A Congestion Avoidance Approach in Jumbo Frame-enabled IP Network

Aos Anas Mulahuwaish, Kamalrulnizam Abu Bakar, Kayhan Zrar Ghafoor
Faculty of Computer Science and Information System
Universiti Teknologi Malaysia
Johor, Malaysia

Abstract— Jumbo frame is as an approach that allows for higher utilization of larger packet sizes on a domain-wise basis, decreasing the number of packets processed by core routers while not having any adverse impact on the link utilization of fairness. The major problem faced by jumbo frame networks is packet loss during queue congestion inside routers is as the RED mechanism that is recommended to combine with jumbo frame treats jumbo frame encapsulation as one packet by drop the whole jumbo frame with packets encapsulate during the congestion time. RED dropping the whole jumbo frame encapsulation randomly from head, middle and tail inside queue of router during periods of router congestion, leading to affect the scalability and performance of the networks by decreasing throughputs and increasing queue delay. This work proposes the use of two AQM techniques with jumbo frame, modified Random Early Detection MRED and developed drop Front technique DDF, which are used with the jumbo frame network to reduce packet drop and increase throughput by decreasing overhead in the network. For the purpose of evaluation, network simulator NS-2.28 was set up together with jumbo frame and AQM scenarios. Moreover, for justification objectives, the proposed algorithm and technique for AQM with jumbo frame were compared against the existing AQM algorithm and techniques that are found in the literature using metrics such as packet drop and throughput.

Keywords- Jumbo Frame; Queue Congestion; AQM; RED.

I. INTRODUCTION

Computer networks have experienced rapid growth over the years, from transferring simple email messages to now being a full media resource where full length movies are commonly transmitted. Many users have begun to use the internet for many things; as a result, the connections of internet have started to become strained where before the common solution of the internet service provider (ISP) was capable of providing sufficient bandwidth to users in the network. However, recent research has found that the users' access speed has increased and thus affects the efficiency of the network. Therefore new techniques need to improve the efficiency of the network traffic. Many techniques from multicasting to packet caching have been used to improve the efficiency of the network, but with limited success as these techniques suffer from one or more drawbacks including global network support, application support, asymmetric and computation overhead. The current assumption with networking research is also that it affects an individual network flow's quality of service (QoS) including the packet loss, end to end delay and jitter; however these researches presented techniques that investigate the possibility of trading a minimal amount of an individual flow's QoS typical delay so as to obtain better overall network performance [1].

One of the issues facing networks is the number of packets required to be processed per second, whereby the gigabit link core router may have to route anywhere from 90,000 to 2,000,000 packets per second. As line speed increases to greater rates, so does the number of packets that need to be processed; one way to reduce the load on the router is to increase the maximum transmission unit (MTU) of the network. Unfortunately while the MTU of Ethernet is 1500 bytes, up to 50% of the packets transferred across the network are 64 bytes or less.

Jumbo frame is a technique that aims to reduce the number of packets processed by the core routers, by reducing the number of packets. This is accomplished by transmitting many packets in the domain into a single large jumbo frame for transmission across the core network. In ordering the aggregate packets together in a jumbo frame, incoming packets are queued briefly by egress point. Once the jumbo frame reaches the egress of the domain, the original packets are rebuilt and transmitted on toward their final destination.

II. RELATED WORK

A jumbo frame has a common size of 9000 bytes, which is exactly six times the size of a standard Ethernet frame [5]. A 9k byte jumbo frame would be 9014-9022 bytes together with the Ethernet headers. This makes it large enough to encapsulate a standard network file system (NFS) data block of 8192 bytes, yet not large enough to exceed the 12,000 byte limit of Ethernet's error checking in cyclic redundancy check algorithm (CRC) [5].Undoubtedly, smaller frames usually mean more CPU interruptions and more processing overhead for a given data transfer size [9]. When a sender sends a data, every data unit plus its headers have to be processed and read by the network components between the sender and the receiver. The receiver then reads the frame and TCP/IP headers before processing the data. This whole process, plus that of adding the header to frames and packets from the sender to the receiver consumes CPU cycles and bandwidth [13]. For these reasons, increasing the frame size by sending data in jumbo frames means fewer frames are sent across the network when considering the fact of high processing cost of network packets [3]. These generate improvements in CPU utilization and bandwidth by allowing the system to concentrate on the data in the frames, instead of the frames around the data. The justification behind increasing the frame size is clear; larger frames reduce the number of packets to be processed per second. A single 9k Jumbo Frame replaces six 1.5k standard frames, producing a net reduction of five frames as only one TCP/IP header and Ethernet header is required instead of six, resulting in 290 (5*(40+18)) fewer bytes transmitted over the network [14].

In terms of improving bandwidth, it takes over 80,000 standard Ethernet frames per second to fill a gigabit Ethernet pipe, which in turn consumes a lot of CPU cycles and overhead. By sending the same data with 9,000 bytes jumbo frames, only 14,000 frames need to be generated and the reduction in header bytes frees up 4 Mbps of bandwidth. The resources used by the server to handle network traffic are proportional to the number of frames it receives. Therefore, using fewer large frames dramatically improves server and application performance, compared to a larger number of smaller frames [14]. Jumbo frame improves core router scalability, by encapsulating packets with the same next autonomous systems (AS) and egress point into larger packets for transmission across the domain. Critically, the design of jumbo frame functions on a domain-wise scale, instead of end-to-end; the external entities (other domains and end hosts) are unaware that any conversion took place. The overall jumbo frame shown in Figure 1:

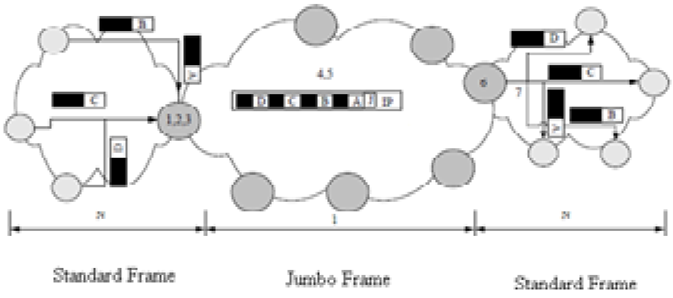

Figure 1. Jumbo Frame Structure

The description of the structure as shown in Figure 1 is that when packets arrive at an ingress node to the domain, the ingress node and the packets are sorted into queues based upon their egress point of the network in their path that is obtained from the border gateway protocol (BGP) routing table [12]. A jumbo frame Encapsulation Timer (JFET) is started for the queue. Packets that are being sent through the same egress point are combined into the same jumbo frame, subject to MTU; once the JFET for the queue has expired, the Jumbo Frame is released towards the next AS. The jumbo frame is routed through the core of the network, with the routing provided by using the standard routing mechanism of the network. The jumbo frame arrives at the egress node and the original packets are separated out, after which the original packets are forwarded onto their final destinations. There are two main benefits of using jumbo frame [1]. The first benefit is that jumbo frame lowers the number of packets that the core routers are responsible for processing, thus allowing better scaling for the network as line speeds increase. The second beneficial aspect of jumbo frame is that data is more efficiently transferred by reducing the number of physical layer headers used (due to a lower number of packets).

A. *Fast Packet Encapsulation*

The jumbo frame is structured to allow for efficient encapsulation, inspection, and de-capsulation [5]; packet overhead is minimal and is offset by the reduction in physical layer headers. The structure of the jumbo frame is shown in Figure 2 containing the following fields:

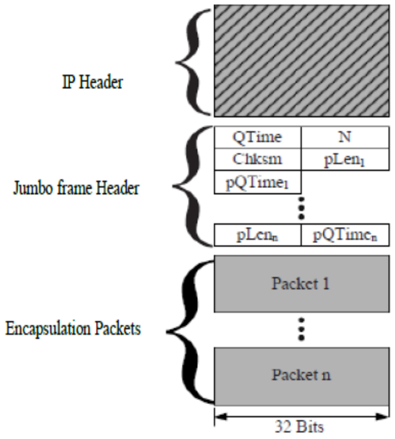

Figure 2. Jumbo Frame Structure

The destination address of the jumbo frame is the same as the first packet stored in the jumbo frame. For a multiprotocol label switching (MPLS) network, the destination address is the MPLS address of the first packet stored in the group. This ensures proper routing for all packets as all encapsulated packets contained in the jumbo frame would arrive at the next correct AS in their path. The design of the jumbo frame allows the original packets to be de-capsulated with minimal effort while also keeping the overhead of the jumbo frame to a minimum. As shown, the overhead of the jumbo frame is 6 + 4N bytes. However, the overhead is offset by the reduction in physical layer headers. The net cost (or benefit) of jumbo frame can be stated as:

Equation 1: $\text{Cost} = \text{HIP} + \text{HJG} + (N \_ 4) - (\text{Hp} \_ (N - 1))$ (1)

The cost of jumbo frame in the above equation comes from the size of the IP header (HIP), the jumbo frame header (HJG), and the number of encapsulated packets (N). The reduction in bandwidth comes from the reduction in physical headers 5 (Hp). For example, if the network is an Ethernet network and two packets were encapsulated into a jumbo frame, then HIP = 20, HJG = 6, N = 2, Hp = 38, and the total cost would be −4 bytes. In other words, 4 bytes of bandwidth would be conserved.

B. *Egress Shaping*

When the jumbo frame reaches its destination, the packets need to be de-capsulated and released to the next node on their path to the destination. If all the packets are released as soon as they are removed from the jumbo frame, this can lead to dropped packets at the client due to the receive buffer overflowing [2]. Hence packets are shaped at the egress according to the differences in their arrival time (pQTime). In other words, if two packets arrive at the ingress node 4 ms apart, they are released from the egress node 4 ms apart.

C. Active Queue Management (AQM) with Jumbo Frame

The structure of the jumbo frame allows active queue management AQM scheme techniques and methods are an important type of technology with aims to improve the utilization of the network [4] and [8]. While jumbo frame can be combined with AQM techniques and methods, this allows the combination between the jumbo frame and AQM techniques and methods to solve many problems in jumbo frame networks, and also enhance the efficiency and scalability of jumbo frame network by decreasing the packet loss and end to end delay, reducing the overhead and increasing throughput for jumbo frame networks to perform optimally, RED is one of the AQM methods that work with jumbo frame [4] and [8], for preventing the gateway router from becoming full and ensuring that jumbo frame can transmit to the destinations.

In [6] and [11], two different methods that RED queues can use to determine the queue utilization are presented. The first is through the number of packets in the queue and the second is to determine queue utilization by number of bytes in the queue. RED detect the congestion in jumbo frame networks, and decrease the congestion of overflow by randomly drop whole jumbo frame, RED treat jumbo frame as a one big packet, so when the drop occur RED will used the same drop operation with standard packet.

D. Random Early Detection (RED) and Drop from Front

Random early detection (RED) Algorithm was first proposed by [6]. This discipline maintains a moving average of the queue length to manage congestion. If this moving average of the queue length lies between a minimum threshold value and a maximum threshold value, then the packet is either marked or dropped with a probability. If the moving average of the queue length is greater than or equal to the maximum threshold then the packet is dropped. Even though it tries to avoid global synchronization and has the ability to accommodate transient bursts, in order to be efficient RED must have sufficient buffer spaces and must be correctly parameterized. In contrast, RED algorithm uses packet loss and link utilization to manage congestion. RED gateways can be useful in gateways 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 the example of a gateway with separate queues for real time and non-real time traffic, RED congestion control mechanisms could be applied to the queue for one of these traffic classes.

The RED congestion control mechanisms monitor the average queue size for each output queue, and by using randomization chooses connections to notify of that congestion. Transient congestion is accommodated by a temporary increase in the queue. Longer-lived congestion is reflected by an increase in the computed average queue size, and results in randomized feedback to some of the connections to decrease their windows. The probability that a connection is notified of congestion is proportional to that connection's share of the throughput through the gateway. In addition, gateways detecting congestion before the queue overflows are not limited to packet drops as the method for notifying connections of congestion. RED gateways can mark a packet by dropping it at the gateway or by setting a bit in the packet header, depending on the transport protocol. When the average queue size exceeds a maximum threshold, the RED gateway marks every packet that arrives at the gateway. If RED gateways mark packets by dropping them, rather than by setting a bit in the packet header, then the RED gateway controls the average queue size even in the absence of a cooperating transport protocol when the average queue size exceeds the maximum threshold. One advantage of a gateway congestion control mechanism is that it works with current transport protocols and does not require that all gateways in the internet use the same gateway congestion control mechanism; instead it could be deployed gradually in the current Internet. RED gateways are a simple mechanism for congestion avoidance that could be implemented gradually in current TCP/IP networks with no changes to transport protocols.

Drop from front technique drops the head of the queue if the incoming packet sees the queue as full. With the drop from front policy that governs when a packet arrives to a full buffer, the arriving packet is allowed in, with space being created by discarding the packet at the front of the buffer. This shows that for networks using TCP, the Internet transport protocol, a drop from front policy results in better performance than is the case under tail dropping and its variations [10]. Drop from Front a partial solution to the problem of throughput collapse in networks where TCP represents a sizeable part of the load. Drop from Front can be used in conjunction with other strategies such as Partial Packet Discard. In [10], showed that moving to a drop from front strategy considerably improves performance and allows use of smaller buffers than is possible with tail drop. Drop from Front is also applicable to both the switch and routers. During congestion episodes when buffers are full, Drop from Front causes the destination to see missing packets in its received stream approximately one buffer drain time earlier than would be the case under tail drop. The sources correspondingly receive earlier duplicate acknowledgements, causing earlier reduction in window sizes.

However, drop from front has the advantage that the switch and router does not need to maintain a table of drop probabilities and does not have to know the traffic type being carried. This is because drop from front also reduces latencies for successfully transmitted packets and hence is a sensible policy to use for delaying sensitive non-feedback controlled traffic as well. This reduction in latency has been described by [15], who considered a "drop from front" scheme for a very different problem where none of the sources were feedback controlled. They found that drop from Front resulted in shorter average delay in the buffer for eventually transmitted packets and recommended its use for time-constrained traffic.

III. Methodology and Results

Modified random early detection (MRED): a RED queue is an important technique that aims to improve the utilisation of the network and remove the synchronisation that tends to occur with TCP flows when the network becomes congested. There are two different methods that RED queues can use to determine the queue utilisation. The first method is to simply use the number of packets, while the second is to use the number of bytes consumed in order to determine queue utilisation. The second method has more overhead, however, it allows for smaller packets to be favoured over larger packets. This effectively gives priority (less chance to be dropped) to smaller packets (eg.TCP acknowledgments). In jumbo frame networks if RED is not modified in any way, jumbo frame will be treated the same as any other packet. This behaviour is not advantageous as a jumbo frame has the same percent chance to be dropped as does any other packet. However, any time a jumbo frame is dropped, all encapsulated packets are lost. Because multiple packets are lost, this can result in poor TCP performance, as packets from the flow can be dropped, thus resulting in a greater than desired reduction of traffic.

MRED will start to calculate the new average queue size and the time for the new flows that coming to the queue and MRED will do compression between the number of jumbo frames and the capacity of the queue and check if the capacity of queue are enough to receive new flows or not. If the queue has enough space for all flows then MRED will allow all jumbo frames to queue up for forwarding out to the different destinations. However if the capacity is not enough, there is a congestion over flow problem that will happen in this queue, all that will be calculated based on the MRED detection mechanism. In this case MRED will do the calculation for each jumbo frame for probabilities drop. From here MRED will check the header of jumbo frame and will exactly check and compare two of fields inside the header. It will check the average length of each jumbo frame to calculate out the percentage of jumbo frame packets to distinguish that jumbo frame is not like any normal packet (this is because the average length size is high), MRED will also check the number of packets which encapsulated within the field header to verify there are encapsulation packets inside. Here MRED will only work with the average length and the number of packets that encapsulated within jumbo frame and will not work with the header of capsule frame.

After that, MRED will register out all of the information from the header for each jumbo frame encapsulation; then based on the percentage of packets that have been encapsulated, MRED will mark jumbo frame for drop sub packets inside, the percentage of packets that will drop are different from jumbo frame to another that are in the same flows. This calculation is based on the percentage of upper and lower bounds for each jumbo frame with the packets encapsulate, in which this calculation based on specific mathematical formulas. MRED will compare the percentage of packets inside each of jumbo frame with average queue size for the queue of router. Then MRED will decide the percentage of packets dropped from each of jumbo frame, to make the average queue size stable between them and during the congestion overflow time, and to reduce losing the whole encapsulation of jumbo frame but for only subs of packets. The marking operation of MRED for jumbo frame and the packets inside are related with time that sets for each jumbo frame, after that the probability marking drop will be set.

In this work MRED are combined with DDF, so based on this mechanism it will only mark the jumbo frame at the head of queue and the packets at the head of those jumbo frames. MRED will distribute the drop marking operation with the different jumbo frames to reduce the congestion and to let some of the packets inside those encapsulations left within without dropping it whole. This mechanism will reduce losing the whole packets inside each jumbo frame at the same flow; Figure 3 shows the diagram of MRED operation structure.

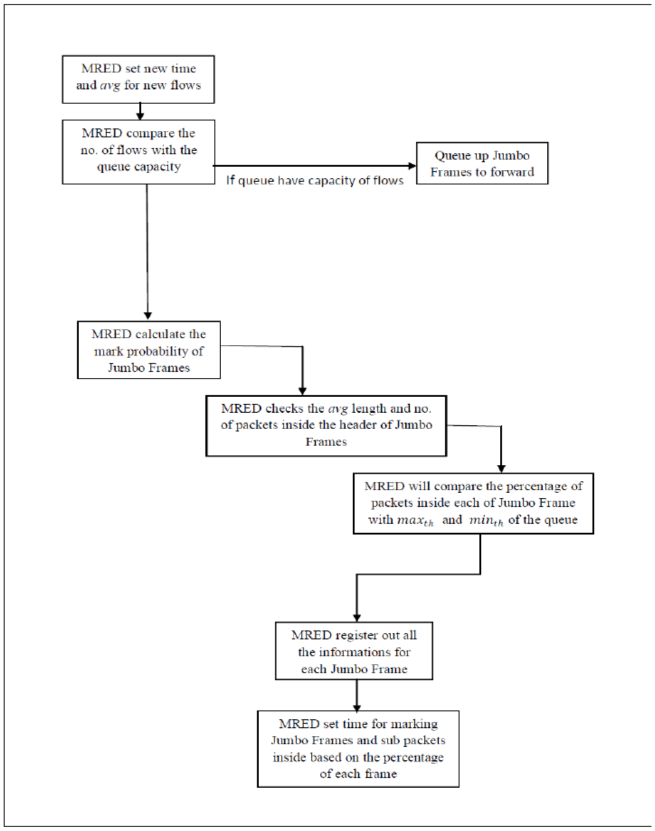
MRED set new time and avg for new flows
MRED compare the no. of flows with the queue capacity
If queue have capacity of flows
Queue up Jumbo Frames to forward
MRED calculate the mark probability of Jumbo Frames
MRED checks the avg length and no. of packets inside the header of Jumbo Frames
MRED will compare the percentage of packets inside each of Jumbo Frame with max_{th} and min_{th} of the queue
MRED register out all the informations for each Jumbo Frame
MRED set time for marking Jumbo Frames and sub packets inside based on the percentage of each frame

Figure 3. MRED operation structure

Developed drop front (DDF): development drop front mechanism is combined with modified RED for the steps of the packets drop in jumbo frame networks. After MRED has marked the jumbo frame that needs to be drop inside it, by calculating the upper and lower bounds for the encapsulations based on the percentage of jumbo frame encapsulations. When the MRED marked jumbo frames for dropping process, only the sub packets inside the jumbo frame will be dropped; the marked sub packets inside jumbo frame encapsulation will be done in the head of encapsulation frame, based on the mechanism of DDF which combines with MRED, so there are no random packets dropped inside jumbo frame. DDF will wait until the processing time finishes for the MRED with all the flow packets, then the time for DDF operation will start; DDF checks how many encapsulated for jumbo frame that marked by MRED, based on the percentage for each jumbo frame inside the queue. After checking the numbers of marked jumbo frames, DDF calculates the sub packets that are marked for drop by MRED inside each marked encapsulation.

DDF will set new time differently with time that was set before by MRED for each encapsulate frame that has been marked by MRED to do drops operation. Each marked jumbo frame has its own time drop packets. This time is set based on how many packets that are marked to drop each sub packets' time that have been marked to drop for this operation. There is a delay time for packet drop from one packet to another and this time will be calculated and set for the total drop operation time for the whole jumbo frame and each jumbo frame have different time with other. The drop operation starts with the first jumbo frame in the head of queue that was marked for drop operation. Inside this marked encapsulation sub packets drop operation will start with the first packet in the jumbo frame encapsulation that has been marked to drop. The DDF operation will start to drop packets by packets inside each jumbo frame encapsulation, and the packets drop will set in sequence number of router queue for each jumbo frames. Then it will send notification to the source for retransmitting the loss packets. In this operation, not all the encapsulation of jumbo frame is lost and the drop operation for the sub packets did not happen randomly but only from the front of jumbo frame, Figure 4 shows the DDF operation.

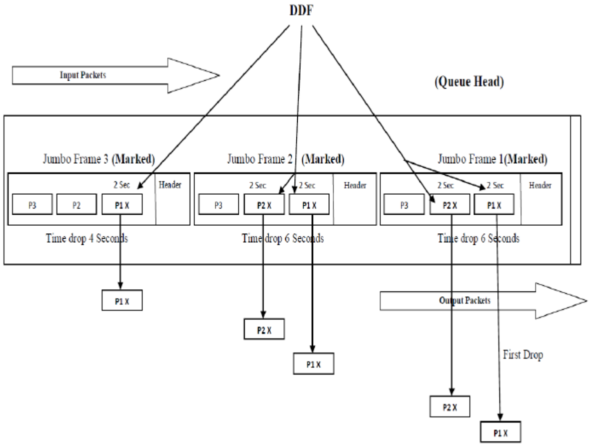

Figure 4. DDF operation

DDF allows the possibility of dropping partial packets without significant overhead. Firstly DDF looks at the number of packets stored in the jumbo frame encapsulations. Once the number of packets to be dropped is decided, the packets will removed from the head of the jumbo frame. The length of jumbo frame is shortened by the lengths of the packets that are to be removed and their lengths in the jumbo frame header are set to zero. The number of packets field for each jumbo frame got marked to drop sub packets will not be modified, and also the average length field in header will not be modified. This is due to the need for correct parsing at the egress router and the need for simplicity in modifying the packets in flight. Removing the lengths that are zeroed out is not a desirable option because multiple memory copies would have occur before the packet could be forwarded. So here jumbo frame will forward out without restructuring the sequencing of packets that were encapsulated, only the number of packets and average length fields in the jumbo frame header are not modified, DDF will set zero at jumbo frame header instead each packet has been dropped directly and one by one based on the time has been set for each jumbo frame marked and for each packets inside need to be dropped to remove the restructure operation, Figure 5 shows the average length of packets after drop operation inside jumbo frame header.. After that jumbo frame will de-encapsulate the rest packets to the destination address by the egress operation. DDF eliminates the random marked jumbo frame and dropped the packets inside encapsulation. However, if the packets are able to be removed randomly by MRED in jumbo frame, the complexity of the partial drop would substantially increase. The increase in complexity is from performing an MRED calculation on each encapsulated and from memory move operations needed to close the gaps in the jumbo frame after drop sub packets in different places in the encapsulation. DDF eliminates the restructure operation for each jumbo frame; all that will decrease the overhead in jumbo frame networks.

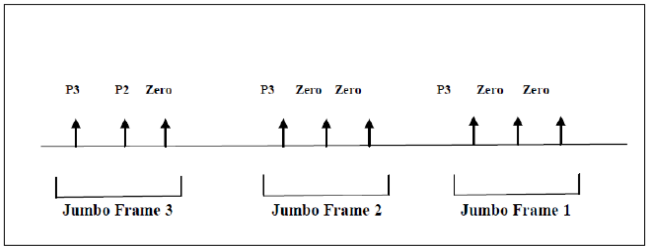

Figure 5. The average length of packets inside jumbo frame header after packets drop

A. *Simulation Setup*

The simulations presented here illustrate MRED with DDF well-understood dynamic of the average queue size varying with the congestion level, resulting from MRED with DDF and normal RED with tail drop fixed mapping from the average queue size to the packet dropping probability and the percentage of throughput. These simulations focus on the transition period from one level of congestion to another.

These simulations used a simple dumbbell topology with 6 nodes, the congested link of 1.5Mbps. The buffer accommodates 20 packets, which, for 3000 byte packet size and MTU 3000 byte, corresponds to a queuing delay of 0.28 seconds. In all of the simulations, weight of queue is set as a default in NS-2 to 0.0027, the choice of Wq determines the queue weight of the averaging for the average queue size, if Wq is too law, then the estimated average queue size is probably responding too slowly to transient congestion, if □□□□ it too high, then the estimated average queue size is too closely tracks the instantaneous queue size, MINth is set to 5 packets, the setting for MINth depends on exactly what the desired tradeoffs is at that router between low average delay and high link utilization. In the NS-2 MINth is set to a default of 5 packets because if MINth is set as small as one or two packets would only denied burstiness in the arrival process, and MAXth is set to 15 packets; there times more than MINth. Maximum value for the current marking of packet probability MAXp is constrained to remain within the range [0.01, 0.5] (or equivalently, [1%, 50%]), and the percentage of Jumbo Frame packets is 0.025, the average size of encapsulated packet is read from the Jumbo Frame header, not calculated at the router.

B. Simulation Scenario

The first scenario is for the increased average queue size in congestion, which used for testing the proposed MRED with DDF and also for test normal RED with tail drop in jumbo frame networks, this scenario is focus for the increase the average queue size in router queue during the congestion over flows at the transition period. The new flows are more than the buffer size capacity, the over flows burst in the specific simulation time, the average queue size has been increased because this over flows and been near or over the MAXth, so the congestion and packet drop happened, with decrease in throughput. This simulation is test the efficiency and scalability for the proposed MRED with DDF algorithm and compare the results with normal RED with tail drop results, for reduce the packet loss in and increase throughput with jumbo frames.

C. Results for MRED with DDF an Increased in Congestion Scenario

For this simulation scenario, the forward traffic consists of two long-lived TCP flows, and the reverse traffic consists of one long-lived TCP flow. At 25 seconds time, there are 20 new flows started, one every 0.1 seconds, each with a maximum window of 20 jumbo frames. This is not intended to model a realistic load, but simply to illustrate the effect of a sharp with the average queue size changing as a function of the packet drop rate. However after roughly 10 seconds, and because of the new 20 flows of jumbo frames the congestion happened, the algorithm of MRED detected the congestion and started to calculate the average queue size in the overflow time, MRED marked packets inside jumbo frames by put the drop probability first, and then mark sub packets inside jumbo frames at the head of queue and at the head of jumbo frames to decrease the congestion and then the drop will be done at the head of those jumbo frames by DDF without changing the length of information inside the header for each jumbo frames marked for drop. Here MRED with DDF has brought the average queue size back down to the range, between (6 – 7 packets). That means the proposed algorithm makes the average queue size away from the MAXth by making the probability of the packet drop less (MINth ≤ avg < MAXth).

The simulations with MRED with DDF have a higher throughput with smaller packet loss (drop), at the first half part of simulation, the throughput percentage is 42.45% and the packet drop is 0.69%. In the end of simulation scenario, the throughput becomes 91.7% and packet drop 8.24%. Figure 6 shows the MRED with DDF an increase in average queue size in congestion, the green trend represents the instantaneous change of queue length and the red trend shows the average queue size.

D. Result for Normal RED with Tail Drop an Increased in Congestion Scenrio

For this scenario simulation, it used the same simulation with MRED and DDF but with normal RED and tail drop instead. There are also at 25 seconds time where there are 20 new flows start, one for every 0.1 seconds, and each with a maximum window of 20 jumbo frames. In Figure 7 the graph illustrates normal RED with tail drop, with the average queue size changing as a function of the packet drop rate. With 20 new jumbo frames flow, congestion happened and packet dropped, because RED detected congestion and the RED algorithm dropped marked jumbo frames totally by tail drop at the tail of queue only. The packet drop rate changes from 0.90% with throughput 41.06% over the first half of the simulation, to 8.50% with the throughput 90.20% over the second half of simulation That means the average queue size here is become near to MAXth because of the algorithm for normal RED with drop tail did not reduce the number of packets that dropped during the congestion time. Due for that reason, the average queue size has been increased and the throughput has been decreased. Figure 7 shows the normal RED with tail drop with an increase in congestion, here can be noticed that at 25 second during the congestion the trend of average queue size increases and almost near with MAXth which means more packet drop happened.

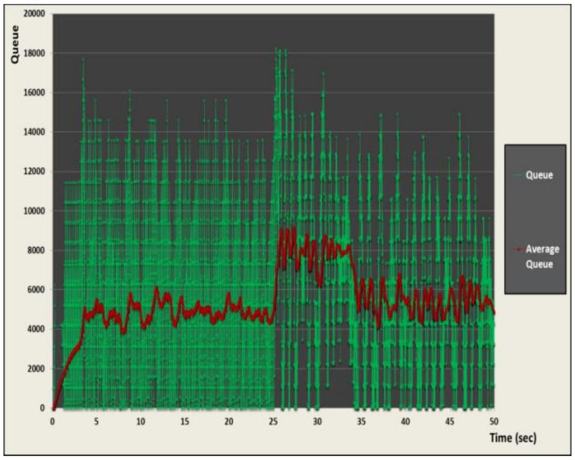

Figure 6. MRED with DDF with an increase avg in congestion

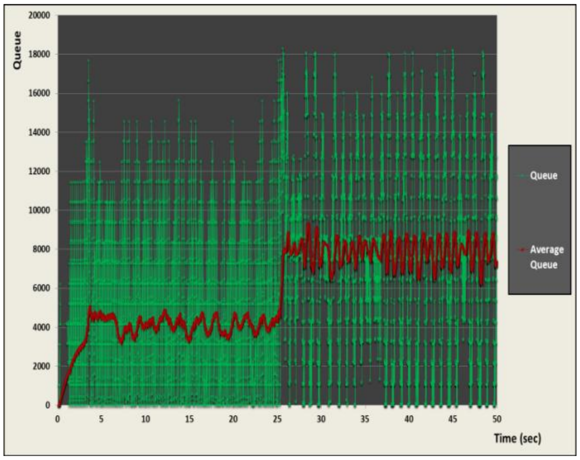

Figure 7. Normal RED with tail drop with an increase avg in congestion

E. Results Comparision

Four scenarios were compared in this study, starting from results for MRED with DDF with an increase of average queue size in congestion compared with results for normal RED with tail drop with an increase in congestion too; results for MRED with DDF with a decrease of average queue size in congestion compared with results for normal RED with tail drop with a decrease in congestion also. All those comparisons are based on the simulation metrics packet drop and throughput.

It can be observe in Figure 8 and 9 the comparison between the results for MRED with DDF and normal RED with tail drop in the same scenario with an increased of average queue size during the congestion. It has shown at the end of simulation lower percentage packet drop decrement 26% when used RED with DDF than in normal RED with tail drop, and throughput increment 1.56% when used with MRED with DDF than in normal RED with tail drop; it can be observed that when there are over flow in the queue the MRED with DDF makes the average queue size lower than MAXth by decreasing drop of jumbo frame encapsulation and just drop packets inside jumbo frame encapsulation during over flow in queue and increases the throughput. This means the proposed MRED with DDF technique achieved the objectives for decreasing the packet drop and increases the throughput with jumbo frame, which will be led to enhance the scalability and efficiency of jumbo frame networks.

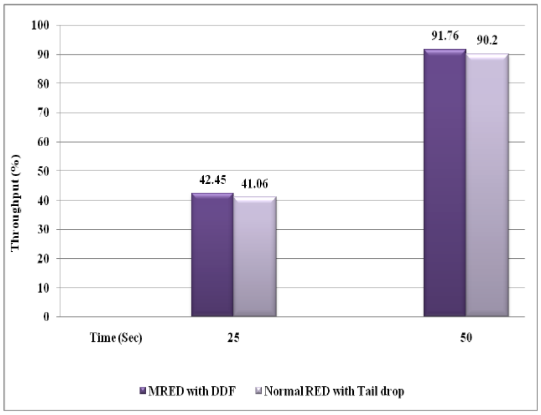

Figure 8. Packet drop rate between MRED with DDF and RED with tail drop in increase of congestion

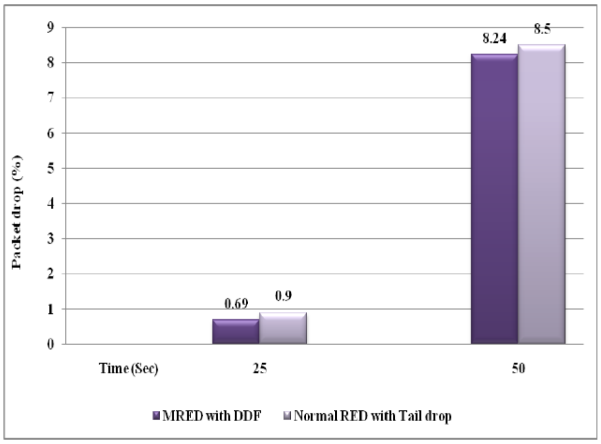

Figure 9. Throughput rate between MRED with DDF and RED wih tail drop in increase of congestion

IV. CONCLUSION

This work has been proposed new scheme in AQM with jumbo frame networks, by combined modified random early detection MRED with developed drop front DDF. The proposed algorithm help to reduce the packet loss in jumbo frame networks, and increase the throughput, by reduce the overhead and enhance the scalability and efficiency for jumbo frame networks. The proposed algorithm has been implemented by NS2 simulator, it have achieved the best results for reducing the packet loss at queue and increase throughput in jumbo frame environments when it compared with a result for applying the normal RED combined with drop tail technique in jumbo frame environments with the same metrics.

REFERENCES

[1] Alteon (1999) “Extended Frame Sizes for Next Generatinon Ethernets”, White paper, Lightwave technology journal, pages 66 -73.

[2] Balakrishnan, H. and Padmanabhan, V. N.,Seshan, S.,Stemm, S. and Katz, R. H. (1998) “TCP behavior of a busy interent server: Analysis and improvements”, INFOCOM'98. Seventeenth annual joint conference of the computer and communication societies.

[3] Chelsio communication white paper (2007), “Ethernet Jumbo Frames. The Good, the Bad and the Ugly” ITG fachbericht photonische netze journal.

[4] Chung, J. and Claypool, M. (2003), “Analysis Active Queue Management”, IEEE international symposium on network computing and applications conference.

[5] Dykstra, P. (1999), “Gigabit Ethernet Jumbo Frames, And why you should care”, White paper, WareOnEarth Communications and Available at: http://sd.wareonearth.com/phil/jumbo.html,1999.

[6] Floyd, S. and Jacobson, V. (1993), “Random Early Detection Gateways for Congestion Avoidance”, IEEE/ACM Transactions on Networking, pages 397-413.

[7] Floyd, S., Gummadi, R. and Shenker, S. (2001) “Adaptive RED: An Algorithm for Increasing the Robustness of RED’s Active Queue Management”, Preprint journal.

[8] Gass, R. (2004), “Packet size distribution” ACM SIGMETRICS performance evaluation review journal, pages 373.

[9] Genkov, D. and Llarionov, R. (2006), “Avoiding IP Fragmentation at the Transport Layer of the OSI Reference Model”, Proceedings of the international conference on computer systems and technologies – CompSysTech, University of Veliko Tarnovo, Bulgaria.

[10] Lakshmant, T. V., Neidhardt, A., Teunis, J. (1996), “The Drop from Front Strategy in TCP and in TCP over ATM”, Proceedings of the fifteenth annual joint conference of the IEEE computer and communications societies conference, pages 1242 - 1250.

[11] Ramakrishnan, K. and Floyd, S. (1999), “A proposal to add Explicit Congestion Notification (ECN) to IP”, IETF RFC 2481.

[12] Rekhter, Y. and Li, T. (1995) “A Border Gateway Protocol”, IETF RFC 1771.

[13] Sauver, J. S. (2003), “Practical Issues Associated with 9K MTUs”, University of Oregon computing center journal.

[14] Sathaye, S. (2009), “Jumbo Framescommons design”.

[15] Y n, N. and Hluchyj, M. G. (1990), “Implication of Dropping Packets from the Front of a Queue”, IEEE transactions on communications.