Implementation of An Autonomic Active Queue Management in Mobile Ad Hoc Network (MANETs)

G.Naveen, *M. Tech Student*, and Mr. D. Sharath Babu Rao, *Faculty*

Department of Electronics & Communication Engineering

Jawaharlal Nehru Technological University

Anantapur, India

Abstract

In order to provide better quality of service to the multimedia applications in Mobile Ad Hoc Network (MANETs), where its resource is limited and changed dynamically, in this paper we introduce an Active Queue management mechanism with autonomic attributes, named Autonomic Active Queue Management (AAQM). With the rapid development of network services and network technologies, the users requirements of network mainly the QoS are to improve. Whereas the traditional Active Queue Management (AQM) mechanisms are not adequate to provide QoS guarantee to multimedia video traffics effectively. AAQM tries to adjust its network behaviour and optimize the overall network performance according to network and service condition information.

1. Introduction

Autonomic network is a research hot spot in future architectures. By introducing the autonomic features to network entities we can solve the increasing complexity of network control and management. Therefore, we can introduce these autonomic mechanisms to Diffserv components, by that it can optimize the network performance by adaptively adjusting the network behaviour in accordance with the real-time condition information.

With the widely multimedia applications in MANETs, it is expected to provide the better quality of service. However, the present IP network is not capable to provide QoS to meet the requirements of real-time video flow. Packet losses have serious impact on compressed video streams. Protecting the video data from packet loss is an important concept in QoS field. But it is difficult to prevent the packet loss to a very low level using traditional packet dropping control technologies. Moreover, even a very small packet loss may damage the video stream quality [1]. A main problem is how to protect the perceived video quality in spite of existing packet loss.

These queue management mechanisms are relatively effective in reducing packet loss by dropping the packets randomly. But in MPEG4 encoded multimedia applications, there is no simple relation between packet loss and video quality perceived at the receiver. Due to the dependency between the video frames, even a small packet loss may cause many successively delivered packets with no use in practice, which will seriously impact the video quality. To reduce the loss of video stream delivery quality and effectively drop video packets, we will introduce autonomic mechanism to queue management mechanism, dynamically adjusting the discarding operation according to the network environment and video or service feature information.

There are two major approaches for to provide better QoS mainly for video applications. They are end system based approach and network based approach respectively. In the end system based approach, the provision of QoS is depends on end system. But it increases the overhead and complexity of the end systems greatly. In the network based approach, the provision of QoS is mainly depends upon the routers.

As a mechanism to support congestion control at intermediate routers, Sally Floyd [1] firstly proposed an default AQM scheme i.e., Random Early Detection. IETF RFC2309 [2] suggested to adopt AQM mechanism at routers, which become a research hotspot. The basic idea of AQM is to avoid congestion problem before the buffer is full by means of randomly dropping packets in advance.

2. Background

The concept of autonomic network is derived from autonomic computing and communication [3] & [9]. The main goal of this mechanism is to add autonomic attributes to network elements, simplifying the network management and control. Autonomic attributes such as self-configuration, self-adaptability, self-management, self-aware and self-optimization are realized based on feedback control loop. The autonomic mechanisms can dynamically adjust network behaviour in accordance with the real-time network environment and optimize the overall network performance.

Among the end system based and network based approaches, the end system based approach is relatively effective, but the complexity and overhead of end system are increased greatly. So, therefore it is expected to adopt the simple and effective queue management approach to provide QoS by network based approach. Most AQM mechanisms where they drop the packets randomly without differentiating their importance. The traditional queue management without considering the packets importance is not suitable for multimedia flow, which will damage the video flow quality severely.

In [4] & [6], a rate based RIO algorithm, named Rb-RIO, is proposed. It classifies packets of I frame, P frame and B frame to three priority classes. I frame packet has smaller drop probability compared with other packets. [5] proposed a priority dropping mechanism, which mapped I frame, P frame and B frame packets of MPEG4 hierarchical video flow architecture to different drop priorities of WRED, which is called REDN3.

3. Weighted Random Early Detection

Congestion avoidance techniques monitor network traffic loads in an effort to anticipate and avoid congestion at common network and inter-network bottlenecks before it becomes a problem. These techniques are designed to provide preferential treatment for premium (priority) class traffic under congestion situations while concurrently maximizing network throughput and capacity utilization and minimizing packet loss and delay. Congestion avoidance is achieved through packet dropping. Among the more commonly used congestion avoidance mechanisms is Weighted Random Early Detection (WRED), which is optimum for high-speed transit networks.

Weighted random early detection (WRED) is a queue management algorithm with congestion avoidance capabilities, where different queue (class) has different queue thresholds. Each queue threshold is associated to a particular IP precedence. For example: A queue may have lower thresholds for lower priority packet and higher thresholds for higher priority traffic. A queue build up will cause the lower priority packets to be dropped, hence protecting the higher priority packets in the higher priority queue.

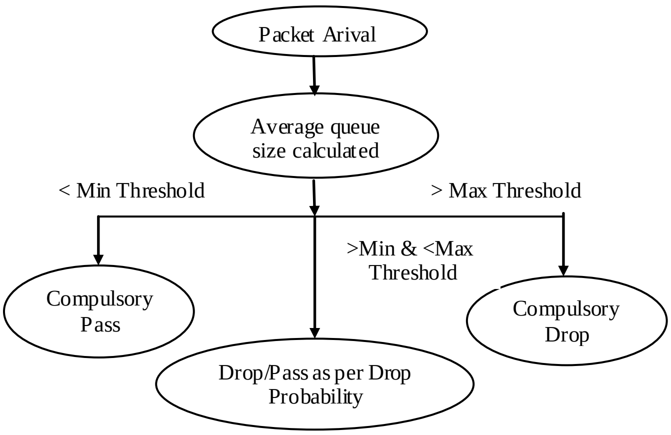


Figure 1. WRED Algorithm Flow Diagram

The packet drop probability is based on the minimum threshold, maximum threshold, and mark probability denominator. When the average queue depth is above the minimum threshold, WRED starts dropping packets. The rate of packet drop increases linearly as the average queue size increases until the average queue size reaches the maximum threshold. When the average queue size is above the maximum threshold, all packets are dropped.

Figure 2, depicts the graphical representation of packet drop probability vs. average queue size for two types of traffic (low priority traffic and high priority traffic). The low priority traffic has lower value for minimum and maximum thresholds compared to higher priority traffic. Hence at times of congested traffic, the drop rate for lower priority traffic will be higher compared to higher priority traffic. WRED parameters including minimum threshold, maximum threshold and drop profile can be adjusted as per the requirement of different types of traffic.

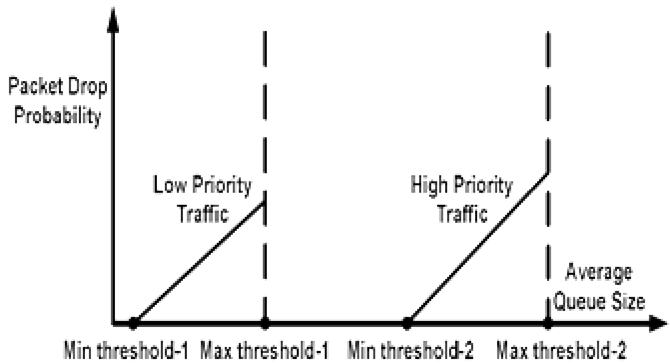


Figure 2. WRED Packet Drop Probability

The available queue management Weighted RED (WRED) [7] & [8] algorithms which is able to differentiate packet priorities. But the priorities are not simplified based on the video packets importance.

4. Autonomic Active Queue Management

The above queue management technique with differentiating video packet importance usually just takes into account the difference of frame type. Where it mapping the packets to different dropping priorities according to frame type and maintaining fixed parameters for respective priority may not leads to obtain desired optimal performance. We should make full use of the packet feature information caused by the video encode process to optimize the queue management operation.

So in order to overcome the above problem, we introduce the autonomic mechanism to our queue management design. Autonomic attributes mainly the self-adaptation and self-configuration are realized base on feedback control loop. Where it is based on context information collected. And the context information used in our mechanism includes network context and service context. We use the context information to dynamically configure and adjust the packet dropping operation.

The self-configuration and self-adaptation autonomic attributes are realized base on a feedback loop. In the feedback loop, the network and service context is collected through the collecting step. Then the configuration is determined based on the service context. And the congestion level is judged according to the network context information. Finally execute corresponding configuration and adjustment. The feedback loop will optimize the overall network performance and adaptively adjust the network behaviour .

The service context information is recorded in the packet header according to their video compression character by source end. When the packet travels in network, nodes are able to abstract the service context information directly from packet header. The service context information, for example video compress character information, such as frame type, frame situation and frame size are abstracted from the IP packet header, and it will determine the configuration of the queue management mechanism parameters. Network environment conditions will change with time dynamically. The length of node's queue is changing according to traffic rate and link bandwidth. The queue management mechanism will collect this network context information to judge the congestion condition. Network context information is monitored by wireless nodes.

4.1. Implementation of Source operation:

MPEG encodes video as a sequence of frames. Usually, video has a high degree of temporal redundancy, i.e., information in successive frames is highly correlated. Standard MPEG encoders generate three types of compressed frames (I, P, or B). An I frame is intra-coded, having no dependence on any other frames. Meanwhile, MPEG uses motion prediction and interpolation techniques to reduce the size of intermediate frames. Two types of motion prediction are used: Forward prediction, where a previous frame is used as a reference for decoding the current frame, and bidirectional prediction, where both past and future frames are used as a reference.

This technique provides a better compression. The encoding of P frames uses forward prediction and the encoding of B frames uses bidirectional prediction. As a consequence, the I frames are normally the largest in size, followed by the P frames, and finally the B frames.

The video sequence may be decomposed into smaller units which are coded together. Such units are called GOPs (Group of pictures). Each GOP holds a set of frames or pictures that are in a continuous display order. Usually GOPs are made independently decidable units to facilitate random access. Such GOPs are called closed GOPs as they contain all relevant decoding parameters so that it can be decoded independent of other units. If a GOP needs other GOPs for decoding, it is called an open GOP.

The GOP pattern specifies the number and temporal order of P and B frames between two successive I frames. Such a GOP pattern is characterized by two parameters: the I-to-I frame distance *(N)* and the I-to-P frame distance (*M*). This structure and the dependency for decoding of each frame in such GOP pattern is illustrated in Figure 3.

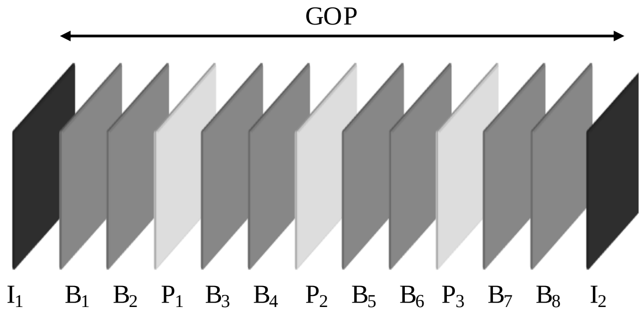


Figure 3. GOP Structure (N = 12 and M = 3)

The hierarchical structure of MPEG encoding with possible error propagation through the frames imposes a great difficulty on sending MPEG video streams over lossy networks. Small packet loss rates may translate into much higher frame error rates. For example, a 3% packet loss percentage could translate into a 30% frame error. This situation may seriously degrade the user perceived quality at the video reproduction. Moreover, network resources may be wasted with the transmission of information that becomes useless to the receiver. Some of the received data may become useless to the decoder as insufficient frame data is available for decoding. Such situation may occur either when there are losses in the network or when some frame in which the current frame depends on to be considered decodable is considered undecodable.

To improve the transfer of comprehensible information, we associate different levels of drop precedence to packets that carry information of different frames. Under this scheme, increasing drop precedence's are associated with packets from I, P, or B frames, respectively. That is, packets transporting fragments of a B frame are more likely to be discarded in a congested router than packets from P frames. Meanwhile, P frame packets have precedence in discard when compared to I frame packets.

We can get the frame importance comparison result in each GOP. I frame is the reference frame to all the frames in the GOP, so I frame has the highest importance. P frames at the front part has higher importance than the P frames at the rear part in each GOP. P and B frame is encoded with prediction encoding, its data amount almost reflects the approximation between the B frame and the prediction frame. Therefore, we can consider that B frame with larger size is more important than smaller one. In summary, the frames in each GOP ordered by importance from high to low are I frame, the P frame at the front part, the P frame at the rear part, B frame with large size, and B frame with small size respectively.

The service context information means the character information generated by the video compression, including video frame type, frame number, frame size and so on.

Precedence	Type of Packet	Precedence index(PI)
Class A	Packet carrying I frame data in GOP	001
Class B	Packet carrying the former part P frame data in GOP	010
Class C	Packet carrying the latter part P frame data in GOP	011
Class D	Packet carrying larger size B frame data in GOP	100
Class E	Packet carrying smaller size B frame data in GOP	101

TABLE 1: PACKET PRECEDENCCE MAPPING

The character information which is generated by the source video codec is recorded into the packet header. We add several fields into the IP header, including frame type, the number of P frame in GOP, the max P frame number, B frame size and the max B frame size, named f_type, f_seq, PN, f_size and BS respectively. For f_type filed, 0 means I frame, 1 means P frame, and 2 means B frame. The number of P frame is the P frame predicted sequence number in GOP, which is between 1 and PN. And the f_size field records the B frame size.

The source end divides the video packets into five priorities according to the video compression character information. Additionally we add another filed into IP header, named PI (priority index). The priority dividing table is shown in table I. The packet importance decreases from classA to classE.

4.2. Implementation of Queue management:

The collecting operation of autonomic loop monitors the queue length to estimate the network congestion situation. The average queue length and real time queue length are used as congestion metric. The interface queue of wireless node for video packets is a FIFO queue in our design. But in the internal queue structure, packets are directly mapping to five virtual queues (VQ) according to their priority indexes (PI) recorded by the source end. Packet drop priority increases from VQ1 to VQ5. The algorithm uses the highest drop priority dropping method. If packet dropping needed, the arriving packet will enter into its corresponding virtual queue, and the first packet of the highest drop priority virtual queue will be dropped. Parameters in the WRED queue Management algorithm are static. The parameter maxp is set as a fixed value. But in our design, the maxp is able to adjust according to the packet's service context Information. The original maxp for I, P and B frame are maxp_i, maxp_p and maxp_b respectively. But the maxp_p is able to be adjusted according to the P frame number. The maxp is adjusted according to P number as

maxp = maxp_i + (maxp_p – maxp_i) * a * (No./PN)

And the maxp_b is adjusted according to the B frame size, as

maxp = maxp_p + (maxp_b – maxp_p) * *b* * (PS/size)

Where a & b are small constant values.

If the average queue length q_{avg} is less than the small threshold min_{th}, then enter the arrival packet into the queue. If q_{avg} is between the two thresholds, then calculate a drop probability, enter the arrival packet, and discard the drop target packet with the probability. The probability calculated is the drop probability of the first packet of the highest drop priority virtual queue. If q_{avg} is larger than the large threshold max_{th}, then enter the arrival packet to its corresponding queue, and drop the first packet in the highest drop priority virtual queue.

PSEUDO-CODE OF AAQM ALGORITHM

Calculate the average queue length q_{avg} and get the current queue length q_{cur} mapping the arrival packet to corresponding virtual queue according to Precedence Index (PI).

//adjusting the maxp parameter & calculate the drop probability

1. if the arrival packet carrying I frame data
2. maxp = maxp_i
3. else
4. if the arrival packet carrying P frame data
5. maxp = maxp_i + (maxp_p – maxp_i) * *a* * (No./N)
6. else
7. maxp = maxp_p + (maxp_b – maxp_p) * *b* * PS/size
8. if ((q_{avg} >= max_{th}) || (q_{cur} = q_{lim}))
9. enter the arrival packet to its corresponding virtual queue drop the packet with highest drop priority
10. else
11. if (q_{avg} < min_{th})
12. enter the arrival packet into its corresponding virtual queue
13. else
14. p_b = maxp * (q_{avg} – min_{th}) / (max_{th} – min_{th})

15. randomize a number u,
16. if ($u <= p_b$)
17. enter the arrival packet to its corresponding virtual Queue drop the packet with highest drop priority
18. else
19. enter the arrival packet into its corresponding queue

Fixed Parameters:

Min_{th}: minimum threshold for queue
Max_{th}: maximum threshold for queue
Max_p: maximum value for p_b

Saved Variables:

q_{avg}: average queue size
q_{lim}: maximum value of queue size
p_b : Packet marking probability
q_{cur}: current queue size

5. Simulation Results

5.1. Simulation Specifications:

Simulator	: MATLAB
Topology	: MANETs topology
Number of nodes	: 10
Radio Transmission Range	: 300m
Simulation time	: 100 sec and 30 sec
Area of the Network	: 500m*500m
Routing Protocol	: DSDV

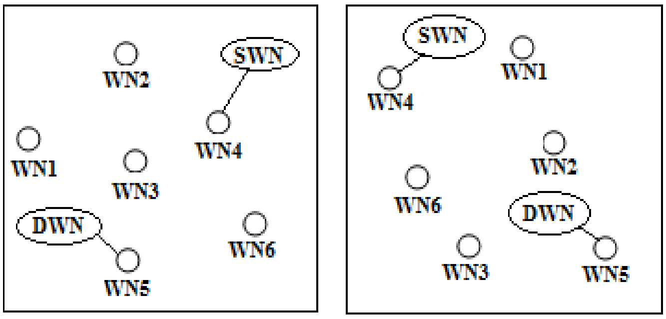
5.2. Simulation Results:

In this part, we use simulation to compare the performance of the proposed AAQM algorithm with WRED in MANETs scenario. The queue management algorithm is applied on the wireless interface queue.

There are 10 wireless nodes move randomly in a given 500x500 m2 square. Video sequences are in the 4:2:0 YUV format. We encoded the video into MPEG4 formatted file, and transmitted through the wireless network. Finally compare the file after transmission with the original file, and calculate the PSNR(Peak Signal-to-Noise Ratio) value. PSNR is one of the most widespread objective metrics to assess the application-level QoS of video transmissions. Node communication radius is set as 300m. Routing protocol uses DSDV.

The two queue management algorithms are simulated using the same parameter value. WRED is used as [5] described, video packets are divided into three priorities according to their frame type.

The simulation MANETs topology, including original topology and final topology after simulation, are illustrated in Fig.4. We send video flow and background traffic at wireless node 4. And wireless node 5 is the destination node. The moving tracks of the two nodes are also presented. For simplicity, we omit the other wireless nodes.



WN: Wireless Node SWN: Source Wireless Node
DWN: Destination Wireless Node

Figure 4: The original and final topology with the moving tracks of the source and destination nodes.

5.2.1. Scenario1:

We transmit the video flow from the source node to the destination node.

The two algorithms are simulated with the same scenario separately. And then analyze PSNR of the video. The PSNR results of .yuv formatted video file using AAQM and WRED are shown in Fig.5.

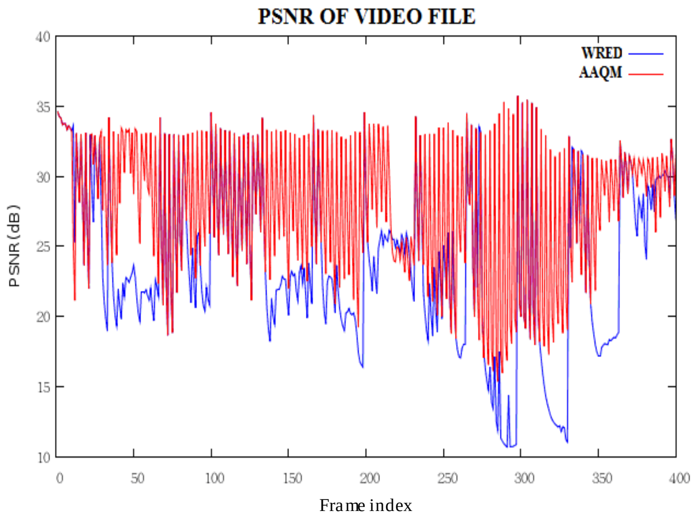


Figure 5: PSNR of video flow in scenario-1

At most time AAQM protects packets with higher importance effectively, avoiding the severe degradation of PSNR. The average PSNR of AAQM is 25.80dB, while WRED is 21.81dB.

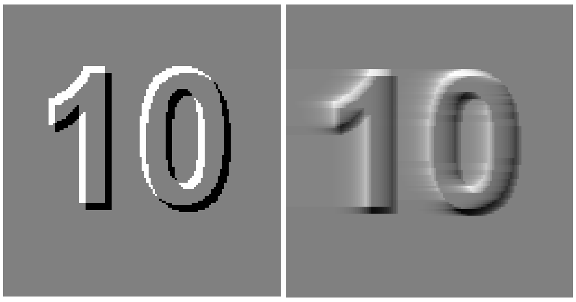


Figure 6: screen snapshots of AAQM (left) and WRED (right)

The comparison of screen snapshots of .yuv formatted video file is shown in Fig.6. Left is got by AAQM, and right is got by WRED. The average throughput of AAQM is 0.1402Mb/s, while the WRED is 0.1408Mb/s. When the throughputs of the two algorithms are almost the same, AAQM are obviously better than WRED. The loss of less important packet only has effect on a few frames near it. But the loss of important packet affects a series of frames. AAQM can protect more important video packets better than RED.

5.2.2 *Scenario2:*

In this scenario, we add background UDP flow between wireless nodes. We use a longer video sequence to analyze. The UDP background flow lasting from the start of simulation to the end with 4M sending rate, and the background video traffic is added at the intermediary time. The UDP background flow is assigned to the two higher priorities with proportion of 7:3 in WRED. While the background UDP flow is assigned to VQ1 to VQ3 with proportion of 7:1:2 in AAQM likewise.

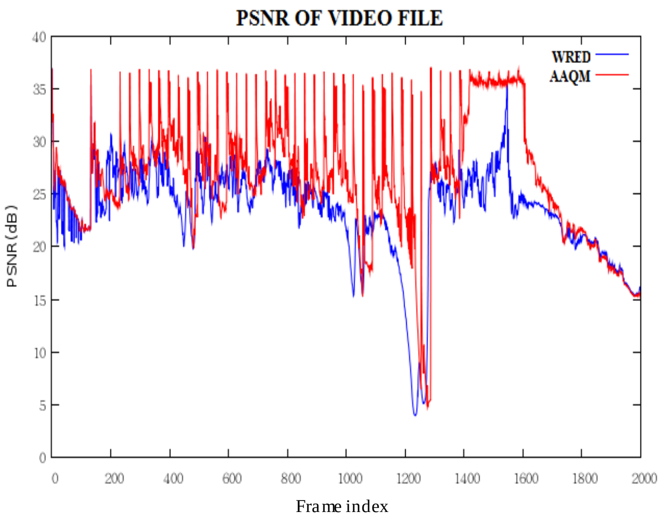


Figure 6: PSNR of video flow in scenario-2

The PSNR results of. yuv formatted video file using AAQM and WRED are shown in Fig.6. The average PSNR of AAQM is 24.12dB, and the average PSNR of WRED is 20.18dB.

Figure 7: screen snapshots of AAQM (left) and WRED (right)

The comparison of screen snapshots of .yuv formatted video file is shown in Fig.7. Left is got by AAQM, and right is got by WRED. From fig. 6 we find that, AAQM protects important video packets more effectively than WRED. The background video flow was added at the intermediate to add the traffic load. The AAQM has higher PSNR than WRED at the most time.

6. Conclusion

This paper proposes an autonomic Active Queue Management (AAQM) mechanism to improve the multimedia video flow delivery quality. We introduce autonomic attributes to queue management algorithm. The mechanism is capable of configuring and adjusting dynamically according to network and service context information. The simulation compared the performance of the proposed AAQM with WRED when transmitting MPEG4 formatted video flow. The result shows that AAQM can protect important video packets and reduce the impact of packet loss on video quality effectively

7. Acknowledgment

Naveen would like to thank Mr. D. Sharath Babu Rao, who had been guiding through out to complete the work successfully, and would also like to thank the HOD, ECE Department and other Professors for extending their help& support in giving technical ideas about the paper and motivating to complete the work effectively & successfully.

8. References

[1] S. Floyd, V. Jacobson. "Random early detection gateways for congestion avoidance". IEEE/ACM Transactions on Networking, IEEE/ACM, vol. 1, August 1993, pp. 397-413.

[2] B. Braden, D. Clark, J. Crowcroft, "Recommendations on Queue Management and Congestion Avoidance in the Internet", IETF RFC 2309, IETF, April 1998.

[3] J.Kephart, D. Chess, "The Vision of Autonomic Computing", IEEE Computer, vol. 36, January 2003, pp.41–50.

[4] J. Chung, M. Claypool. "Rate-Based Active Queue Management with Priority Classes for Better Video Transmission", Proceedings of the Seventh International Symposium on Computers and Communicaitons, ISC C, July 2002, pp.99-105.

[5] A. Ziviani, J.F. Rezende, "Improving the Delivery Quality of MPEG Video Streams by Using Differentitated Services". 2nd European Conference on Univeral Multiservice Networks(ECUMN) 2002.

[6] D.Clark, W. Fang, "Explicit allocation of best-effort packet delivery service", IEEE ACM Transactions on Networking, IEEE, vol. 6,August 1998, pp.362-373.

[7] Technical Specification from Cisco, "Distributed Weighted Random Early Detection", online available at http://www.cisco.com/univercd/cc/td/doclproducsoftware/io/wred.pdf

[8] U. Bodin, O. SchelCn, S. Pink, "Load-tolerant Differentiation with Active Queue Management", ACM Computer Communicaitons Review, ACM, vol. 30, July 2000, pp.4-16.

[9] S. Dobson, S. Denazis, etc., "A survey of autonomic communications", ACM Transactions on Autonomous and Adaptive Systems, ACM, vol. 1, December 2006, pp.223-259.

Authors Profile

G.Naveen received B.Tech Degree in Electronics & Communication Engineering from Madanapalle Institute of Technology and Science, Madanapalli, Kurnool, India. Presently he is pursuing his M. Tech in Digital Systems & Computer Electronics specialization in the Department of Electronics & Communication Engineering from Jawaharlal Nehru Technological University, Anantapur, India. His research interests include Wireless Communications, Computer Networks and Electronics and Communications.

Mr. D. Sharath Babu received his B.Tech Degree in Electronics & Communication Engineering from Sri Krishna Devaraya Engineering college, gooty, India. And he received his Master of Engineering (ME) from Osmania University, Hyderabad. Presently he is pursuing his Ph.D in "VLSI Implementation of Gigabit Ethernet MAC" in the Department of ECE, from JNTU Anantapur, India. He is presently working as a Lecturer in the Department of Electronics & Communication Engineering, JNTU University, Anantapur. His research interests include VLSI and Wireless Communications.