Q-DRAM: QoE-based Dynamic Rate Adaptation Mechanism for Multicast in Wireless Networks

Kandaraj Piamrat*, Adlen Ksentini†, Jean-Marie Bonnin‡, and César Viho†

*INRIA Rennes-Bretagne Atlantique, Campus Beaulieu 35042 Rennes, France
Telephone: (33) 299847137, Fax: (33) 299847171, Email: kandaraj.piamrat@inria.fr
†IRISA/University of Rennes I, Campus Beaulieu 35042 Rennes, France
Telephone: (33) 299847416, (33) 299847142, Fax: (33) 299847171, Email: {adlen.ksentini, cesar.viho}@irisa.fr
‡Institut Telecom/Telecom Bretagne, CS-17607, 35576 Cesson-Sévigné, France
Telephone: (33) 299127007, Fax: (33) 299127030, Email: jm.bonnin@telecom-bretagne.eu

Abstract—The deployment of real-time and multimedia applications over Wireless LAN has gained a growing interest in this last decade. These applications, e.g. video streaming, require tight guarantee on Quality of Service (QoS). Also, they use multicast communication in order to reduce the bandwidth consumption. However, in WLAN, multicast packets are sent with the basic rate (lowest rate); which results in capacity wasting because of longer channel occupancy. Moreover the lack of feedback mechanism makes it difficult to deal with reliability or service quality. In this paper, we propose to rely on the WLAN multi-rate capability, in order to transmit multicast packets with higher and dynamic rate than the basic rate. Unlike other existing protocols that use a static-threshold to decide when to change transmission rate, we propose a novel dynamic rate-adaptation mechanism based on Quality of Experience (QoE), namely Q-DRAM. According to the clients’ feedback on QoE, we adapt the multicast rate: (i) when users had bad QoE we reduce the multicast transmission rate; (ii) when users had good QoE, we increase the multicast transmission rate. Simulation results show that Q-DRAM increases the wireless channel utilization and maximizes users’ QoE, compared to existing solutions as well as to the IEEE 802.11 standard.

I. INTRODUCTION

The rise of wireless communications has pushed research and development in this area to grow up very quickly. IEEE 802.11 [1] based wireless communications have been widely deployed. Commercial products and numerous access networks are available. Moreover, the standard has provided many specifications for the deployment of wireless networks; one of which is the multi-rate transmission capability provided by 802.11 physical layers. For example: 1, 2, 5.5, 11 Mbps data rates are available in IEEE 802.11b [2]; or 6, 9, 12, 18, 24, 36, 48, 54 Mbps are also available in IEEE 802.11g [3]. These different data rates come from different modulation techniques and channel encoding schemes; for example, in IEEE 802.11b, DBPSK (Differential Binary Phase Shift Keying), DQPSK (Differential Quadrature Phase Shift Keying), CCK (Complementary Code Keying) 5.5 and CCK 11 correspond to data rate of 1, 2, 5.5, 11 Mbps respectively. In wireless environment, different factors such as path loss, fading, or interference in the channel have direct impact on the variation of the received signal to noise ratio (SNR), which results in variation in Bit Error Rate (BER). The lower the SNR the more difficult it is for the modulation scheme to decode the received signal, resulting in higher BER; hence the need of rate adaptation.

In this paper, we are interested in deploying multi-rate feature in multicast transmission. It can be noticed that multicast over wireless networks is a fundamental communication function because wireless network is inherently broadcast by nature. This means that a packet that is sent only once, will reach all intended recipients in multicast group. Therefore, multicast is an efficient method to transmit the same data to a group as it allows transmission of data to multiple destinations using fewer network resources. More recently, the fast-growth of wireless network and its application has pushed the deployment of multicast communication over wireless networks. As we can see, various applications support multicast; for example, conference meeting, mobile commerce (mobile auctions), military command and control, distance education, entertainment service, and intelligent transport system.

However, multicast application has some constraints. Multicast traffic has been set to the lowest transmission rate (basic rate) in order to reach all mobile nodes especially the further ones because they are subject to important signal fading and interference. The lower rates disadvantage transmission in terms of channel occupancy since they take longer time than the higher rates to send the same amount of information. This performance anomaly has been presented in [4], it is mentioned that slow host may considerably limit the throughput of other hosts roughly to the level of lower rate. Another constraint in multicast transmission is the lack of acknowledgment and retransmission due to huge amount of traffic overhead these packets will generate. This is severe when transmission mode is multicast because the number of acknowledgment/retransmission will be multiplied by the number of recipients in the multicast group; this could cause collision due to feedback implosion.

The lack of feedback results in two main disadvantages, firstly it is more difficult to know the current situation of the mobile node without feedback mechanism. Therefore, many of the schemes insist the use of feedback mechanism; for example, they make use of RTS/CTS (Request/Clear To Send) frames or channel probing mechanism. Secondly no feedback means no recovery from the loss or error; this makes multicast transmission unreliable. Some researchers have proposed reliable multicast protocols such as [5] or [6]
to deal with unreliability issue in multicast. In this paper, we do not focus on this problem since we assume that for real-time traffics like UDP-based streaming, reliability is not a crucial issue. It is preferable to lose a few packets than waiting for retransmission, which delays packets delivery. Hence, we focus here on the performance of the network in respect of user satisfaction and network utilization.

As to improve performance of multicast transmission, we apply rate adaptation mechanism using quality of experience as indicator for rate selection. Quality of Experience or QoE [7] is the overall acceptability of an application or service as perceived subjectively by the end-user. It is a new concept, which is more appropriate when dealing with multimedia service such as video or voice over IP. With these types of applications, quality of service is hardly determined only by technical parameters such as BER, SNR, etc. It makes more sense to evaluate quality by users’ opinion on their perception of the application; that is why it is called Quality of Experience. It can be evaluated in terms of Mean Opinion Score (MOS) as shown in Table I. As can be noticed, it is difficult to ask people to evaluate the score and then adjust the transmission rate in real-time. The evaluation procedure is very complex and time-consuming, it also needs manpower. Thus, it is not practical for real-time usage. For these reasons, in our scheme we use Pseudo Subjective Quality Assessment (PSQA) [8], a real-time QoE assessment tool based on Random Neural Network, to evaluate QoE from which our protocol adapts transmission rate accordingly.

<table>
<thead>
<tr>
<th>MOS</th>
<th>Quality</th>
<th>Impairment</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Excellent</td>
<td>Imperceptible</td>
</tr>
<tr>
<td>4</td>
<td>Good</td>
<td>Perceptible but not annoying</td>
</tr>
<tr>
<td>3</td>
<td>Fair</td>
<td>Slightly annoying</td>
</tr>
<tr>
<td>2</td>
<td>Poor</td>
<td>Annoying</td>
</tr>
<tr>
<td>1</td>
<td>Bad</td>
<td>Very annoying</td>
</tr>
</tbody>
</table>

The rest of this paper is organized as follow. We first give a comprehensive background on related works in section II. Then we continue with our proposed scheme in section III, in which we explain our strategy, the corresponding algorithm, and PSQA tool. Section IV demonstrates the result obtained by our mechanism. Finally, we end this paper with conclusions and future works in section V.

II. RELATED WORKS

In this section, we begin by giving some backgrounds concerning firstly rate adaptation mechanisms in IEEE 802.11 for unicast transmission. Then we continue with rate adaptation mechanisms designed for multicast transmission.

A. Rate adaptation mechanisms in IEEE 802.11

The first and widely used rate adaptation protocol in commercial products is Auto Rate Fallback (ARF) [9] proposed by Kamerman and Monteban. In ARF, when SNR decreases, an access point tries to recover by decreasing the bandwidth. In fact, the access point switches to a higher rate when a certain number (ten) of packets has been successfully received; it switches back to the lower rate when a failure occurs right after rate increase. If a failure occurs when the number of consecutive successful transmissions is less than ten, the access point switches to a lower rate only after two consecutive failures. Regardless of its wide implementation in commercial products, the protocol has a drawback resulting from the static-threshold approach, which does not adapt well to varying condition in wireless networks.

To solve disadvantages from static-threshold approach, Lacage et al. have proposed Adaptive ARF (AARF) [10]. The authors also use threshold-based mechanism as in ARF but instead of setting it to a fixed number, the threshold follows binary exponential backoff continuously at runtime to better reflect to the channel conditions. This means they multiply by two the number of consecutive successful transmission required to switch to a higher rate. The mechanism increases the period between successive failed attempts to use a higher rate results in fewer failures and retransmissions, thus the overall throughput is improved. Despite that AARF is an efficient mechanism; it cannot be used in multicast transmission since the implementation of this protocol requires acknowledgment and retransmission, which are disabled in multicast.

Another popular protocol is Receiver-Based Auto Rate (RBAR) [11] proposed by Holland et al. It has the goal of performance optimization in wireless networks using also rate-adaptation mechanism at MAC layer. In RBAR protocol, RTS/CTS mechanism is enabled in order to get/send feedback from receiver. In fact, RTS is sent out before each transmission by the sender and it is received by the receiver who computes the SNR of the frame. After consulting a table mapping of SNR and rate, the receiver sends back the transmission rate for the sender to use in the next transmissions in CTS. RTS and CTS headers have been modified for the purposes. This mechanism is based on SNR (computed with a priory channel model), which is a physical parameter that does not always correlate well with human perception. Moreover, RTS/CTS mechanism is unusable in multicast transmission.

B. Rate adaptation in wireless multicast

Based on similar idea of using RTS/CTS in RBAR, Basalamah et al. have proposed Rate Adaptive Multicast protocol (RAM) [12] for channel estimation and rate selection. In this protocol, multicast receivers make use of RTS to measure channel condition and send back transmission rate for sender to use in CTS. In case that a member does not receive the data frame correctly, it will send a NACK (Not Acknowledge). For enhancing the throughput, the authors added a frame sequence field to RTS. This field is used by the member to check whether multicast data frame is a new frame or retransmission. If a frame is a retransmission of a previously successfully received frame, a member will not participate in this multicast transmission. This reduces the number of retransmission. It can be noticed that the protocol makes use of RTS/CTS, NACK and retransmission, which are disable in multicast. In addition, there are many modifications to existing frames.
To overcome feedback implosion problem, Choi et al. [13] proposed Leader-based Rate Adaptive Multicasting for Wireless LANs (LM-ARF) protocol that deploys leader-based feedback approach and adapts data rate according to ARF. One of the receiving stations, which is the leader, is responsible for sending ACKs on behalf of the participating multicast stations. If any multicast receiver, which is not the leader, fails to receive a multicast frame, it will send a negative acknowledgment (NAK) to request retransmission. The AP adjusts the contention window size the same way as that of a unicast transmission thus keeping fairness between unicast flows. New frame type called CTS-to-Self frame has been added in order to guarantee the channel access and announcing the transmission of a multicast frame. This mechanism covers several aspects such as fairness, reliability, and performance; however, since it uses ARF, it also inherits the static-threshold approach and drawback of ARF as well.

To avoid using RTS/CTS, Villalón et al. [14] proposed Auto Rate Selection for Multicast (ARSM) protocol that uses multicast channel probe operation (MCPO) with multicast probe frame sent out by AP before sending multicast traffic. In this protocol, the user having the lowest SNR will be the one in charge of replying to the AP by multicast response. The AP then selects the multicast data rate based on feedback in three different ways: explicit, implicit, and no feedback. For avoiding collision, multicast users select backoff timer according to their SNR value.

Taking into account user perception, Park et al. proposed SNR-based Auto Rate for Multicast (SARM) [15]. It adapts transmission rate according to SNR of the node experiencing the worst channel condition. SNR references are obtained from a table listing required SNR for PSNR (peak signal to noise ratio) to be higher than 30 (representing good quality) for each transmission rate. By changing multicast transmission rate on the basis of SNR values reported by mobile nodes, the wireless channel is used more efficiently. To overcome the lack of feedback mechanism in multicast, the authors propose a channel probing mechanism to inform the access point of the channel quality at mobile nodes. To avoid collision when nodes transmit feedback to the access point, the author also proposed a backoff timer for each mobile node based on the received SNR. This scheme seems to have the closest objective to ours (good user-end quality), thus we decided to compare it in our results. For better comprehensibility, we summarized the described schemes in Table II below.

### Table II
**Summary of Rate Adaptation Protocols**

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Threshold</th>
<th>Metric</th>
<th>Feedback</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unicast</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARF</td>
<td>static</td>
<td>tx failure</td>
<td>ACK</td>
</tr>
<tr>
<td>AARF</td>
<td>dynamic</td>
<td>tx failure</td>
<td>ACK</td>
</tr>
<tr>
<td>RBAR</td>
<td>static</td>
<td>SNR</td>
<td>RTS/CTS</td>
</tr>
<tr>
<td>Multicast</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RAM</td>
<td>static</td>
<td>SNR</td>
<td>RTS/CTS, NACK</td>
</tr>
<tr>
<td>ARSM</td>
<td>static</td>
<td>SNR</td>
<td>Channel probing</td>
</tr>
<tr>
<td>LM-ARF</td>
<td>static</td>
<td>tx failure</td>
<td>Leader-based</td>
</tr>
<tr>
<td>SARM</td>
<td>static</td>
<td>SNR, PSNR</td>
<td>Channel probing</td>
</tr>
</tbody>
</table>

III. THE PROPOSED SCHEME: Q-DRAM

We have noticed from previous works that all proposed schemes use a static-threshold rate adaptation for multicast, and none of them has considered dynamic threshold adaptation; the parameters are number of transmission failure or SNR as shown in Table II. The problematic issue associated with static approach is the adaptability to the network condition fluctuation, which is common in wireless environment. Another point to notice is that all the schemes handle rate adaptation according to statistics from packet or frame levels, only SARM use the concept of PSNR to deal with user perception. Unfortunately, technical parameters do not reveal quality of experience as perceived by the user and it is still questionable whether PSNR has relationship with QoE [16].

In order to overcome different limitations and to adapt to environment and user perception dynamically, we propose QoE-based Dynamic Rate Adaptive Multicast (Q-DRAM). It is a novel mechanism that dynamically adjusts transmission rate according to user-end perception by mean of quality of experience. In fact, it is more appropriate to adapt rate in multimedia applications by taking into account the quality perceived at user-end rather than SNR or transmission failure. Therefore, the main idea of Q-DRAM is to use QoE feedback from mobile stations to provision network condition and then to adjust transmission rate accordingly.

#### A. Using PSQA to obtain QoE in real-time

In order to obtain QoE in real-time, we use PSQA (Pseudo-Subjective Quality Assessment) [8], which is based on statistic learning using Random Neural Network (RNN). The idea is to train the RNN to learn the mapping between QoE scores and technical parameters so that we can use a trained-RNN as a function to give QoE score in real-time. We have successfully deployed PSQA in other resource management mechanisms such as admission control [17] and network selection [18]. For more details about how PSQA works, please refer to the mentioned references.

For Q-DRAM, we have trained and validated PSQA using statistics of application frame level (I/P/B) to map with users’ perception. In other words, the following parameters are used: loss rate of the I frame, loss rate of the P frame, loss rate of the B frame, and mean loss burst size of the I frame. The last parameter is used to capture the way losses are distributed in the flow as it affects dramatically the perception of video [19].

For communications between access point and mobile nodes, we use IEEE 802.11k standard [20], which specifies many measurement requests and reports that can be useful for Q-DRAM. It can be noticed that with IEEE 802.11k measurements, our control traffic is less significant in terms of overhead as control traffic is sent much less frequently than other packet-level schemes. For example, control traffic can be sent every second in our scheme comparing to every single packet in the other packet-level schemes.

#### B. Strategies for selection of the threshold

The most important decision to make in rate adaptation is mainly on how long to wait (backoff) before changing rate.
For switching down, the decision is quite simple because we do not want to stay in bad situation so we switch rapidly to a lower rate. From the literature, there are two causes for switching down. The first one is failure due to varying network condition; this is naturally happened when network condition changes because of mobility, interference, etc. To avoid changing rate up and down all the time (ping-pong effect), the sender should wait for some consecutive failures before switching down. The second cause is due to the action that sender just took to switch to a higher rate; in this case of failure here, the sender switches immediately to the previous rate because the new rate appears to be too high. For our scheme, we decided to switch down immediately after both cases. Note that failure in our scheme occurs when QoE is less than a desired threshold (more details in subsection III-C). The ping-pong effect will not affect our scheme since we use time scale in unit of second, which is long enough to avoid it.

For switching up, we use dynamic-threshold strategy called binary exponential backoff (BEB) similar to AARF. This strategy allows us to adapt to varying network condition. With BEB, we increase the backoff exponentially when failure occurs or repeats after the successful attempt of rate increase. It means that if the QoE is less than desired (fail) right after switching up (just up); we switch down immediately and before trying to switching up again we wait longer by setting the backoff timer to be twice of the previous value. For the other case of failure (varying condition), we do not update backoff stage. Fig. 1 illustrates how BEB works in our scheme. At the beginning, the backoff timer is set to minimum value (thMIN). During multicast session, it will be reset to thMIN again after a successful attempt of rate increase. The backoff timer cannot exceed thMAX.

![Binary exponential backoff in Q-DRAM](image)

**C. Algorithm**

We describe in Fig. 2, the behavior of an access point in our scheme during multicast session. We assume PSQA running on every multicast node.

At the beginning, the access point transmits multicast traffic at its highest rate. The AP monitors its ongoing clients every monitoring interval (mi) in unit of second because this scale is more appropriate than scaling in packet when dealing with human perception. When the timer rings, AP begins by sending requests to multicast members in the order of membership precedence. This is to avoid collision of reports sending back from members. When a report is received, AP updates the minimum MOS (min) of the group accordingly. Once the last report has been received, it compares min with the desired QoE called as lower bound (lb). This lb is computed by adding a margin (mg) to a reference score (rf). If min is less than lb, then AP switches immediately to one-step lower rate until minimum rate; in case of unsatisfied QoE just after rate increase, the backoff stage is updated. If min is higher than lb, then AP increases the counter (representing the duration that AP has been waiting); if the counter reaches a threshold (thi) where i is backoff stage, then AP switches to one-step upper rate and the backoff stage is reset.

**IV. PERFORMANCE EVALUATION**

**A. Simulation Setup**

For the simulation, we used WLAN based on IEEE 802.11b operated in an infrastructure mode, that is all traffic passes through an access point. The video sequence is an H.264-encoded sequence of duration 60 seconds. It is encoded at 384 Kbps and streamed in multicast mode using UDP. Fig. 3 illustrates our topology in indoor environment; there is one video server on the Internet with three multicast nodes connected to it via an access point. At the beginning, all nodes are located near by the access point (less than 50m radius). At 10s, station1 (st1) moves away from the access point (150m), and then at 40s it begins to move back to its initial position. After extensive simulations, we decided to used mi=1s, mg=1, rf=3 (according to ITU), lb=4, thMIN=1s, and thMAX=4s. Hence, backoff timers corresponding to each backoff stage are {0:1; 1:2, 2:4}. We set thMAX to 4s in order to react rapidly to condition change.

![Topo of our scenario](image)
implementation flaws of the original version with wireless update patch from Fiore [22]. The patch includes realistic channel propagation, Ricean propagation model, 802.11 bug fixes, multiple data transmission rates support. We implement video streaming application by adding a video packet transmission module in NS-2. We adapt data rate in MAC level in real-time according to PSQA score using our algorithm.

B. Results

We demonstrate the results in terms of goodput (for network utilization) and QoE (for user perception). We compare our scheme to 1Mbps static-rate multicast (conservative approach), 11Mbps static-rate (throughput maximizing approach), and SARM-like mechanism (user-perception approach).

1) Goodput: We first illustrate in Fig.4 the average goodput of all stations obtained from each scheme. Then, we detail to see how an individual station behaves. For that, we present two more graphs concerning a fixed station (st0) located near by the AP in Fig. 6 and a mobile station (moving away from and back to the AP) in Fig. 7. Note that we normalized the goodput according to the encoding rate of the video thus we obtained the results in scale [0:1].

It can be seen from Fig.4 that our scheme provides highest average goodput. More importantly, our average goodput is significantly higher than all others schemes during node movement (10s to 50s); however, we also observed the fluctuation generated by attempts of rate increasing during this period. Furthermore, it can be noticed that the goodput is lowest when transmitting at 1Mbps; this proves the problem of bandwidth wasting in multicast. On the contrary, using maximum rate (11Mbps) gives high goodput at the beginning and at the end; but when the distance increases because of mobility, channel condition degrades (high BER) and this strategy performs badly. For SARM, it performs slightly better than basic rate in general; even so, there is no improvement during mobility.

The results obtained in Fig.4 are confirmed in Fig. 5, in which we present selected rates of Q-DRAM and SARM during the simulation. We can notice that Q-DRAM uses high transmission rates when possible; which results in better goodput comparing to SARM. During mobility (low SNR), our scheme attempts to increase rate as soon as it detects good channel condition, hence the consequence in several rate switching. But again, Q-DRAM still outperforms the other schemes during this mobility period.

For a fixed station situated near by the AP (Fig.6), the goodput is excellent when using 11Mbps and Q-DRAM. This is because the station, close to the AP, can profit fully from short distance (high SNR), which enables the use of reliably high transmission rate. On the contrary, using maximum rate (11Mbps) gives high goodput at the beginning and at the end; but when the distance increases because of mobility, channel...
performance as the rate is too high; this generates high BER and high Frame Error Rate (FER) seen in Fig.8. However, we noticed that even when using lowest rate as in SARM and 1Mbps, the goodput also stays in bad situation.

Note that we present only the goodput of multicast stations. If we consider also background traffic, its goodput will be increased when the rate increases and we gain more goodput as much as we stay at higher rates. This is because sending at faster rate allows more timeslots for other traffics.

2) Quality of Experience: We illustrate two QoE graphs concerning minimum QoE in time and average QoE of all stations in Fig.9 and Fig.10 respectively.

Fig.9 presents the scores obtained by a member encountered the worst channel condition (lowest MOS); of which we took as reference for adjusting rate in our scheme. It can be seen that Q-DRAM performs the best regardless of some drops resulting from failed attempts during mobility. We prefer to try often in order to gain more throughputs. As seen before, 11Mbps performs the worst during mobility because of high rate. SARM and 1Mbps give similar performance as SARM has adapted to use 1Mbps during node movement; even so, this rate is not fast enough to transmit all encoded data.

Finally, Fig. 10 illustrates the overall performance of the network regarding user satisfaction by mean of average MOS of all stations. It can be noticed that since we use QoE as indicator in Q-DRAM, we obtained a great performance in QoE (the average MOS is at least 3.5 along the session). However, there are a few drops in the graph due to the failed attempts of rate increase. We also observed that the problem of SARM-like mechanism may be caused by PSNR definition that does not correlate well with QoE.

V. CONCLUSIONS AND FUTURE WORKS

We have seen that using default basic rate for multicast transmission (conservative approach), has disadvantages not only in network utilization but also in quality perception at the users. Deploying the maximum rate (11Mbps) gives great performance if network condition is good, however when the condition degrades this very high rate leads to poor performance. Many schemes, including SARM, make use of SNR as metric for changing rate but SNR does not always imply quality of experience, which is essential in real-time multimedia applications. To obtain good network utilization while maintaining user satisfaction, we have proposed a novel mechanism using QoE to trigger rate adaptation in wireless multicast. Our scheme is dynamic, it can thus adapt to varying wireless environment better than a static-threshold approach.

As a consequence, we achieved good performances in both network utilization and user perception. In the future, we plan to investigate on rate adaptation in unicast transmission and also to conduct deeper experiment on mixed scenario of unicast, multicast, multimedia and background traffic.

REFERENCES