Packet Value Based Scheduling for Wireless Local Area Networks

Aruna Jayasuriya
CQUniversity
School of Engineering and Built Environment
Gladstone, Australia
Email:a.jayasuriya@cqu.edu.au

Anselm Teh
Institute for Telecommunications Research,
University of South Australia
Mawson Lakes, Australia

***Abstract*—Due to the popularity of realtime and streaming multimedia applications and technical improvements and availability of wireless networks, the demand for accessing realtime applications with strict Quality of Service (QoS) requirements over wireless networks has significantly increased in recent years. One important aspect of supporting QoS for such applications is the scheduling of Variable Bit Rate (VBR) flows in a dynamic network. In this paper we propose a method to define a time dependant value of a packet, based on importance of the packet to the stream when it will delivered to the destination. We then propose a scheduling scheme based on the packet value. Using an OPNET based simulation study we then show that the proposed scheduler can decrease the number of packets with excessive delays, while meeting other QoS parameters such as jitter, loss and goodput.**

I. Introduction

The popularity of realtime communication over the Internet has risen sharply over the last few years, as Voice-over-IP (VoIP) and video conferencing provide a relatively inexpensive and ubiquitous method to stay connected around the world. Due to their interactive nature, realtime applications have strict Quality of Service (QoS) requirements. Even transient fluctuations in resource availability can lead to degradation of user satisfaction. With the increasing popularity of wireless technologies such as WiFi and WiMAX around the globe, it is imperative that these wireless access networks address the challenges in meeting the QoS requirements for multiple realtime flows.

Many realtime applications generate Variable Bit Rate (VBR) traffic. Scheduling methods designed to provide fair throughput for mainly constant bitrate traffic sources are often not capable of meeting the QoS requirements of realtime applications. The total bitrate of a set of VBR flows may fluctuate greatly, and there are times when the generated traffic may exceed the medium capacity, causing delays. Common techniques that aim to provide a fair division of bandwidth, such as Round Robin, were not designed with realtime flows in mind and cannot constrain packet delay when bitrate varies quickly. While general realtime scheduling methods do better in this regard, their performance can be improved by tailoring the scheduling algorithms for use with realtime communication traffic. In this paper we propose a method to schedule packets based on their importance to the application output, and show that this can greatly reduce the number of realtime packets excessively delayed in a local wireless network, while keeping jitter and packet loss within acceptable limits.

Rest of this paper is organised as follows. Section II provides an overview of different methods that have been used to schedule realtime flows. This is followed by the proposed delay dependent packet value mechanism and an scheduling algorithm based on packet value. Section IV presents the simulations used to evaluate the performance of the proposed scheduling algorithm. The last section presents the concluding remarks.

II. Scheduling Realtime VBR Traffic

To ensure that a network can provide good Quality of Experience (QoE) for realtime applications, Quality of Service (QoS) must be tightly controlled. In this regard, end-to-end delay must be kept below 150 ms, jitter below 30 ms, and packet loss below 1 % [1][2]. With multiple VBR traffic flows generating traffic in a dynamic wireless network, it can be difficult to schedule traffic to ensure that these QoS thresholds are met for each realtime flow. In this section we look at some existing scheduling methods and how they fare in scheduling multiple VBR flows in an 802.11e network.

In each simulation in this section, we consider a wireless infrastructure network where each of the flows in the network has the same priority. The traffic sources in the network include MPEG flows of differing traffic characteristics, with bit rates with different means and variances. Each flow has a packet generation interval of 40 ms, and a Gamma distributed [3] packet size with characteristics from trace information in [4]. The mean and standard deviation of the flow are shown in Table I.

TABLE I
Simulation traffic flow statistics

	Mean Bitrate (bits per 40 ms)	Bitrate SD (bits per 40 ms)
low rate	10300	401
medium rate	21000	11000
high rate	44300	22700

In our simulations we aim to constrain delay jitter and packet loss to lower thresholds than the end-to-end values mentioned above. For the local network, we use thresholds of 40 ms delay, 10 ms jitter, and 0.5 % loss. We also aim to keep the percentage of packets suffering a delay over 40 ms to below 1%.

The following simulations were performed in OPNET, using an 802.11e model with an 11 Mbps Physical layer. Only the Hybrid Coordination Function (HCF) Channel Access (HCCA) contention free access method was used [5], and beacons were disabled to accurately measure and compare scheduler performance when using different scheduling methods. All flows had start times exponentially distributed with a mean of 80 ms, Each simulation is run for 200 s and is repeated three times using three different seeds. The results are then averaged. Results were collected only after the system has reached steady state.

The scenario we examine in this section comprises 4 low rate flows, 3 medium rate flows, and 7 high rate flows. This scenario has a utilisation of 94.1%. The predicted percentage of packets delayed beyond 40 ms is 6%. This value is an estimation of the percentage of total packets delayed over a maximum delay bound in the worst case scenario, from [6]. It is based on the assumption that each flow sends packets at the same time, at the beginning of each scheduling interval. For a scenario where flows have random start times, we expect the average delay to be lower. In these simulations, no packets are dropped due to buffer overflow or physical layer collisions.

We first examine the performance of the Weighted Round Robin (WRR) scheduler in scheduling realtime flows in this network. The WRR scheduler is proportionately fair in terms of throughput, and works by stepping through the list of flows in the network and polling each one in turn. If a station is known to have no data ready to send, e.g. through monitoring of bandwidth requests, it is not polled. Once the end of the list is reached, the scheduler starts again from the beginning. In the simulation, each Transmission Opportunity (TXOP) is weighted by the current queue size information—the scheduler allocates a TXOP equal to the queue size, up to the maximum MAC Service Data Unit (MSDU) size of 2304 bytes. If a queue has size greater than 2304 bytes, it can send only one maximum size MSDU in each TXOP. In each of our simulations we used fragmentation as it is commonly used when dealing with delay sensitive flows. Table II presents the simulation results. The average percentage of packets delayed is 4.4%, below the predicted critical instance value of 6%.

TABLE II
WRR AND EDF SIMULATION RESULTS - SCENARIO: 4 LOW RATE, 3 MEDIUM RATE, 7 HIGH RATE

	WRR	EDF
Packets delayed (%)	4.4	1.2
Mean jitter (ms)	4.4	1.0
Mean delay (ms)	6.8	10.6
Mean total goodput (Mbps)	6.4	6.4

Figure 1 shows a graph of the MAC end-to-end delay for each MSDU received at the QoS enabled Access Point (QAP). The delay is measured from the time a packet arrives at the source queue at a QoS Station (QSTA) until it is received at the QoS Access Point (QAP) MAC layer. In each of this type of graph in the paper, the station addresses are set in order of flow rate. The 4 low rate flows have addresses 1–4, the 3 medium rate flows have addresses 5–7, and the 3 high rate flows have addresses 8–14. Each dot in the graph represents the delay value for a packet. Each point on the graph shows the delay of a single packet in one of 14 stations, randomly distributed within the width of the corresponding bar. The density of each bar at a given delay value provides an indication of the likelihood of packets suffering that delay. From Figure 1, we can see that the high rate flows have more packets delayed beyond the threshold than the low rate flows. This is because they have more data to send, and the flows are not polled in any order related to the delay these flows have already suffered.

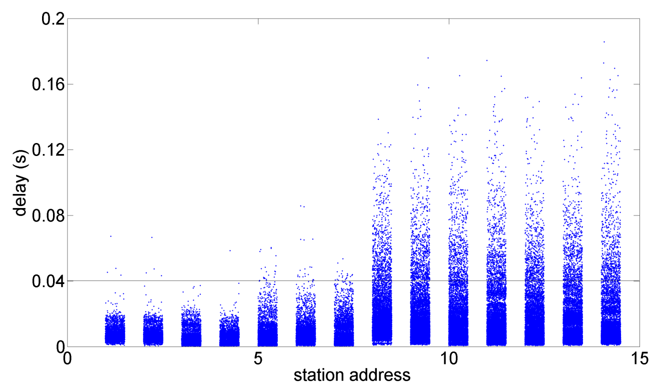

Fig. 1. RR packet delay graph - 4 low rate, 3 medium rate, 7 high rate

Earliest Deadline First (EDF) (also known as Earliest Deadline Driven (EDD)) [7] is a well-known realtime scheduling algorithm, which processes the tasks in order of the times by which they are required to be completed. Whenever a task has been completed, the scheduler looks through the list of waiting tasks and selects the one with the earliest deadline to be processed next. In a single processor system, for any set of tasks where all tasks can be processed by their deadline time, the EDF scheduler is able to schedule these tasks so that they are completed by their deadline. If the set of flows cannot be completed by their deadline, the performance of the EDF scheduler becomes unpredictable [8]. A related deficiency of the EDF scheduler is that the EDF algorithm does not take into account the time it takes for each job to be processed; the EDF scheduler may assign the highest priority to tasks that have no chance of completing on time [8].

Table II shows the simulation results for the EDF scheduler. It shows that EDF scheduler performs better than WRR in terms of average percentage of packets delayed, average jitter, and average packet delay. Figure 2 shows the spread of delay values for EDF scheduler. We can observe in Figure 2 that EDF has a more even delay spread than WRR, and lower peak delay.

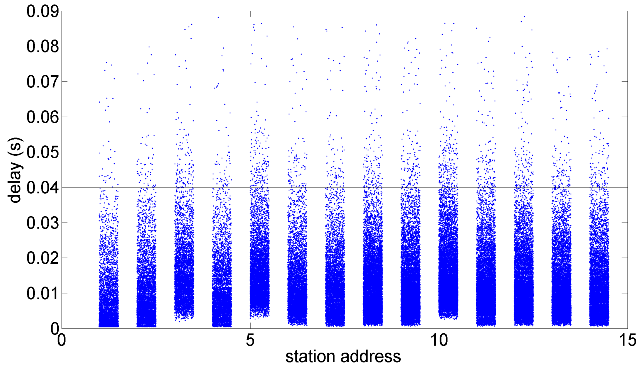

Fig. 2. EDF packet delay graph - Scenario 4 low rate, 3 medium rate, 7 high rate

From these results we conclude that although the EDF scheduler is better than the WRR scheduler at meeting the flow requirements, still a substantial number of packets are delayed beyond the 40 ms threshold. As the EDF scheduler allows transmission of waiting packets in order of deadline as soon as they are able to be sent, the high delays occur when the amount of data generated exceeds the capacity of the network. As the scheduler does not take the execution time of a task into account or calculate if it is possible to finish the task before deadline, the EDF scheduler may continue to schedule packets in order of deadline even when they cannot be scheduled on time.

As the traffic generated by a set of VBR flows is expected to fluctuate, there will often be excessively delayed packets in a moderately utilised network. Each time a packet is delayed excessively, this causes the next packet in the schedule to be delayed, and so on. If we also take into account the fact that wireless networks are dynamic in nature, it can be surmised that a method to ameliorate the delays caused by these late packets may improve the performance of realtime flows. We propose a method of assigning a "value" to realtime packets based on their delay requirements and scheduling packets based on maximising the value delivered by the scheduler.

III. Delay Dependent Packet Value

To place an initial value on a packet arriving at the MAC layer, we must consider its importance to the quality of the application output. While it is generally true that packet size is proportional to the information it contains, this may not directly relate to QoE for the user. Because of the interrelated nature of video frames and the complexity of understanding speech, one packet may be more important than another to understand a word or prevent artifacts in a video stream without being significantly different in other ways. Due to this fact, we cannot easily place a value on packet data as it is serviced at the MAC layer. Existing schemes work towards classifying specific packets at a higher layer to allow for intelligent scheduling and prioritisation at lower layers. Packet classification schemes for this purpose exist for MPEG video [9] and VoIP streams [10][11]. Results from work in [10], [11] show that source-driven packet marking schemes outperformed techniques that did not use source information. We hence conclude that higher layer knowledge of packet properties is required to evaluate the importance of a single packet. Because of this, aside from flow priority, we should not place a value on each packet as it enters the MAC layer. If schemes such as those mentioned above are used at the source node, we can use this information to provide an accurate initial value for each packet. If no such mechanism is used at the source, we should treat all packets in a flow as having an equal value according to the priority of the flow.

After the initial values of packets in a queue are decided, we must consider the value of packets over time. Realtime flows should adhere to specific delay, jitter, and packet loss thresholds to ensure a good quality experience for the user. However, if a QoS threshold is exceeded by a packet, it does not necessarily mean that the packet has zero value. In terms of delay, when a packet has passed its deadline it may still provide useful information to the application stream, although the quality of experience at the output may drop. The packet still contains useful information as the delay increases, until such time that sending the packet would be of no use to the realtime application, or the processing of the packet at the destination would result in a poor output; in which case the packet should be dropped rather than wasting bandwidth to complete the transmission. Modern VoIP and video codecs incorporate various mechanisms to cope with a certain percentage of lost packets [12][13]. These loss recovery techniques can handle a small amount of loss without any noticeable decrease in QoE, while transient delay spikes are more difficult to handle. Large delay spikes can result in the packet being dropped at the destination if the delay jitter is larger than the jitter buffer. When a packet is dropped, the freed bandwidth can be used to transmit queued packets on time that might otherwise have been delayed, keeping delay, jitter, and loss within acceptable ranges.

We propose a packet valuation system as follows. If a realtime application does not use a specific method to mark the value of a packet, its initial value depends on the priority of the flow. Each traffic flow has two associated delay thresholds. The first threshold D_1 is the point after which a packet's value begins decreasing. From a QoE point of view, a packet should be delivered before this time to provide the best result. After D_1 is passed, the value of the packet decreases linearly until it reaches zero value at threshold D_2. D_2 is the point beyond which receiving the packet will not improve the QoE of the application. Once a packet has been delayed more than D_2, it should be dropped.

In terms of implementation, the values of D_1 and D_2 can be provided through the QoS parameter list submitted upon the admission of any flow to the network. When calculating the delay of a packet to determine its scheduling value, delay should be calculated as the time from when the packet was received at the queue, until the time when transmission would be completed if the task was polled immediately, i.e. the virtual finishing time. In this paper we assume that the default initial packet value for each flow is one. We find the value of a packet i as follows:

1) Calculate the delay d_i for the virtual finishing time of packet i. Let current time be t, transmission duration be Tx_i, and overheads duration be O_i. Then:

$$d_i = t + Tx_i + O_i \quad (1)$$

2) Calculate the value v_i of the packet based on d_i and the thresholds D_{1i} and D_{2i}:

$$v_i = \begin{cases} 1 & \text{if } 0 \leq d_i \leq D_{1i} \\ 1 - \frac{d_i - D_{1i}}{D_{2i} - D_{1i}} & \text{if } D_{1i} \leq d_i \leq D_{2i} \\ 0 & \text{if } D_{2i} < d_i \end{cases} \quad (2)$$

Where flows have different default values or where packets are assigned different initial values at a higher level, v_i can be used as a value multiplier to alter the packet value according to delay. To apply this method of calculating packet value in a dynamic realtime scheduler, we use the method as shown in Figure 3.

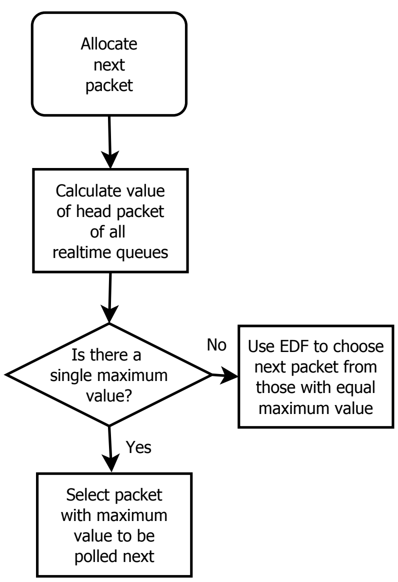

Fig. 3. Value base scheduler operation

We first calculate the value of the head packet of the queue for each flow in the network, then select the packet with the maximum value to be polled. If there is more than one packet with the same maximum value, we select the packet with the earliest deadline. Any packets that have zero value at the time of scheduling are dropped. In this case, the dropping of uplink packets that have been delayed past the D_2 threshold is done spontaneously on the mobile stations, and may be done through setting the value of the *dot11MSDUlifetime* field in MAC frames. This scheduler assumes that an admission control mechanism is in place to ensure that the network is not overloaded.

IV. Simulations and Results

The simulation environment and parameters used for this study are the same as used for the simulations described in Section II. We refer to the proposed value based realtime scheduler described in Section III as Modified EDF (Mod EDF) in the rest of this paper. In both the EDF and Mod EDF case, packets are dropped when reaching the D_2 threshold. $D_1 = 40$ ms and $D_2 = 60$ ms for all flows. The value of D_2 was chosen to be between D_1 and $2D_1$, so that any late packets are dropped before they cause the packet next in line in the queue to be delayed more than D_1.

Table III compares the performance of the new scheduling method, labeled Mod EDF, with EDF. Mod EDF shows a substantial reduction in the proportion of packets delayed as compared to EDF. EDF produces a proportion of packets delayed slightly above our 1% threshold, while Mod EDF produces a value well within the threshold. Mean delay is slightly lower in Mod EDF, and jitter is very similar in both methods. These results are well within our target jitter range. From the goodput results we notice that there is no difference in goodput between the scheduling methods, hence the differences in QoS are not due to a drop in network utilisation.

The mean packet drop percentage for EDF and Mod EDF are low, and well within the acceptable range for realtime flows. Although Mod EDF has a lower percentage of packets delayed, it has a higher packet drop percentage than EDF. As EDF prioritises packets by deadline even when late, the maximum delay is kept low and fewer packets reach D_2 and are dropped. On the other hand, Mod EDF actively delays late packets to give priority to packets that can be sent on time. These late packets are dropped if there is no free time for them to be sent before D_2. In summary, the EDF scheduler tends to send delayed packets after the deadline, while Mod EDF scheduler tends to delay and drop them, allowing more time for on-time packets to be sent. At this load, the Mod EDF scheduler produces a more desirable result than EDF.

TABLE III
EDF and Modified EDF simulation results

	EDF	Mod EDF
Packets delayed (%)	1.1	0.001
Mean jitter (ms)	1.1	1.0
Mean delay (ms)	10.5	9.9
Mean drop percentage	0.01	0.1
Goodput (Mbps)	6.4	6.4

Figure 4 shows the delay of each MSDU during the Mod EDF simulation. We can see that there are much fewer packets above the delay threshold. To further examine simulation results between EDF and Mod EDF for multiple scenarios, we used our admission control method described in [6] to select 10 different scenarios, with a predicted worst case percentage of delayed packets from 1–10%. Each of these scenarios was run with three different seeds for a simulation time of 200 seconds, and the results were then averaged.

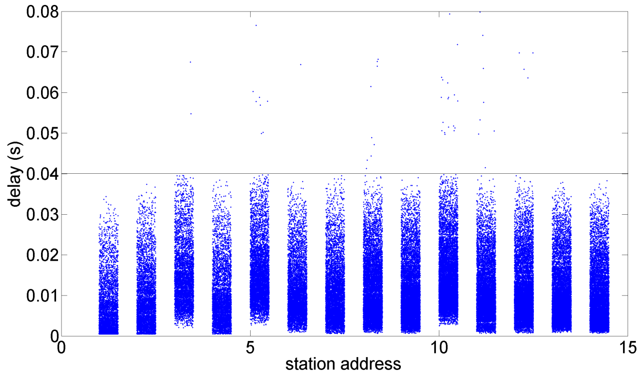

Fig. 4. Mod EDF with Drop packet delay graph - Scenario: 4 low rate, 3 medium rate, 7 high rate

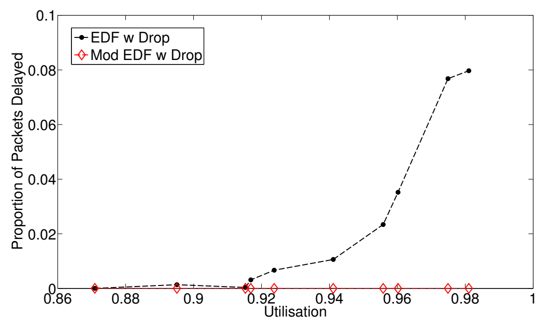

Fig. 5. Proportion of delayed packets

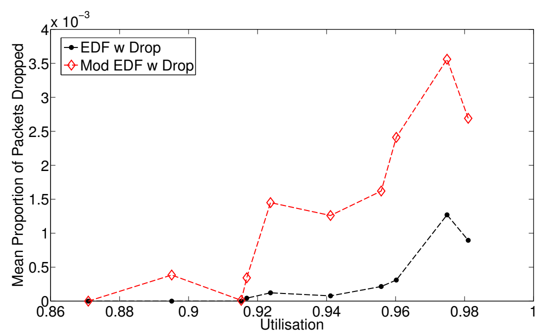

Fig. 6. Proportion of dropped packets

In the following figures, each point represents the results from a different scenario. Figure 5 shows the mean proportion of packets delayed against utilisation for each scenario. The number of packets delayed past deadline increases as utilisation increases. Comparing EDF and Mod EDF, we see that the proportion of packets delayed in EDF increases much faster than the Mod EDF results.

Figure 6 shows the proportion of packets dropped against utilisation. The mean drop ratio for Mod EDF increases faster than EDF, however both remain below the acceptable limit of 0.5%.

We can also observe in Figures 5 and 6 that the relationship between the performance parameters and utilisation is sometimes erratic. This is because each scenario contains a different combination of flows which produce different packet size distributions even if they have similar means and variances. As the instantaneous size of a packet directly relates to its processing time, different combinations of flows affect scheduler operation and hence delay, even if the two scenarios have similar means and variances.

V. Conclusion

In this paper we have proposed a simple method to valuate packets, allowing packets from realtime flows to be scheduled effectively as network load changes. We define the packet value to be constant until its delivery deadline. Once the delivery deadline has expired the packet value will decrease linearly until it reaches zero; beyond this delay, delivering the packet does not add any value to the stream. When scheduling packets the scheduler will select the packet that has the highest value at that time. We have shown through OPNET simulations that the proposed scheduler can decrease the number of packets delayed excessively, while keeping other QoS parameters such as jitter and loss within the set thresholds.

References

[1] *One-way transmission time*, ITU-T Std., 1996.

[2] T. Szigeti and C. Hattingh, *End-to-End QoS Network Design: Quality of Service in LANs, WANs and VPNs*. Indianapolis, Indiana: Cisco Press, 2005.

[3] S. Domoxoudis, S. Kouremenos, V. Loumos, and A. Drigas, "Measurement, modelling and simulation of videoconference traffic from VBR video encoders," *Second International Working Conference, HET-NETs 04*, Jul 2004.

[4] F. Fitzek and M. Reisslein. MPEG-4 and H.263 video traces for network performance evaluation. TKN, Berlin University of Technology. [Online]. Available: http://www-tkn.ee.tu-berlin.de/research/trace/trace.html

[5] *Part 16: Air Interface for Fixed Broadband Wireless Access Systems, Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands*, IEEE Std., 2005.

[6] A. Teh, A. Jayasuriya, and P. Pudney, "Admission control in wireless infrastructure networks based on the predicted percentage of delayed packets," in *Communications, 2008. APCC 2008. 14th Asia-Pacific Conference on*, Oct. 2008, pp. 1–5.

[7] J. A. Stankovic, M. Spuri, K. Ramamritham, and G. C. Buttazzo, *Deadline Scheduling for Real-Time Systems: EDF and Related Algorithms*. Springer, 1998.

[8] R. K. Abbott and H. Garcia-Molina, "Scheduling real-time transactions: A performance evaluation," *ACM Transactions on Database Systems*, vol. 17, pp. 513–560, Sep 1992.

[9] W. Tan and A. Zhakor, "Packet classification schemes for streaming MPEG video over delay and loss differentiated networks," *Proc. IEEE Packet Video Workshop*, Apr 2001.

[10] J. De Martin, "Source-driven packet marking for speech transmission over differentiated-services networks," *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, vol. 2, pp. 753–756 vol.2, 2001.

[11] H. Sanneck, N. T. L. Le, A. Wolisz, and G. Carle, "Intra-flow loss recovery and control for VoIP," in *MULTIMEDIA '01: Proceedings of the ninth ACM international conference on Multimedia*. New York, NY, USA: ACM, 2001, pp. 441–454.

[12] B. Wah, X. Su, and D. Lin, "A survey of error-concealment schemes for real-time audio and video transmissions over the internet," *Multimedia Software Engineering, 2000. Proceedings. International Symposium on*, pp. 17–24, 2000.

[13] C. Perkins, O. Hodson, and V. Hardman, "A survey of packet loss recovery techniques for streaming audio," *Network, IEEE*, vol. 12, no. 5, pp. 40–48, Sep/Oct 1998.