducing the congestion window thereby leading to maximum utilization of network and resources [26]. This is termed as the Multilevel Explicit Congestion Notification (MECN). The ISQ messages are generated from the period between initial congestion detection and until the source end system adjusts, since the back path from the point of congestion to the source is not congested the ISQ messages can reach the source with out any complication.

The algorithm for QMCCA is as given below.

For every packet arrived
    calculate the average queue size(\(avg\))
    if\(\text{minth}<\text{avg}<\text{maxth}\)
    \{
    Calculate probability \(\text{pa}\)
    With probability \(\text{pa}\):
    if ECT bit is set and
    if the packet is not already marked
    \{
    Mark the arriving packet
    Send an ISQ to the sender
    \}
    else
    Drop the arriving packet
    \}
 else if \(\text{maxth}\leq\text{avg}\)
 Drop the arriving packet
else if \(\text{maxth}\leq\text{avg}\)
 Drop the arriving packet
 Send an ISQ to the source.

Fig.1. gives the flowchart of the algorithm Queue Management Congestion Control Algorithm(QMCCA)
A QMCCA capable router implements the Active Queue management algorithm RED in it and hence it requires few modifications. CE and ECT both the bits are maintained as in ECN and is forwarded to the transport layer when the IP message is demultiplexed. If the average queue size goes between the minimum threshold and maximum threshold for the incoming packet, the RED probability may choose the packet to be marked. If the ECT bit of this packet is set and if it is not already marked (i.e. the CE bit is not set) then mark the packet (i.e. set the CE bit) and send back an ISQ message to the source. If RED chooses the packet and the ECT is not set, then the packet is dropped. ISQs are not generated, even if congestion occurs incase if the packets are marked previously by another router. If the packet has the ECT bit set and it has been selected to be dropped, then ISQ is generated even if the packet is marked earlier by some other router. Upon identifying the flow causing the congestion, the source reacts by halving both the congestion window and the slow threshold value for that flow. The following are the advantages of QMCCA with respect to the drawbacks in ECN. QMCCA uses existing network layer signaling the source quench mechanisms to notify the transport layer of congestion problems. Hence all IP transport protocols are notified in the same manner about network congestion. It does not require the use of any transport layer protocol for congestion notification. It is therefore protocol independent and can be used by other transport protocols such as UDP. Also, there may be value in providing a common mechanism for notifying all transport protocols about congestion. QMCCA has the router send the congestion notification as soon as congestion occur in the router to the sender. MECN is used in QMCCA since the network resources can be more effectively be used, because the feedback gives indication of the congestion level at the overloaded point in the network. When the congestion is between the thresholds packets are marked and not dropped. Lower loss rates, reduction in number of TCP timeouts and retransmissions, faster congestion notification, and lower packet delay are the advantages of such method.

3.2. EQMCCA

Both ECN and BECN mechanism are used in QMCCA. It generates ISQ and also sets the CE Bits when congestion occurs. In case of the lighter congestion, there is an overhead in the generation of ISQ messages. Due to the overhead required for generating ISQ, the performance of internet routers are affected. The control plane of the routers will generate the ISQ. It is compared to be slower than the data plane. For the high speed routers, the transfer of packets between the control plane and data plane should be reduced ECN notifications are more reliable than ISQ notifications.

When the destination receives the ECN notification, The Destination generates ECN Echo acknowledgements continuously when it receives ECN notification until the sender notifies the receiver. In that case the sender will activate the CWR (Congestion Window Reduced) Mechanism and reduce both the congestion window and the threshold value. The congestion notifications cannot be reliably delivered since the source have not detected the loss of the ISQ.

When ISQs are sent to the source additional reverse traffic will be introduced. It is preferable to minimize such control traffic.

On the other hand the RTT taken by the CE mechanism will lead to delay of notifying the congestion which cause higher queue variance and reduced throughput.

EQMCCA tries to solve these issues by using CE mechanism and ISQ for high congestion and CE for low congestion. High and low congestions can be differentiated by choosing an intermediate value between the min and max values of RED called IntThres (Intermediate Thresh). IntThres can be set anywhere between min and max values and it is configurable. The router behaves as an QMCCA when the queue is above IntThres and as ECN when the queue is below it. Whichever notification reaches the sender first should cause a window reduction.

In order to establish ECN capability, the end hosts perform end to end negotiations initially. The sender should be respond to both ECE and ISQ notifications. The first packet to be send after window reduction will have Congestion Window Reduced (CWR) bit in order to stop the receiver from sending more ECE acknowledgements. Before the sender starts increasing the window, it waits for a full RTT.

The algorithm of EQMCCA is given in Fig.2.

For each packet arrival
Calculate the average queue size avg
If (avg > maxth)
{
    if (ECT bit marked in the chosen packet)
    {
        Drop the packet;
        Send an ISQ2 back to the source;
    }
    Else drop the packet;
}
else if (avg<minth )
queue the packet;

else if (minth<avg<IntThres)
{
    if (ECT bit marked in the chosen packet)
    {
        Mark the CE bit if not already marked;
        else
        Drop the packet;
    }
}
else if (ECT bit marked in the chosen packet)
{
    Mark the CE bit if not already marked;
    Send ISQ1 to the Source;
}
else
{
    Drop the Packet;
    Send ISQ2 to the source;
}

Fig 2. Flowchart for EQMCCA

4. SIMULATION SCENARIO

Using NS2 the Simulations for the two algorithms were analysed.

Fig 3. Wireless Network topology for simulations

Using a simple topology the model of the TCP network is given in Fig. 3. In Internet through a router TCP sources send data towards their destinations. In the network the Router represents the bottleneck node. By receiving the acknowledgement each source increases its transmission rate, and this would exceed the outgoing link capacity of the bottleneck node. The window size is reduced to half, when the TCP source detects a packet drop. In this way each source attempts to determine the available capacity in the network. The topology uses IEEE 802.11 Wireless Local Area Network (WLAN) scenario with channel data rate of 11 Mbps [27]. The wired network LAN or Internet with a link of capacity 100 Mbps is connected to the access point. There are TCP flows between the Internet and the wireless nodes; the precise number of flows is varied from 10 to 100 for different scenarios. Lmax = 500 packets, Lmin = 100 packets, IEEE 802.11 propagation delay =0.8 ms, nominal round trip delay=80ms, Probability based drop-ping pmax =0.2. We consider fixed length packets, each of 1000 bytes. The maximum window size of the TCP flows is kept at the default ns-2 value of 20 packets. At the access point (router) AQM is implemented. Due to fading, the channel between the wireless nodes and the access point is time varying. These capacity variations are performed using the model supported in ns-2 for wireless networks. The sampling time was taken as 40 ms.

The two algorithms QMCCA and EQMCCA were used for the analysis and the two QoS metrics used for the experimentations are Gain in goodput, Packet Loss and Reverse ISQ traffic. Based on the number of data packets received by the receiver the goodput for a TCP flow is computed.
The number of ACKs received by the sender within simulation time is used for the computation. Percentage loss measures the ratio of the number of packets dropped at the bottleneck link to the total number of packets injected into the bottleneck link for a particular flow or set of flows. Percentage of ISQ reverse traffic is computed as the number ISQs generated in the reverse direction of data flow as a ratio of the total number of packets in the reverse direction.

5. RESULT AND DISCUSSIONS

The results of the simulation are given as graphs below for the comparative study between the algorithms.

Fig. 4. Gain in Goodput percentage

Fig. 4. Shows the percentage of the goodput gain by varying the number of TCP flows. Due to early congestion notification by QMCCA, it has a gain of 35.5% compared to 28% of EQMCCA who generates ISQs only after it crosses the IntThres value. Fig. 5. shows the percentage of loss by varying the number of flows from 10 to 100. It is found that QMCCA algorithm has the minimum number of packets dropped and when the number of flows increased; there was a constant increase in the number of packets dropped. EQMCCA has more packet loss since RTT time taken for the feedback to reach the TCP source was high. Fig. 6. Shows the reverse traffic ISQ generated by both the algorithms. In QMCCA, there is a two way feedback one in the forward direction using the CE bit and other in the reverse direction using ISQ leading to reliability and fast reaction.

5. CONCLUSION

The characteristics of the wireless networks and the different AQM algorithms available in the wireless networks are discussed in this paper. A feedback mechanism is needed to inform the TCP whether the packet loss was due to congestion or not, due to the inconsistent nature of wireless networks. The TCP will respond by decreasing the window accordingly. The two QoS Congestion Control AQM algorithms namely QMCCA and EQMCCA are compared. The gain in goodput, percentage of loss and reverse ISQ traffic are the metrics in which comparisons are made. Significant performance difference between the algorithms are revealed through simulation results. QMCCA gave better results when concerned
about the loss percentage. The sender is informed about the packet loss in QMCCA using both ISQ and CE. The congestion can be rectified as early as possible since the senders could react quickly. The EQMCCA reaction time is high leading to more packet losses, since the RTT taken by EQMCCA is more. When the queue size reaches the intermediate threshold value ISQ gets generated. Hence it can be concluded that in an wireless environment the proposed QMCCA performs better than EQMCCA in terms of goodput and loss percentage.

REFERENCES

[18] Arash Dana and Ahmad Malekloo, "Performance Comparison between Active and


