



HAL
open science

A new design of the IEEE 802.11 MAC layer to enhance the scalability of the DMS service

Yousri Daldoul, Djamel-Eddine Meddour, Toufik Ahmed

► To cite this version:

Yousri Daldoul, Djamel-Eddine Meddour, Toufik Ahmed. A new design of the IEEE 802.11 MAC layer to enhance the scalability of the DMS service. Local Computer Networks (LCN), 2012 IEEE 37th Conference on, Oct 2012, France. pp.160–163. hal-00966581

HAL Id: hal-00966581

<https://hal.science/hal-00966581>

Submitted on 26 Mar 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A New Design of the IEEE 802.11 MAC Layer to Enhance the Scalability of the DMS Service

Yousri Daldoul^{1,2}, Djamal-Eddine Meddour¹, Toufik Ahmed²

¹France Telecom - Orange Labs, France

²LaBRI, University of Bordeaux 1, France

{yousri.daldoul, djamal.meddour}@orange.com, tad@labri.fr

Abstract- The Directed Multicast Service (DMS) is defined within the IEEE 802.11v amendment to enhance the reliability of the multicast transport over the WLAN networks by converting a multicast traffic into unicast. Therefore DMS guarantees the same unicast reliability degree to multicast traffic at the cost of the bandwidth. While DMS can be used to deliver a standard multicast stream to a limited number of receivers, it does not scale well for high throughput traffic like High Definition (HD) TV streams. In this paper we study the DMS scalability and we evaluate the transmission queue behavior using both constant and variable bit rate traffic. We show that in the case of a real time video streaming, losses caused by queue rejections may exceed the packet loss rate of the legacy multicast, limiting therefore the reliability of the DMS service. We propose a scalable version of DMS called S-DMS which executes the DMS service on the base layer and switches between DMS and the legacy multicast for the enhancement layer of a video traffic to avoid queue rejections. We show that S-DMS is able to increase considerably the number of admitted multicast members and protects the essential packets from both queuing rejections and channel losses. (*Abstract*)

Keywords- IEEE 802.11v; Directed Multicast Service; multicast to unicast conversion; reliable multicast transport; scalable DMS

I. INTRODUCTION

Nowadays, multimedia services are becoming very famous and are delivered everywhere using the new generation of smart phones and tablets in addition to the dedicated devices like TVs. To provide such services without any location constraints, the wireless technologies remain the preferred delivery way. Considering resource limitation of wireless technologies, designing a transport protocol requires at the same time to guarantee the Quality of Service (QoS) of the delivered services and to optimize the use of the bandwidth.

The IEEE 802.11 WLAN is one of the most available wireless networks. It is deployed everywhere thanks to the reduced cost of the required infrastructure and is able to provide the appropriate bandwidth for many services even those requiring high data rates like multimedia traffics. While many services require a unicast transport, other services need multicast like TV and several other real time broadcast services. Since the legacy multicast procedure does not use any feedback policy and is therefore unreliable, the use of such transport will reduce considerably the QoS of the delivered services.

The Directed Multicast Service (DMS) is one of the principle solutions trying to resolve the unreliability issue of the multicast transport. This service is defined by the IEEE 802.11v amendment [4] and operates in the MAC layer. DMS allows transmitting each multicast packet in unicast to each multicast member. Thus this ACK policy guarantees the same unicast reliability level to multicast services on the cost of the bandwidth. Therefore DMS can be used to stream a standard multicast traffic to a limited group size, but does not scale well for High Definition (HD) streams like HDTV even using the high throughput 802.11n PHY layer since the effectively available bandwidth remains limited and variable [15, 16] for three main reasons. 1) The channel bit error rate (BER). 2) The interference with other systems using the same spectrum. 3) The bandwidth is shared with other applications, cells and devices including the legacy low-throughput devices.

Even if DMS resolves efficiently the transmission failures using the unicast retransmission policy, packet losses caused by the transmission queue rejections may exist, limiting therefore the reliability of the service. Such rejections are due to queue overflow and are frequent with real time traffics having a bit rate exceeding the available bandwidth. Since most multimedia services are delivered using UDP and are time sensitive applications, packets rejected by the transmission queue are definitely lost.

In this paper we evaluate the scalability of DMS using constant and variable bit rate (CBR and VBR) traffics and we measure the transmission queue overflow. We show that the packet loss rate caused by queue rejection exceeds in many cases the packet error rate caused by the wireless channel conditions. We design a scalable version of the DMS service called S-DMS. Our service is appropriate for layered streams. In such stream, every picture contains one base layer ensuring the minimum quality and one or many enhancement layers progressively improving the video quality. In the remaining we consider that a layered stream contains one base layer and only one enhancement layer which is the superposition of all the available enhancement layers. S-DMS classifies multicast packets into two layers: base and enhancement layers, and sends each multicast packet of the base layer using DMS. Packets belonging to the enhancement layer are sent either using DMS or using the legacy multicast, depending on the available bandwidth. S-DMS is able therefore to reduce

significantly the queue rejection ratio. We show that S-DMS is able to increase considerably the number of admitted members while guaranteeing an optimized QoS for all the joining clients. We evaluate S-DMS with layered streams encoded using the H.264 Scalable Video Coding (H.264/SVC) standard [18].

The remainder of this paper is organized as follows. Section II introduces related work of some proposed multicast protocols we have studied, highlighting their problems and drawbacks. In section III we evaluate the scalability of DMS and the behavior of the transmission queue using both CBR and VBR traffics. We present the scalable DMS (S-DMS) service in section IV. We devote section V to evaluate the performance of S-DMS. Finally, we conclude in section VI.

II. RELATED WORKS

Several protocols were designed to resolve the unreliability issue of the multicast transport in the IEEE 802.11 networks [3-14]. In [3] the IEEE 802.11aa draft defines the Reliable Groupcast with Retries (GCR) service. This service provides two retry policies for the group addressed packets: GCR-Unsolicited-Retry and GCR-Block-Ack.

When the GCR-Unsolicited-Retry policy is used, the multicast source defines a retry limit, say “ N ”, and transmits each multicast frame “ N ” times without waiting for any feedback after each transmission. The retransmission of the same frame several times allows the sender to increase the probability of the correct reception at the multicast receivers. However the use of this policy does not guarantee the QoS and generates a significant overhead particularly when the value of “ N ” is high.

The GCR-Block-Ack feedback policy guarantees the same unicast reliability to the multicast traffic. Within this policy the sender transmits a block of frames and asks each multicast member to acknowledge individually using an individual Block Ack Request (BAR) frame. A member is allowed to send a Block Ack (BACK) only when it receives a BAR. The GCR-Block-Ack feedback procedure is illustrated in Fig. 1.

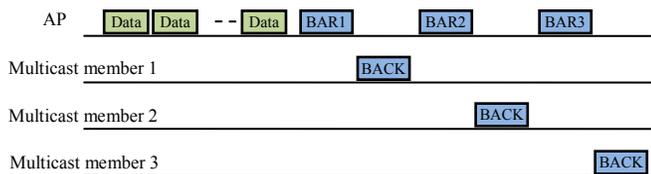


Figure 1. Typical frame exchange scenario with GCR-Block-Ack policy

When the Block Ack is used with the unicast transfer, the originator may send packets using the “Normal Ack” policy if there are no enough buffered packets to be transmitted using the Block Ack facility. However the 802.11aa proposal does not define a simplified reliable policy to deliver few packets. Thus, to take advantage of the GCR-Block-Ack reliability, a Block Ack agreement should be established for all the multicast session duration and should be used even for transmitting a single packet. This case incurs an important overhead and reduces the efficiency of the feedback policy [5].

The implementation of the GCR service within a legacy STA requires some modifications in the MAC layer. However some hardware modifications are required to support The GCR-Block-Ack policy since the BACK is managed by the radio card in order to respect the timing constraints. The GCR-Unsolicited-Retry policy does not have such constraints and requires software modifications only to be available.

In [6], authors design the DirCast system to enhance the QoS of multicast services over the IEEE 802.11 networks. DirCast selects one member called “target client” for each multicast group, then sends multicast packets in unicast to the selected client. The other clients receive the packets by monitoring the channel in the promiscuous mode. Since transmission failures may occur while listening in the promiscuous mode, DirCast sends extra FEC packets. These FEC packets allow the non-target members to recover any packet they missed. However, the transmission of the additional packets will increase the delivery delays and may incur congestion losses. Furthermore only promiscuous-enabled clients may participate in multicast sessions. Hence, users should switch manually to this mode. Moreover, the use of the promiscuous mode increases the processing load on the STA’s kernel and reduces therefore the system performance.

SoftCast, a cross-layer design for mobile video, is presented in [8]. SoftCast compresses the video pixels to bit sequences like a video codec would do, then maps these bits to complex samples in order to protect them from channel errors and packet losses, like a PHY layer would do. However, the SoftCast encoder uses only linear real codes for both compression and channel coding in order to ensure that the final coded samples are linearly related to the original pixels. Therefore, increasing the channel noise progressively perturbs the transmitted bits in proportion to their importance to the original video; a high-quality channel perturbs only the least significant bits while bad channel conditions still preserve the most significant bits. Thus, each receiver decodes the received signal into a video whose quality is proportional to the quality of its specific instantaneous channel.

The design of SoftCast relies on the following 3 assumptions. 1) The bit errors are caused only by the channel noise. 2) All received packets are processed and forwarded to the upper layer even if they contain errors. 3) Neither feedback nor bit rate adaptation are used. However the first assumption is not compliant with a typical 802.11 network where errors and losses are frequently caused by interference and packets collisions. In this case, the received bits will be correlated with the collision source content and not with the original video pixel. In SoftCast all received packets are processed even if they contain errors. However an error may occur in the MAC header preventing the MAC layer from forwarding the packet to the appropriate application. Moreover, SoftCast can not use any encryption algorithm to protection the data because this will remove the wanted linearity between the transmitted signal and the original video pixel.

D-Cast, a similar proposal to SoftCast, is presented in [9] with the main aim of enhancing the compression performance of the SoftCast video encoder. However, D-Cast still has the aforementioned weakness of SoftCast.

Medusa [10] is designed to allow packets retransmission based on the encapsulated content priority. Yet the retry policy is implemented and managed by a third party server called *the medusa proxy*. This proxy is installed between the video server and the AP. The proposed retry policy operates based on “reception reports” periodically generated by each client and emitted to the proxy. Such design incurs important delays for packet loss recovery since the proxy should wait the next report of each client to decide whether or not a particular packet should be retransmitted.

The proposal of [11] combines the use of DMS with the legacy multicast to carry a video stream to a multicast group based on the frame priority. In this proposal, the Intra coded frames (I frames) are always transmitted using DMS while the remaining frames are always delivered using multicast. Yet, I frames have the most important size among the video pictures, generating therefore the highest instantaneous bit rate. Hence the proposal of [11] can not avoid the queue rejections better than DMS and still has a limited scalability. The main difference between S-DMS and [11] is three-fold. 1) S-DMS is designed for a layered stream where the bit rate of the base layer is always lower than the stream bit rate. Hence S-DMS is able to protect base layer packets against queue rejections for important multicast group sizes. 2) S-DMS switches between DMS and multicast to transmit the enhancement layer packets while the proposal of [11] sends lower priority packets always using multicast. Thus S-DMS is more reliable for carrying enhancement data. 3) S-DMS achieves selective rejection to protect base layer packets from queue overflow. Such a mechanism is not defined within [11].

In spite of its limited scalability, DMS remains among the simplest reliable multicast protocols to implement. All required changes for a legacy AP to support DMS are software updates and legacy STAs can take advantage of this service even without supporting all the 802.11v specifications. However, no work has studied the scalability of this service. In the next section we try to fill this gap and we show the impact of queue overload on the reliability of DMS.

III. OVERVIEW OF THE DIRECTED MULTICAST SERVICE SCALABILITY AND ITS IMPACT ON THE TRANSMISSION QUEUE

In basic UDP connections, the application uses the default socket configuration¹ which blocks the transmission when the transmission queue is full and resumes when the queue is available again. This configuration allows the application to avoid losses caused by queue overflow. However real time applications like live555 [20] are more sensitive to delays and configure therefore the socket in order to reject packets

coming to date once the queue is overloaded. In the remaining we consider that all packets forwarded to the MAC layer once the queue is full are rejected.

We evaluate the queue overload using DMS with a CBR and a VBR flows. We propose a simple analytical model to estimate the average dequeue rate of the transmission queue using a CBR stream. We define X as the number of transmission attempts. The probability for a given member M_i , $i=1..K$, to receive correctly a packet in any of the N first transmissions is given in Equation (1):

$$P^i(X \leq N) = 1 - (1 - Ps(i))^N \quad (1)$$

where $Ps(i)$ is the packet success rate for M_i . Therefore, $Ps(i)$ is a function of the data length and the bit error rate. The probability to receive a packet correctly in $X=0$ attempts, $P^i(X=0) = 0$. For simplicity we consider that all packets have the same length. We express the MAC layer efficiency as perceived by a given member M_i in Equation (2):

$$E^i = (\text{data_length} / \text{PHY_rate}) / \text{transmission_time} \quad (2)$$

We obtain the average packet transmission time in Equation (3) for a retry limit of 7:

$$T_{packet}^i = \sum_{N=1}^7 (BT(N) + T_{transmission_overhead} + T_{MPDU}) \cdot (1 - P^i(X \leq N - 1)) \quad (3)$$

where $BT(N)$ is the average backoff time of the N^{th} transmission,

$$T_{transmission_overhead} = T_{PHY_header} + SIFS + T_{ACK} \quad (4)$$

and:

$$T_{MPDU} = (MPDU_overhead + Data_Length) / Rate \quad (5)$$

Hence, we derive in Equation (6), the efficiency equation as perceived by M_i :

$$E_i = \frac{\text{Data_Length} / r_i}{T_{packet}^i} \quad (6)$$

where r_i is the used rate to send packets to member M_i . To estimate the average dequeue rate we compute the average transmission rate according to Equation (7). We consider the case of CBR stream delivered to K members using DMS.

$$r_{Tx} = \frac{K}{\sum_{i=1}^K \frac{1}{r_i E_i}} \quad (7)$$

For simplicity we consider that all members are served with the same data rate and have the same transmission efficiency. Hence we get the average transmission rate in Equation (8):

$$r_{Tx} = r \cdot E \quad (8)$$

Finally we derive the average dequeue rate in Equation (9):

$$r_d = \frac{r_{Tx}}{\text{data_length}} = \frac{r \cdot E}{\text{data_length}} \quad (9)$$

We depict the analytical (Theo) dequeue rate and the dequeue rate measured using ns3 [19] in Fig. 2. We obtained these results using three different CBR streams of 12, 15 and 18

¹The default configuration of UDP sockets blocks sending until queue space is available by disabling `O_NONBLOCK` under Linux and `FIONBIO` under Windows. Live555 sets `O_NONBLOCK/FIONBIO`. Thus if space is not available, the packet is definitely lost.

packets per frame (ppf) and 25 frames per second (fps). Each packet has a MAC data length of 1500 Bytes. We fixed the maximum queue size to 150 packets² and used a transmission rate of 54Mbps to serve all the 6 members. We configured the wireless channel to have a packet loss rate of 5%.

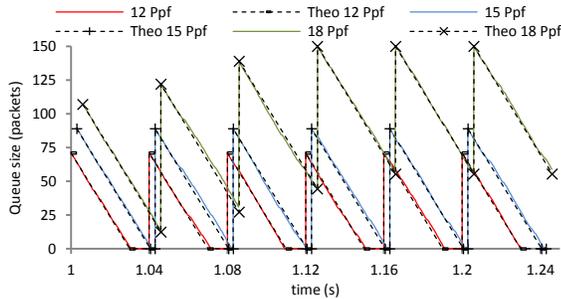


Figure 2. Queue behavior using DMS and a CBR stream

Fig. 2 shows that the AP may serve up to 15ppf to 6 members without causing any queue rejection. However, a flow of 18ppf exceeds the dequeuing rate and causes several rejections starting from the fourth frame in this example.

We compute in Fig. 3 the allowed number of ppf for a CBR video stream of 25 fps for different multicast group sizes. We consider that all members are served using the same data rate. The results are obtained using our analytical model. We assumed that only one stream is served and no other traffic is sharing the channel. Therefore the obtained results show the maximum allowed ppf and may decrease considerably according to the medium load and the channel conditions.

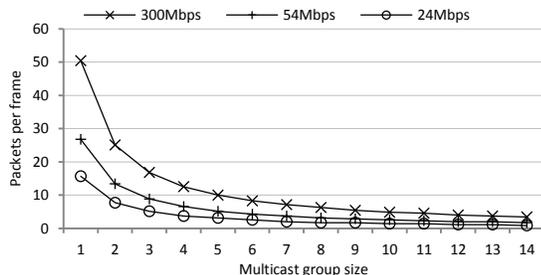


Figure 3. The maximum allowed stream rate using DMS

In the remaining of this section we evaluate DMS using VBR video streams. We considered 3 uncompressed sequences from [21]. Then we encoded them into 5 streams using JSVM9.16 [22]. All the videos have 25fps and are summarized in Table 1.

Table 1. Videos characteristics

Video name	Resolution	Video bit rate	Base I. bit rate	Enhanc. I. bit rate
CITY_4CIF	704 × 576	8.69Mbps	554kbps	8.15Mbps
CITY_CIF	352 × 288	1.35Mbps	147kbps	1.2Mbps
CITY_QCIF	176 × 144	375kbps	40kbps	335kbps
FOOT_CIF	352 × 288	3Mbps	570kbps	2.45Mbps
FORE_CIF	352 × 288	1.47Mbps	181kbps	1.3Mbps

²The maximum queue size in the MadWifi [23] and ath9k [24] drivers and in the ns3 simulator are 50, 123 and 400 packets respectively by default. We reduced the maximum queue size of ns3 to reduce the queuing delays. We set the size limit to 150 packets.

In Fig. 4 and Fig. 5 we evaluated the scalability of DMS using a PHY data rate of 54Mbps to serve all the group members. The wireless channel generates an average packet loss of 5%.

Fig. 4 shows that DMS is able to serve the 4CIF resolution of “CITY” stream to only one member without causing any queue rejection. Once the group size exceeds one member, the packet loss ratio caused by queue rejection exceeds the one caused by the wireless channel. Therefore, the legacy multicast becomes more reliable than DMS. For the CIF and the QCIF resolutions, the scalability of DMS is limited to 3 and 13 members respectively in case of a non shared medium. We see that the scalability of DMS is very limited. This scalability will be considerably decreased when the medium is shared with other STAs and other streams.

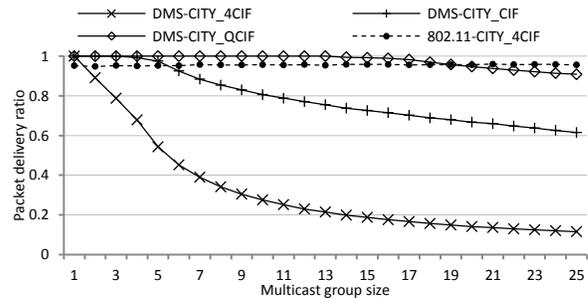


Figure 4. DMS scalability using VBR with different resolutions

Fig. 5 evaluates the scalability of DMS using 3 different video contents of the same resolution. We show that DMS is able to stream the CIF resolution to 3 or 4 members, depending on the video bit rate, without causing any queue rejection. Therefore, DMS can not be deployed to serve 2 or more multicast sessions to more than 3 receivers in each group. We show that the packet delivery ratio of the legacy multicast is higher than DMS once the group size exceeds 6 members.

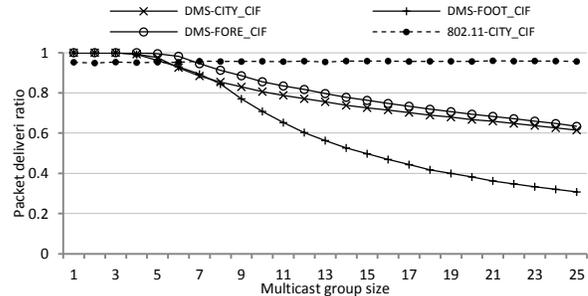


Figure 5. DMS scalability using different video contents

The reduced scalability of DMS allows a private usage of this service to provide a reliable multicast transport for 2 or 3 family members or friends. However, DMS can not be used in a public environment like in an airport or in a coffee to stream a TV channel or a live football match to many receivers while providing Internet connection to other STAs.

IV. THE SCALABLE DIRECTED MULTICAST SERVICE

Like DMS, S-DMS operates in the MAC layer. The basic concept of S-DMS relies on 3 main operations. 1) Upper layer

packets are classified into one base layer and one enhancement layer. 2) S-DMS transmits packets belonging to the base layer using DMS and packets belonging to the enhancement layer using either DMS or the legacy multicast, depending on the available bandwidth. 3) If a packet of the base layer requires being enqueued when the transmission queue is full, S-DMS erases the last enqueued packet of the enhancement layer to provide space for the incoming packet.

The first requirement to run S-DMS is to provide information on the layer to which the enqueued packets belong. This information is available at the application level and may be forwarded to the MAC layer using several ways. The first way is to stamp packets by adding the required information into the socket structure which carries a packet from the application level to the MAC layer. The second way is to add the needed information into an additional header and encapsulate it within the MAC layer. This header will be processed and removed in the MAC layer. Another possible way is to parse the enqueued packets in the MAC layer. However the use of this way requires that the MAC layer should be aware about the upper layer protocols and the encapsulations headers.

In this paper we do not define a special way to stamp a packet, but we consider that such a way is already implemented. Furthermore, stamping a packet is necessary in the sender side only. Once the packet is transmitted, the destination MAC layer does not need to know any special information about the packet and should process it like an ordinary packet.

To accomplish the two other functionalities of S-DMS, we defined a new queuing concept based on the use of the main transmission queue together with one virtual queue called the Virtual Enhancement layer Queue (VEQ) and one queue reserved to save unicast packets converted from one multicast packet. The latter queue is called *DMS Queue*.

The new design of the MAC queue is illustrated in Fig. 6.

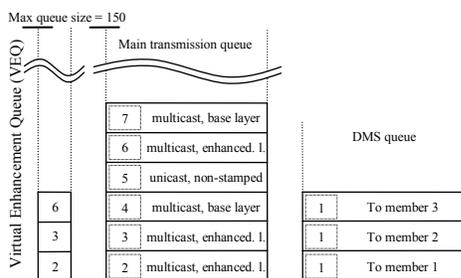


Figure 6. Queuing management entities

The second requirement of the S-DMS service is to decide whether to use DMS or the legacy multicast to send a packet of the enhancement layer. Packets belonging to the base layer are always sent using DMS. Since estimating the available bandwidth is not an easy task, the best way to evaluate the average load of the WLAN is to measure the queuing delay [17]. Therefore we define a queuing delay limit called *DelayToUseDMS*. All unicast packets and non-stamped multicast packets are dequeued and transmitted without any

additional processing. Concerning stamped multicast packets, at the dequeuing time, if the packet belongs to the base layer or if the queuing delay of this packet does not exceed *DelayToUseDMS*, then the packet is converted to unicast and saved into the *DMS Queue*. Afterwards packets in this intermediate queue are transmitted till the queue is empty. The entire dequeuing process is illustrated in Fig. 7.

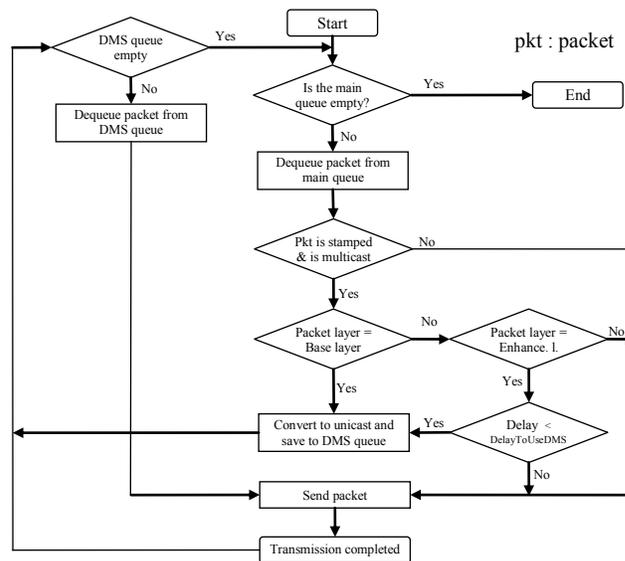


Figure 7. Flowchart of the dequeuing process using S-DMS

The value of *DelayToUseDMS* has no impact on the queue rejection ratio of the base layer packets since they are always transmitted using DMS. However a relatively high delay value may be useless if the enhancement layer packets are rejected before switching to multicast. Furthermore, if the delay value is relatively very low, the enhancement layer packets may be always transmitted using the legacy multicast. We believe that an appropriate value of *DelayToUseDMS* should be about one time the packetizing rate of the video stream. Such a value allows the MAC layer to accelerate the transmission rate in order to make space for new packets arriving at the next packetizing interval. Thus for a video stream of 25fps, *DelayToUseDMS* may be set to 40ms.

The third operation of S-DMS relies on the use of the VEQ. This queue does not store real packets, but saves only references of the enhanced layer packets stored in the main queue. Thus when the main queue is full and one packet should be erased to make space to the incoming base layer packet, VEQ is used to provide a speed way to erase the appropriate packet from the main queue.

When a unicast packet, a non-stamped multicast packet or an enhancement layer multicast packet arrives at the MAC layer and the main queue is full, the arriving packet is automatically rejected since we consider that the arriving packet has the same priority than any other enhancement layer packet already stored. The entire enqueueing process is illustrated in Fig. 8.

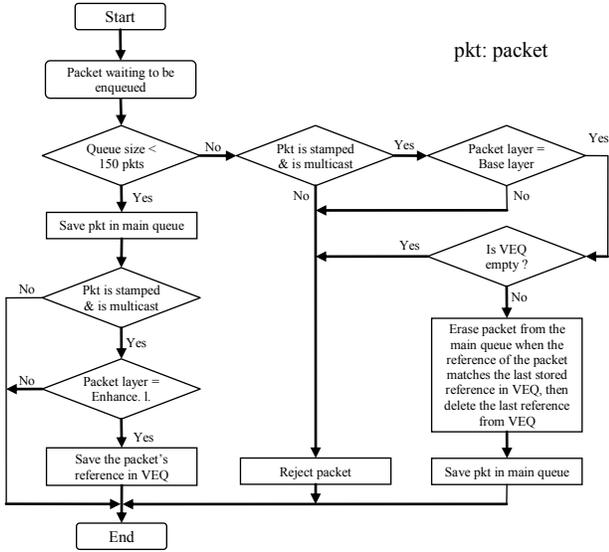


Figure 8. Flowchart of the enqueueing process using S-DMS

The implementation of S-DMS requires only some software updates in the multicast source and does not need any hardware change. The main challenge of S-DMS is to avoid any modification or additional configuration at the receiver equipment in order to be compliant with legacy 802.11 STAs.

In S-DMS we do not define an admission policy for joining members, consequently any client willing to participate in the streaming session is accepted. We leave the decision to join or to leave a session to the only intension of the customer who may decide based on the received video quality. Moreover, this paper does not define a particular procedure to manage group membership. Like in 802.11aa, we consider that such procedure is already defined and the members' addresses of each group are managed and stored in the MAC layer.

V. SIMULATION RESULTS

In this section we evaluate the S-DMS overall performance using the ns3 simulator. In Fig. 9 we measure the queue load during the video packetizing intervals and we present the maximum queue size recorded after the enqueueing of one video frame and the minimum size measured just before the packetizing of the next frame. The gap between the maximum size and the minimum size represents the number of served packets. We limit our measurements on 25 packetizing intervals. These results are obtained using 4 different CBR streams of 15, 25, 45 and 55 ppf. All streams have 25 fps where 20% of the packets of every frame belong to the base layer and 80% to the enhancement layer. Each packet has a MAC data length of 1500 Bytes. We used a transmission rate of 54Mbps to serve all the 6 members. We consider that the channel has a packet loss rate of 5%. In the remaining of this section we set the value of *DelayToUseDMS* to 40ms.

In Fig. 9(a) all frame packets are transmitted and the queue is emptied before packetizing the next frame. Thus all packets are delivered using DMS. In Fig. 9(b) and 9(c) however, the

available bandwidth is not enough to send all the enqueued packets using DMS. Therefore S-DMS switches to multicast to accelerate the dequeuing rate in order to avoid the queue rejections. In Fig. 9(d) the packet incoming rate exceeds the dequeuing rate even using multicast. Thus S-DMS performs a selective rejection to protect the base layer packets.

In Fig. 9, we notice that the length of the first segment of each sub figure has the smallest size among other segments. This is explained by the fact that during the transmission of the first video frame, none of the stored packets has reached *DelayToUseDMS*. Thus only DMS was used. Starting from the second packetizing time, the queuing delay of several packets exceeded the predefined delay limit. Therefore multicast is used together with DMS, allowing the delivery of more packets. Hence the segments size increase.

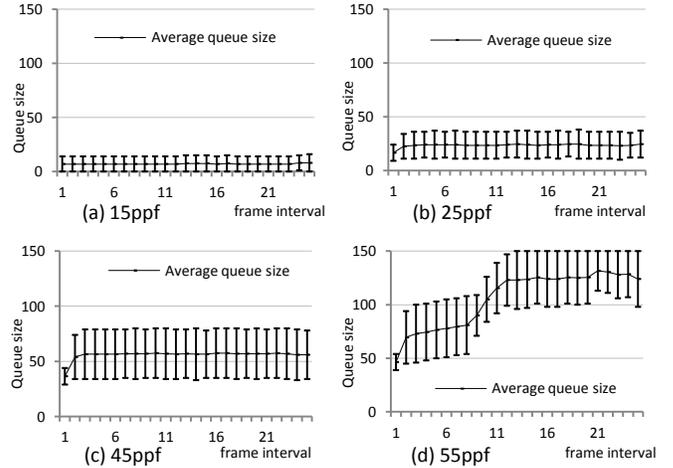


Figure 9. Queue load using S-DMS and a CBR stream

In the remaining, we evaluate S-DMS with VBR streams using the SVC videos illustrated in Table 1. We depict in Fig. 10 the queuing rejection ratio for 4 different videos.

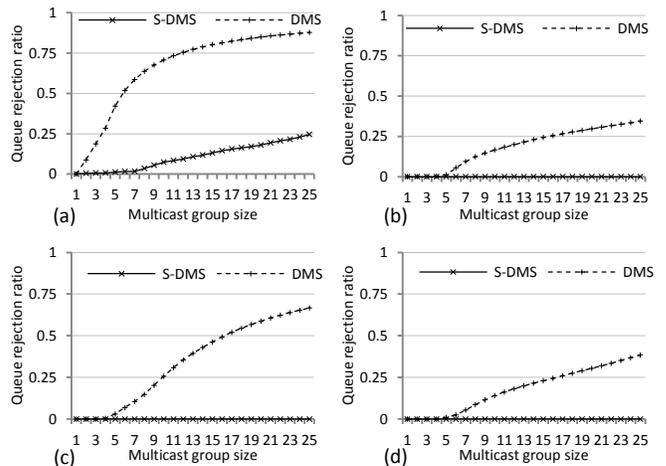


Figure 10. Comparing the Queue rejection ratio of S-DMS with that of DMS using 4 different video streams: (a) CITY_4CIF, (b) CITY_CIF, (c) FOOT_CIF and (d) FORE_CIF

Fig. 10(a) shows that S-DMS reduces significantly the number of rejected packets compared to DMS for a high bit rate video. However, Fig. 10(b), (c) and (d) show that S-DMS avoids queue rejections for all the considered group sizes.

In Fig. 11 and 12, we present the frame delivery ratio based on the successfully constructed frames at the receivers' side. A frame is successfully constructed if all its packets are correctly received. We consider two different videos. We notice that S-DMS is able to deliver reliably the base layer frames to at least 25 members while the scalability of DMS is limited to 4 members in Fig. 11 and to 10 members in Fig. 12. Moreover we observe that there is lesser decrease in frame delivery rate among the enhancement layer using S-DMS compared to DMS. Fig. 11 shows that the legacy 802.11 multicast outperforms S-DMS when delivering enhancement layer frames to more than 17 members. This is explained by the fact that S-DMS performs several queue rejections to protect the base quality as depicted in Fig. 10(a).

We notice based on Fig. 11 and 12, that using 802.11, the base layer delivery rate exceeds significantly the enhancement layer delivery rate despite the packet error rate is the same. This is justified by the fact that the base layer size in each video picture is lower than the enhancement layer size, and the probability to receive correctly a particular video frame decreases with the increasing number of the packets encapsulating the frame.

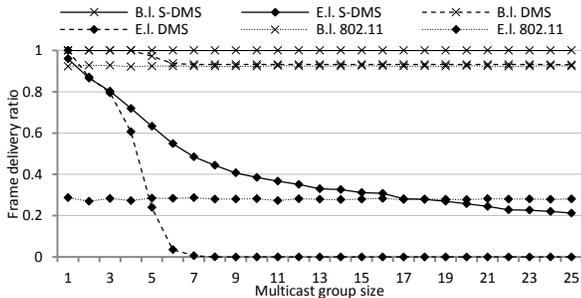


Figure 11. Frame delivery ratio using CITY_4CIF

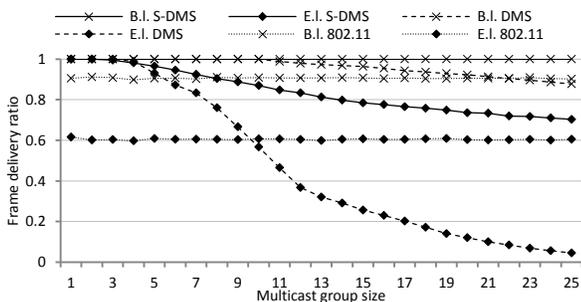


Figure 12. Frame delivery ratio using FOOT_CIF

In the remaining of this section we changed our simulation scenario and we consider the case of a good channel condition where the packet losses are caused only by transmission collisions. We set up an infrastructure WLAN with one AP and 6 associated STAs. The AP streams one multicast video to

a multicast group of 6 members. In this scenario we consider that the channel is shared between all the nodes and every node generates single unicast traffic of a constant bit rate. We increase this bit rate progressively to evaluate the performance of S-DMS and we compare it with DMS and the legacy 802.11 multicast procedure.

In Fig. 13 the AP streams CITY_4CIF in addition to one unicast flow. We see that S-DMS delivers reliably the base layer to all members when the generated unicast traffic rate is up to 1.9Mbps per node. In this scenario we show that DMS is useless and does not guarantee any QoS. Moreover, almost all the enhancement layer frames are lost using DMS.

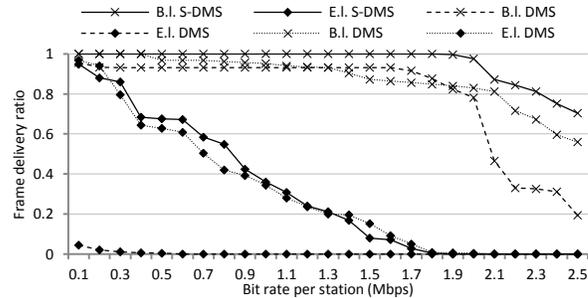


Figure 13. Frame delivery ratio for 6 members using CITY_4CIF

The comparison of the 3 protocols using a lower bit rate stream is illustrated in Fig. 14. We notice that S-DMS outperforms significantly DMS. S-DMS also outperforms 802.11 since our protocol is able to avoid the queue overflow and delivers several enhancement layer packets reliably using unicast once the bandwidth is available. The protection feature of the base layer allows S-DMS to preserve the base quality against both queue rejections and channel losses.

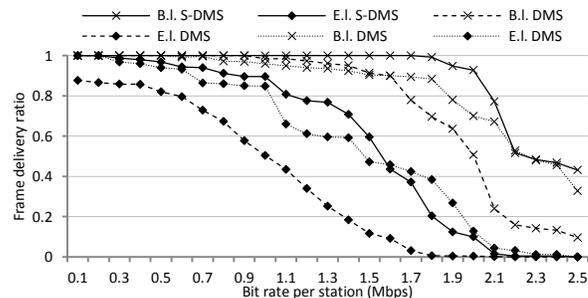


Figure 14. Frame delivery ratio for 6 members using FOOT_CIF

VI. CONCLUSION

In this paper we studied the DMS proposal to guarantee the same unicast reliability level to the multicast transport and we show that DMS has a very limited scalability even using high PHY data rates. We demonstrate that the major weakness of DMS is the uncontrolled transmission overflow. Therefore we introduced a scalable version of the DMS service called S-DMS which behaves better against queue rejections. This service is designed for layered streams like those coded using SVC. S-DMS classifies a multicast video into one base layer

and one enhancement layer, then transmits the base layer using DMS and switches between DMS and multicast to deliver the enhancement layer. S-DMS is able to perform a selective reject in order to protect the most important packets from queue rejection. We show throw simulation that our proposal can deliver reliably the base layer to an important group size and outperforms therefore DMS. S-DMS is also able to avoid sharp decrease of the provided QoS.

REFERENCES

- [1] "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", IEEE std 802.11, 2007.
- [2] "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Enhancements for Higher Throughput," IEEE std 802.11n, 2009.
- [3] "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: MAC Enhancements for Robust Audio Video Streaming," IEEE P802.11aa/D8.01, January 2012.
- [4] "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: IEEE 802.11 Wireless Network Management," IEEE std 802.11v, February 2011.
- [5] Y. Daldoul, T. Ahmed, D. E. Meddour, "IEEE 802.11n aggregation performance study for the multicast," in IFIP Wireless Days (WD) 2011.
- [6] R. Chandra, S. Karanth, T. Moscibroda, V. Navda, J. Padhye, R. Ramjee, L. Ravindranath, "DirCast: A practical and efficient Wi-Fi multicast system," in ICNP 2009.
- [7] N. Choi, Y. Seok, T. Kwon, Y. Choi, "Transparent Unicast Translation to Improve Quality of Multicast over Wireless LAN," IEEE CCNC, 2010.
- [8] S. Jakubczak and D. Katabi, "A cross-layer design for scalable mobile video," in ACM MOBICOM 2011.
- [9] X. Fan, F. Wu, D. Zhao, "D-Cast: DSC based Soft Mobile Video Broadcast," in International Conference on Mobile and Ubiquitous Multimedia, MUM 2011.
- [10] S. Sen, N. K. Madabushi, S. Banerjee, "Scalable WiFi Media Delivery through Adaptive Broadcasts," in NSDI 2010.
- [11] Z. Feng, G. Wen, C. Yin, H. Liu, "Video Stream Groupcast Optimization in WLAN," in International Conference on Internet Technology and Applications, iTAP 2010.
- [12] Y. Daldoul, D. E. Meddour, T. Ahmed, "The impact of the reliability on the fairness between the unicast and the multicast in highly loaded WLANs," in IEEE LCN 2011.
- [13] S. Choi, N. Choi, Y. Seok, T. Kwon, Y. Choi, "Leader-based Rate Adaptive Multicasting for Wireless LANs," in IEEE GLOBECOM, 2007.
- [14] N. Choi, Y. Seok, T. Kwon, Y. Choi, "Multicasting multimedia streams in IEEE 802.11 networks: a focus on reliability and rate adaptation," in Wireless Networks 2011, Volume 17, Number 1.
- [15] M. Li, M. Claypool, R. Kinicki, "WBest: A bandwidth estimation tool for IEEE 802.11 wireless networks," in IEEE LCN 2008.
- [16] S. U. Rehman, T. Turetletti, W. Dabbous, "Multicast Video Streaming over WiFi Networks: Impact of Multipath Fading and Interference," in IEEE ISCC 2011.
- [17] Sangho Shin, Henning Schulzrinne, "Call Admission Control in IEEE 802.11 WLANs Using QP-CAT," in IEEE INFOCOM 2008.
- [18] H. Schwarz, D. Marpe, T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard," in IEEE Transactions on Circuits and Systems for Video Technology 2007.
- [19] NS-3, the Network Simulator, home page: <http://www.nsnam.org/>
- [20] The LiveMedia Framework, home page: <http://www.live555.com/>
- [21] YUV test video sequences, downloading FTP server: <ftp://ftp.tnt.uni-hannover.de/pub/svc/testsequences/>
- [22] "JSM Software Manual," version 9.16, 17 December 2008, cvs: garcon.ient.rwth-aachen.de/cvs/jvt
- [23] The MadWifi drive, home page: <http://madwifi-project.org/>
- [24] The ath9k driver, <http://linuxwireless.org/en/users/Drivers/ath9k/>