http://ansinet.com/itj
ITJ
ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 5 (4): 689-697, 2006
ISSN 1812-5638

TCP Protocol and Red Gateway Supporting the QoS of Multimedia Transmission Over Wireless Networks

Mohammad Al Nabhan, Sufian Yousef and Jaafer AL-Saraireh
Department of Design and Communication Systems, Anglia Ruskin University, United Kingdom

Abstract: This study presents the adaptation of Random Early Detection (RED) Gateway as an active buffer management algorithm for multimedia traffic in Wireless networks. We studied and modified features of RED algorithm showing that an appropriate RED algorithm can improve multimedia data quality of service by making a better control during congestion. Also we present some enhancements on the TCP protocol for wireless networks adaptation. The main contribution in this research is to propose a new RED gateway scheme that will work efficiently in wireless environments.

Key words: RED, Multimedia, QoS, TCP, ECN

INTRODUCTION

The explosive increase in the volume and verity of multimedia traffic has placed a growing emphasis on the queuing management and congestion control in wireless networks. The future mobile networks are expected to merge most types of services including internet and multimedia applications based on Internet protocol toward all IP. We can see the explosive growth in the use of wireless computers equipped with wireless network interfaces conversion with each other using IP. Nevertheless, current TCP/IP protocols do not perform well in wireless environment because of mobile environment nature.

The QoS degradation in the wireless networks not only caused by the congestion, but due to high possible high Bit Error Rate (BER) of the radio channels, fading and interference between channels. Some suggestions are to use small packet sizes at the wireless path, which will reduce packet errors caused by bit errors. The packet size should be flexible, because BER is not stable factor in wireless networks. Similar to wired networks, appropriate buffer management strategy in wireless networks can manage the buffer efficiently, avoid or relieve congestion, so as to guarantee QoS. Because of unique issues in the wireless channel, strategies in wired networks can not apply directly to wireless network (Floyd *et al*., 1993).

Several previous works suggested using resource management combined with fair queuing disciplines. Other approaches are adaptive applications at application level. On the other side many new protocols such as Loss-Delay Adjustments Algorithm LDA (Sisalem and Schulzrinne, 1998), Rate adaptation protocol RAP (Loguinov and Radha, 2003) and TCP Friendly Rate Control Protocol TFRC (Padhye *et al*., 1998) have been developed for multimedia applications. While the principles behind RED gateways are fairly general and RED gateways can be useful in controlling the average queue size even in a network where the transport protocol can not be trusted to be cooperative, RED gateways are intended for a network where the transport protocol responds to congestion indications from the network. The gateway congestion control mechanism in RED gateways simplifies the congestion control job required of the transport protocol and should be applicable to transport layer congestion control mechanisms other than the current version of TCP, including protocols with rate-based rather than window-based flow control.

RED gateways can be useful with a range of packet-scheduling and packet-dropping algorithms. For example, RED congestion control mechanisms could be implemented in gateways with drop preference, where packets are marked as either essential or optional and optional packets are dropped first when the queue exceeds a certain size. Similarly, for a gateway with separate queues for real-time and non-real time traffic, for example, RED congestion control mechanisms could be applied to the queue for one of these traffic classes (Vu, 2000).

RED GATEWAYS GOALS AND GUIDELINES

This section summarizes some of the design goals and guidelines for RED gateways. The main goal is to

Corresponding Author: Mohammad Al Nabhan, Department of Design and Communication Systems, Anglia Ruskin University, Chelmsford, Essex, CM1 1SQ, United Kingdom Tel: +441245493131/3028

provide congestion avoidance by controlling the average queue size. Additional goals include the avoidance of global synchronization and of a bias against bursty traffic and the ability to maintain an upper bound on the average queue size even in the absence of cooperation from transport layer protocols. The first job of a congestion avoidance mechanism at the gateway is to detect incipient congestion as defined in by Jain and Ramakrishnan (1977) a congestion avoidance scheme maintains the network in a region of low delay and high throughput. The average queue size should be kept low, while fluctuations in the actual queue size should be allowed to accommodate bursty traffic and transient congestion. Because the gateway can monitor the size of the queue over time, the gateway is the appropriate agent to detect incipient congestion. Because the gateway has a unified view of the various sources contributing to this congestion, the gateway is also the appropriate agent to decide which sources to notify of this congestion (Vu, 2000).

In a network with connections with a range of roundtrip times, throughput requirements and delay sensitivities, the gateway is the most appropriate agent to determine the size and duration of short-lived bursts in queue size to be accommodated by the gateway. The gateway can do this by controlling the time constants used by the low-pass filter for computing the average queue size. The goal of the gateway is to detect incipient congestion that has persisted for a long time (several roundtrip times) (Vu *et al*., 2000).

The second job of a congestion avoidance gateway is to decide which connections to notify of congestion at the gateway. If congestion is detected before the gateway buffer is full, it is not necessary for the gateway to drop packets to notify sources of congestion. In this paper, we say that the gateway marks a packet and notifies the source to reduce the window for that connection. This marking and notification can consist of dropping a packet, setting a bit in a packet header, or some other method understood by the transport protocol. The current feedback mechanism in TCP/IP networks is for the gateway to drop packets (Jacobson, 1998; Floyd and Jacobson, 1994).

Networks contain connections with a range of burstiness and gateways such as Drop Tail and Random Drop gateways (Jacobson, 1998), have a bias against bursty traffic. With Drop Tail gateways, the more bursty the traffic from a particular connection, the more likely the gateway queue will overflow when packets from that connection arrive at the gateway. Another goal in deciding which connections to notify of congestion is to avoid the global synchronization that results from notifying all connections to reduce their windows at the same time. Global synchronization has been studied in networks with Drop Tail gateways (Zhang and Clark, 1990) and results in loss of throughput in the network. Synchronization as general network phenomena has been explored by Floyed and Jacobosn (1994). In order to avoid problems such as biases against bursty traffic and global synchronization, congestion avoidance gateways can use distinct algorithms for congestion detection and for deciding which connections to notify of this congestion.

The RED gateway uses randomization in choosing which arriving packets to mark; with this method, the probability of marking a packet from a particular connection is roughly proportional to that connection's share of the bandwidth through the gateway. This method can be efficiently implemented without maintaining per-connection state at the gateway (Condder *et al*., 2000). One goal for a congestion avoidance gateway is the ability to control the average queue size even in the absence of cooperating sources. This can be done if the gateway drops arriving packets when the average queue size exceeds some maximum threshold (rather than setting a bit in the packet header). This method could be used to control the average queue size even if most connections last less than a roundtrip time (as could occur with modified transport protocols in increasingly high speed networks) and even if connections fail to reduce their throughput in response to marked or dropped packets (Jacobson, 1998; Jain and Ramakrishnan, 1997).

THE RED ALGORITHM

The random early detection (RED) algorithm is becoming a de-facto standard for congestion avoidance in the internet and other packet switched networks, Braden *et al*. (1998) states RED should be used as a default mechanism for managing queues in routers unless there are good reasons to use another mechanism. To this end strong recommendation for testing, standardization and widespread deployment of active queue management in routers to improve the quality of service over wireless networks.

The RED algorithm calculates the average queue size, using a low pass filter with exponential weighted moving average (Vu, 2000). After computing the average queue size avg it computes the dropping probability p_b based on the instantaneous queue size and a weight factor. In addition RED maintains two thresholds of queue size min_{th} and max_{th}. When the average queue size is less than the minimum threshold, no packets are marked. When the average queue size is grater than the maximum thresholds, every arriving packet is marked. Marked packets are in fact dropped, this ensures that the average does not

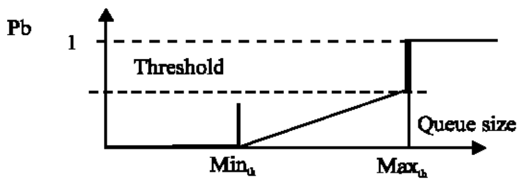

Fig. 1: General algorithm for RED gateways

significantly exceeds the maximum threshold. When the average queue size is between the minimum and maximum thresholds each arriving packet is marked with probability p_a, where p_a is a function of the average queue size. The algorithm is described as below by Floyd *et al*. (1993) and Jacobson (1998).

RED algorithm in general:
If avg< min_{th}, no packet drop
If min_{th}<= avg <= max_{th}, drops each packet with the probability P_d.
If avg > max_{th}, drop every arriving packet.

For each packet arrival:
    Calculate the average queue size avg
If min_{th}<= avg <= max_{th}
    Calculate probability p_a
        With probability p_a:
            Mark the arriving packet
Else IF avg > max_{th}
            Mark the arriving packet

Thus the RED gateway has two separate algorithms (Fig. 1). The algorithm for computing the average queue size determines the degree of burst ness that will be allowed in the gateway queue. The algorithm for calculating the packet marking probability that determines how frequently the gateway marks packets; given the current level of congestion the detailed RED Algorithm is shown bellow. The goal is for the gateway to mark packets at fairly evenly spaced intervals, in order to avoid biases and to avoid global synchronization and to mark packets sufficiently frequently to control the average queue size (Floyed *et al*., 1993; Jacobson, 1998).

Initialization
  avg ← 0
  Count ← -1
For each packet arrival
Calculate new avg. Queue Size avg:
  If the queue is nonempty
    avg ← (1-w_q) avg + w_q q
Else
    m ← f(time-q-time)
    avg ← $(1-w_q)^m$ avg
If min_{th} <= avg <= max_{th}
    Increment count
    Calculate probability p_a:
    $p_b \leftarrow max_p\,(avg-min_{th})/(max_{th}-min_{th})$.
    $P_a \leftarrow P_b/(1-count.P_b)$.
With probability P_b
    Mark the arriving packet
    Count ← -0
Else if max_{th} <= avg
    Mark the arriving packet
    Count ← -0
else Count ←-1
When queue becomes empty
q-time ← time

Saved variables:
Avg: Average queue size.
q-time: Start of the queue idle time.
Count: Packet since last marked packet.

Fixed parameters:
w_q: Queue weight (is determined by the size and duration of burst in queue size that are allowed at the gateway)
min_{th}, max_{th}: Maximum and minimum thresholds of the queue
max_p: Maximum value for p_b.

Other:
Pa: Current pkt- marking probability
q: Current queue size.
Time: Current time.
F (t): A linear function of the time t.

The gateway's calculations of the average queue size takes into account the period when the queue is empty (the idle period) by estimating the number m of small packets that could have been transmitted by the gateway during the idle period. After the idle period the gateway computes the average queue size, if m packets had arrived to an empty queue during that period. As avg varies from min_{th} to max_{th}, the packets marking probability p_b varies linearly from 0 to max_p:

$$p_b \leftarrow max_p(avg-min_{th})/(\,max_{th}-min_{th}).$$

The final packet marking probability pa increases slowly as the count increases since the last marked packet:

$$Pa \leftarrow P_b/(1\text{-count. } P_b).$$

This insures that the gateways does not wait too long to mark the packets when the average queue size avg exceeds max_{th}. One option for the RED gateway is to measure the queue in bytes rather than in packets. With this option, the average queue size accurately reflects the average delay at the gateway. When this option is used, the algorithm would be modified to ensure that the probability that a packet is marked is proportional to the packet size in bytes:

$p_b \leftarrow max_p (avg\text{-}min_{th})/(max_{th} - min_{th})$.
$p_b \leftarrow p_b$ PcketSize/maximumPacketSize.

In this case a large FTP packet is more likely to be marked than is a small TELNET packet. The queue weight w_q is determined by the size and duration of bursts in queue size that are allowed at the gateway. In this paper our primary interest is in the functional operation of the RED gateways and the most efficient implementation of the RED algorithm to support multimedia applications over wireless networks.

EFFECT OF RED ON MULTIMEDIA TRAFFIC

In IP networks, packets that is bigger than MTU (Maximum Transmission Unit) value of out going interface will be broken to some smaller packet before sending. Each of these small packets has its own IP header and offset information in order to help IP layer at receiver to reunite. Even MTU is bigger than the suggested value of packet size mentioned by Vu *et al.* (2000) for wireless networks, but Multimedia data frames still be broken, such as video frames, voice, music, etc. There is a problem with fragmentation at IP layer: when one piece of those fragments can not reach to the destination, which is often in error channels like wireless networks, the whole data packet will be discarding at IP layer, without any inform to higher layer. In its turn, Multimedia application hardly recover the lost packet.

Some existing solutions suggest using FEC for some data packets (Vu, 2000), so that the higher layer can recover loss packets from successful ones. However it cost much CPU resource and times delay, especially when data packet size is big such as in multimedia video frames. With RED, new incoming packet will be dropped probability P_d to avoid congestion. This means the newer packets, which have more valuable for Multimedia data will be discarded, instead of older packets. More over, RED influents multimedia performance, since dropped packet causes whole data frame at application layer become useless, or wasting more resource to recover the error (Cnodder *et al.*, 2000; Chen *et al.*, 1999).

Table 1: The first modification of RED algorithm using TOS field

RED-0	RED-1	Step
If Pa< avg/q	If P_a< avg/q	(1)
Mark each arriving packet	Drop packets belong to old frame, in queue	(2)
count ← 0	count ← 0	(3)

We suppose a method to help RED working more efficiently. RED should recognize fragment from one data frame and when it needs to drop one fragment it also drops others from the same frame. RED should also drop old packet in queue with probability P_b, so that sender can detect faster when some dropping occurred. In order to achieve those features, not only RED getaway must be changed, but also it needs support from the transport protocol.

Design issues and our modifications on red: RED should recognize fragment from one data frame and when it needs to drop one fragment it also drops others from the same frame. We suggest that the transport protocol marks frames ID at IP packet header, so RED gateway can recognize fragments, which belong to the same data frame. To detect frames in the same data frame we propose to use TOS field which is an unused field at the IP layer, this field can be used to store data frame ID. This field can be read by RED gateway at the base station level, so BS can manage the buffer actively. This will be our first modification on the RED algorithm as shown in Table 1.

In RED-1 packets will be dropped according to the frame ID shown in TOS field. This will solve the problem of discarding newer incoming packets, which have more valuable for multimedia data, in which packets from the same data frame or old data frames will be dropped to make it easer for the multimedia application to recover the packet loss that happens due to congestion in the wireless link (Mahadevan and Sivalingam, 2000). As shown in Fig. 2.

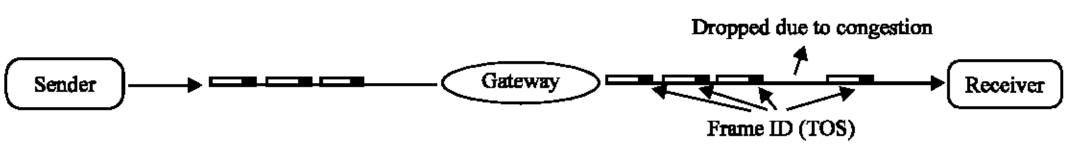

Fig. 2: Dropping packets due to congestion coherence using TOS field

Table 2: The second modification of RED algorithm using the packet sizes

Red- 1	Red-2	Red-3	Step
Count ← Count +1	Count ← Count +1	Count ← Count +1	(1)
$p_b \leftarrow max_p . \frac{avg - min_{th}}{max_{th} - min_{th}}$	$p_b \leftarrow max_p . \frac{avg - min_{th}}{max_{th} - min_{th}}$	$p_b \leftarrow max_p . \frac{avg - min_{th}}{max_{th} - min_{th}}$	(2)
	$p_b \leftarrow p_b . L/M$		(3)
$p_a \leftarrow p_b /(1 - count.p_b)$	$p_a \leftarrow p_b /(1 - count.p_b)$	$p_a \leftarrow \frac{p_b . L}{(1 - count.p_b).M}$	(4)

In order to allow transmission for bursts of huge packets (vidoe, images, etc.) over wireless linkes without any degradation of QoS level, we propse another modification for RED Algorithm in which it takes into account the packet size when estimating the drop probability, so it weights the drop probability by the packet size(we denoted it by RED_2), while the brevios algorithm (that we denoted by RED_1) does not consider the packet size while estimating p_b. This kind of discrimination between small and large packets is intended to avoid extra delay, incurred by retransmissions, of sensitive interactive traffic which generally consists of large packets. gives the steps needed for estimating the drop probability, p_b, on each packet arrival for RED-1 and RED-2.

In Table 2 the significance of the used parameters and variables is as follows: p_b is a temporarily dropping probability, max_p is an upper bound on the temporarily packet drop probability, min_{th} and max_{th} are the two thresholds limiting the region where packets are randomly dropped, L is the size of the incoming packet, M is the maximum packet size and count is the number of accepted packet since the last drop or since avg exceeded min_{th}. Note that the only difference between the two algorithms (RED-1 and RED-2) is the third step in RED-2 where the temporarily dropping probability p_b is weighted by the packet size. An attractive property for RED-1 resulting from using the count variable is that the drop probability is uniformly distributed. And then we finally present RED-3 that will illustrate the whole modification.

As we said before in order to make RED works more efficiently in case of congestion while sending multimedia data over the wireless link it needs support from the transport protocol. This will be presented in the following section.

A transport protocol supporting multimedia in wireless networks: We propose a new protocol, running over UDP, which allows application layer protocols to classify data depending on data's dependency of time sensitive, loss sensitive. With help of Proxy at Base station, wireless errors can be distinguished with congestion errors. Packet size is calculated to optimize through put, depend on BER level. If there is a Base Station (BS) which is connected to

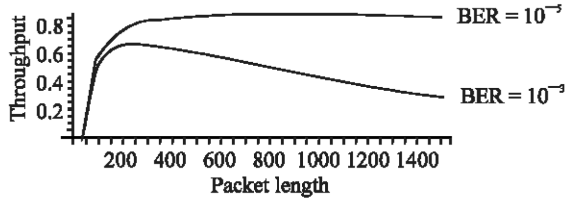

Fig. 3: Wireless Bit Error Rate (BER) and dynamic packet length

the wired network on one end to the wireless network on the other end. We consider the transmission in both directions separately: transmission from Bobile Host (MH) to Fixed Host (FH), bandwidth compensation in scheduling at BS is used for error control behaviour.

However, BER changes frequently and hardly to be measured. We supposed that errors occur in wireless path caused by BER. Thus, when wireless errors detected Sender at MH or BS will reduce packet length to 1/2 to reduce possibility of Packet corruption. To detect quickly wireless Error, besides containing sequent number, packet header also contains packet predictor number, which informs BS a number of packet will be sent next time.

Wireless error can be detected by ACK packets send back to sender from receiver when data packet is successfully received. When RED discards packet in it queue, sender will faster detect the error. The congestion occurred when buffer at BS is overflow or when after some time out but no more packets arrives from FH. BS will inform MH and FH in (BER = 10^{-5} or BER = 10^{-3}) (Fig. 3). Every ACK packet about status of Congestion. Sender at FH will reduce sending rate to 1/2 according to TCP behaviour.

When applied to wireless networks where transmission errors are frequent, TCP is found to have poor performance, This is because the assumption behind TCP congestion control algorithm, that the majority of packet losses are caused by congestion is no longer true (Pang *et al*., 2003; Chunlei and Jain, 2004). When a wireless loss is treated as a congestion loss, the effective TCP transmission rate drops to half. If transmission errors happen frequently, the effective transmission rate of the wireless link becomes almost zero even though the network is not congested.

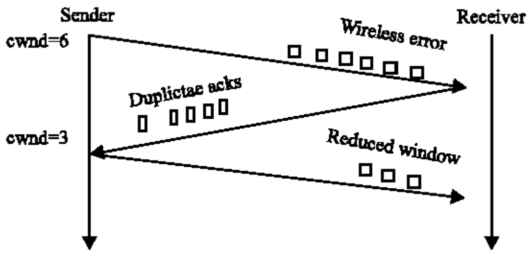

Fig. 4: Window reduction due to wireless loss

A scenario of such transmission rate drop is shown in Fig. 4. Suppose the network between the source and the destination can sustain a window size of six packets. TCP is transmitting at this rate when a transmission error on the wireless link causes a packet loss, resulting in out-of-order packets at the destination. Following current congestion control algorithm, the destination sends duplicate ACKs back to the source. Upon the receipt of three duplicate ACKs, the source assumes that congestion has happened in the network, retransmits the lost packet and reduces the window by half. Thus, the connection that can send six packets in a window is now sending only three as shown in Fig. 4, even though the network is not congested. The rapid development of mobile and wireless networks is a driving force for wireless TCP enhancements. In the past few years, numerous enhancements have been proposed. Theses enhancements differ in their signaling and data recovery mechanisms, applicable network and traffic configurations and locations where changes need to be made (Chunlei and Jain, 2004; Mahadevan and Sivalingam, 2000). These approaches have big impacts on the feasibility, generality, effort and performance of the enhancements (Pang *et al*., 2003). In this study we classify and evaluate the approaches used by major enhancement proposals in the literature and propose a new enhancement that requires only local changes at the wireless hosts.

An important assumption of our enhancement of TCP is the Explicit Congestion Notification (ECN) (Cnodder *et al*., 2000; Lu *et al*., 1998). It uses two bits in the IP header and two bits in the TCP header for ECN capability negotiation and feedback delivery. When its queue length exceeds a threshold, a router marks the packet as congestion experienced. At the destination, the congestion experience bit is copied to the ECN-echo bit in the TCP acknowledgment and sent back to the source with the ACK. The key of wireless TCP enhancement is the ability to determine the cause of packet losses. We find that the ECN signals carried by neighboring packets provide a simple and effective way to determine the cause. Unlike packet drops that lack coherence among neighboring packets, packet markings are coherent in a sequence of packets. If neighboring packets are marked as congested, the lost packet is most likely a congestion loss. If none of the neighboring packets is marked, then the lost packet must be a wireless loss.

The proposed enhancement is a transport layer signalling enhancement with link layer retransmissions. By utilizing the congestion coherence of ECN marking, it provides a light-weight TCP enhancement on wireless links. It has all the desirable characteristics. Even though this enhancement needs ECN support in all routers in the wired network, we still consider it as a local enhancement. This is because ECN is a protocol to improve wired networks even though no enhancement for wireless links is needed. The modifications to the existing TCP algorithm are made in the wireless end. It should be noticed that the modifications to the receiving and sending algorithms are made on the same end.

Destination algorithm

- The TCP destination follows existing algorithm for sending new acknowledgments, first and second duplicate acknowledgments.
- When the third duplicate acknowledgment is to be sent, TCP destination checks whether the coherence context is marked. If yes, the acknowledgment is sent right away. Otherwise, it is deferred for w ms, which is predetermined according to the time to complete a local link layer retransmission. A timer of t ms is started.
- If the expected packet arrives during the t ms, a new acknowledgment is generated and the timer is cleared.
- If the timer expires, all deferred duplicate acknowledgments are released.

Source algorithm

- The TCP source follows existing algorithm for sending packets and updating the congestion window upon receiving new acknowledgments and first and second duplicate acknowledgments.
- When the third duplicate acknowledgment arrives, the source checks whether any acknowledgment in the coherence context is an ECN-Echo. If yes, the packet corresponding to the duplicate acknowledgments is sent right away and the congestion window is reduced to half if a reduction has not been done in the previous RTT. Otherwise, the source ignores the duplicate acknowledgment and a timer of t ms is started.

- If a new acknowledgment arrives during the ms, the timer is cleared and new packets are sent as if the duplicate acknowledgments did not happen.
- If the timer expires, the packet corresponding to the duplicate acknowledgments is sent and the congestion window is reduced to half if a reduction has not been done in the previous RTT.

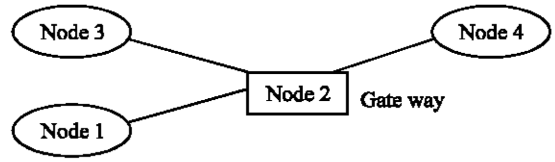

Fig. 5: Our simulation network

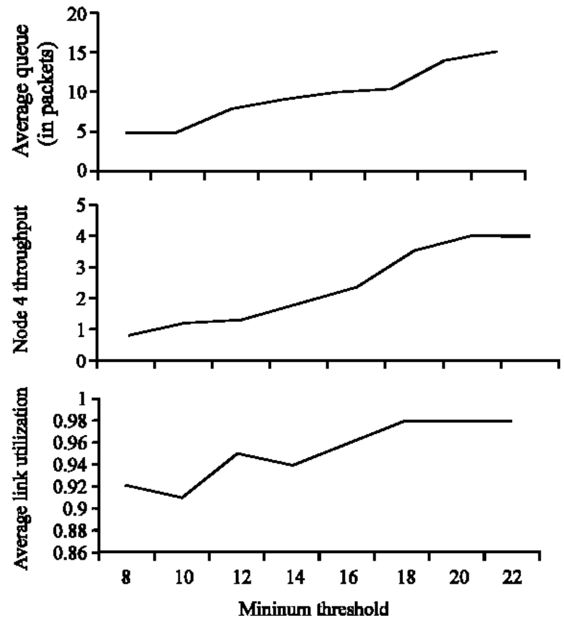

Fig. 6: Simulation with old RED gatways

EMPIRICAL WORK

Video traffic and some forms of interactive multimedia traffic are examples of bursty traffic seen by the gateway. In our simulation section we use FTP connections with infinite data, small windows and small roundtrip times to model, the less-bursty traffic and we use FTP connections with smaller windows and longer roundtrip times to model the more-bursty traffic. Bursty traffic at the gateway can result from an FTP connection with a long delay-bandwidth product but a small window; a window of traffic will be sent and then there will be a delay until the acknowledgment packets return and another window of data can be sent.

The simulation platform we used is the QualNet simulator (Qual Net, Network simulator, 2005), which is the successor of the previous GloMoSim simulation library. Most configuration parameters of the protocol stack in our simulations use the default values. The channel bandwidth is 2Mbps, channel propagation model is the two-ray ground reflection model and transmission range is 367m. IEEE 802.11 MAC DCF is adopted. TCP NewReno and we simulate the network shown below

Network components:

- Two mobile Nodes (node 1, node 3).
- Base Station (node 2) representing a gateway.
- One pc client (node 4).

Network connections:

- Wireless connections between the mobile nodes and the base station which represents the gateway.
- Special super application (for multimedia data support) connection between a mobile node1 and the base station.
- A CBR (centralize Band with reservation) connection between node 3 and node 2.
- Wired connection between the pc (node 4) and Base station (node 2).

We consider our simulations from the network in Fig. 5. Node 4 packets' have a roundtrip time that is six times that of the other packets. Connections 1-3 have a maximum window of 12 packets, while connection 4 has a maximum window of 8 packets. Because node 4 has a large roundtrip time and a small window, node 4 packets often arrive at the gateway in a loose cluster. By this, we mean that considering only node 4 packets, there is one long inter arrival time and many smaller inter arrival times.

Figure 5 through 7 show the results of simulations for the network in Fig. 5. In which two RED gateways were tested, the existing Old RED gateway and our Modified RED gateway respectively. The simulations in Fig. 6 and 7 were running both with minimum threshold ranging from 8 packets to 22 packets. Each simulation was run for ten seconds and each mark represents one-second period of that simulation. For Fig. 6 and 7 the x-axis shows the Minimum Threshold and the y-axis shows node 4's throughput as a percentage of the total throughput through the gateway. In order to avoid traffic phase effects (effects caused by the precise timing of packet arrivals at the gateway).

The gateways cannot be compared simply by comparing the maximum queue size; the most appropriate comparison is between RED gateways that maintains the same average queue size. With the old version of RED

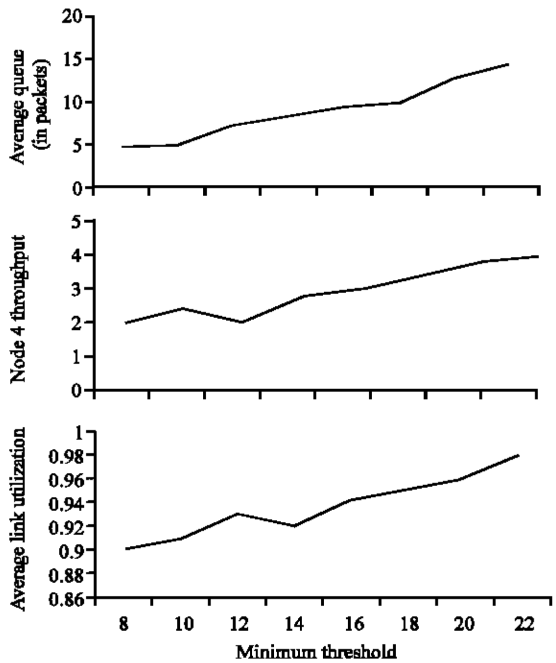

Fig. 7: Simulation with modified RED gateways

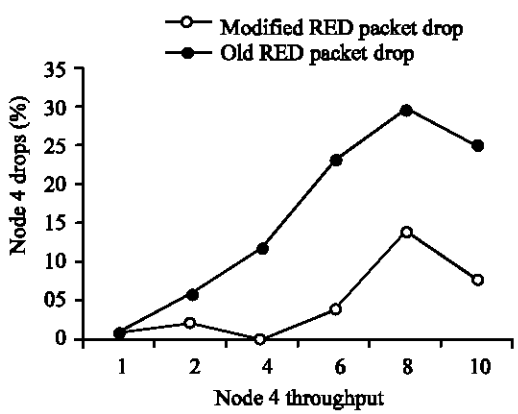

Fig. 8: Comparison between modified and old RED packet dropping

gateway, the queue is more likely to overflow when the queue contains some packets from node 4. In this case, node 4's packets have a disproportionate probability of being dropped; the queue contents when the queue overflows are not representative of the average queue contents. Figure 7 shows the result of simulations with our modified version of RED gateways. The x-axis shows minimum threshold and the y-axis shows node 4's throughput. The throughput for node 4 is close to the maximum possible throughput, given node 4's roundtrip time and maximum window. The parameters for the RED gateway are as follows wq: 00.02 and max_p: 1/50 the maximum threshold is twice the minimum threshold and the buffer size, which ranges from 12 to 56 packets, is four times the minimum threshold.

Figure 8 shows that in the simulations with Old RED gateway, node 4 receives a huge amount of packet drops. Each mark in Fig. 8 shows the results from a one-second period of simulation. The Squares shows the simulations with Old RED gateway, the triangles shows the simulations with our modified RED gateways, For each one-second period of simulation, the x-axis shows node 4's throughput (as a percentage of the total throughput) and the y-axis shows node 4's packet drops (as a percentage of the total packet drops).In contrast, for simulations results shown in Fig. 8 the old RED gateway version shows weakness and biasness against the bursty multimedia traffic from node 4. In the other hand. our modified RED version reduces the amount of packets being dropped and shows a high performance against multimedia traffic.

CONCLUSION AND FURTHER RESEARCH

In this study we proposed a modification on the original RED gateway used in the wired network to adapt it in wireless networks. By Modifing RED's algorithm, detecting early congestion and dropping packets proportionally to a flow's channel bandwidth usage. We also address the various enhancements that need to take place in the TCP protocol to make it suitable with RED for providing QoS assurances in transmitting bursty multimedia traffic in wireless environment. Experimental results are used to validate the enhancements of the new proposed model.

The minimum and maximum thresholds min_{th}, max_{th} are determined by the desired average queue size (Jacobson, 1998; Chunlei and Jain, 2004). The average queue size which makes the desired tradeoffs (such as the tradeoff between maximizing throughput and minimizing delay) depends on wireless network characteristics, are left as a question for our further research.

REFERENCES

Braden, B., D. Clark and B. Davie, 1998. Recommendations on queue management and congestion avoidance in the internet. Internet Performance Recommendations. 1: 1-17.

Chen, J., K.M. Sivalingam, P. Agrawal and R. Acharya, 1999. Scheduling multimedia services for a low-power MAC in wireless and mobile ATM networks, IEEE Transactions on Multimedia, 12: 187-201.

Chunlei, L. and R. Jain, 2004. Congestion coherence: A local enhancement for tcp over wireless links, IEEE Transactions on Networking, pp: 1-18.

Cnodder, D., O. Elloumi and K. Pauwels, 2000. Effect of different packet sizes on RED performance. Traffic and Routing Technologies project Alcatel Corporate Research Center, Francis Wellesplein, 1-2018 Antwerp, Belgium, pp: 1-9.

Floyd, S., V. Jacobson and L. Berkeley, 1993. Random early detection gateways for congestion avoidance. IEEE/ACM Transactions on Networking, 1: 397-413.

Floyd, S. and V. Jacobson, 1994. The synchronization of periodic routing messages, in IEEE/ACM transactions on Networking, 2: 122-136.

Jacobson, V., 1998. Congestion Avoidance and Control. ACM Special Interest Group on Data Communication, 34: 314-329.

Jain, R. and K.K. Ramakrishnan, 1997. Congestion avoidance in computer networks with a connectionless network layer: Concepts, goals and methodology, proc. IEEE Comp. Networking Symp.,Washington, DC., pp: 134-143.

Loguinov, D. and H. Radha, 2003. RAP: End-to-end rate-based congestion control: Convergence properties and scalability analysis. IEEE/ACM Transactions on Networking (TON), 11: 564-577.

Lu, S., T. Nandagopal and V. Bhargavan, 1998. A wireless fair service algorithm for packet cellular networks. In: Proc. ACM MobiCom, Dallas,TX pp: 10-20.

Mahadevan, I. and K.M. Sivalingam, 2000. Architecture and experimental results for Quality of Service in mobile networks using RSVP and CBQ. Wireless Networks, 6: 221-234.

Padhye, J., V. Firoiu, D. Towsley and J. Kurosee, 1998. Modelling tcp throughput: A simple model and its empirical validation, In ACM SIGCOMM'98, Vancouver, 1: 1-23.

Pang, O., C.L. Soung, C.P. Fu, Wang, O. Victor and K. LI, 2003. Performance Study of TCP Veno over WLAN and RED Router, pp: 1-5.

QualNet, Network simulator Available at http://www.qualnet.com, 2005.

Sisalem, D. and H. Schulzrinne, 1998. the loss-based adjustment algorithm: A Tcp-friendly adaptation schemes: In Proc. of Intern.Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV), Cambridge, England, 9: 564-577.

Vu, T., 2000. Effect of RED and different packet sizes on Multimedia performance over wireless networks. TU Ilmenau, Germany, pp: 1-7.

Vu, T., D. Reschke and W. Horn, 2000. RCAP and Dynamic Packet Size Mechanism (DPSM) for Multimedia in wireless network, NetObject TU Ilmenau, Germany,

Zhang, L. and D. Clark, 1990. Oscillating Behaviour of Network Traffic: A Case Study Simulation. Internetworking: Research and Experience, 1: 101-112.