Multicast in 802.11 WLANs: An Experimental Study

Diego Dujovne, Thierry Turletti
Projet Planète, INRIA
2004 Route des Lucioles, BP93
Sophia Antipolis, France
diego.dujovne@sophia.inria.fr, turletti@sophia.inria.fr

ABSTRACT

While the deployment of WiFi networks continue to grow at an explosive rate, the multicast multimedia delivery service on WiFi compliant devices is still in its early stage of development. The real culprit is the IEEE 802.11 MAC protocol, and in particular, the absence of feedback mechanism when multicast is used. Recently, a leader-based mechanism has been proposed to overcome this problem. In this paper, we measure the characteristics of the legacy multicast transmission mechanism and analyze its flaws. Then, we study the performance of the leader-based approach and compare its performance with the standard multicast service. The analysis is done on a large set of measurements made with our wireless testbed. Such measurements are an important complement to previous simulation studies and help in the design of the best mechanism to replace the faulty legacy multicast mechanism. Our study confirms that the leader-based mechanism outperforms the standard open-loop multicast mechanism while keeping fairness among other traffic.

Categories and Subject Descriptors

C.2.1 [**Computer Systems Organization**]: Network Architecture and Design—*Wireless communication*

General Terms

Experimentation

Keywords

IEEE 802.11 WLAN, wireless measurement, multicast transmission, packet loss correlation.

1. INTRODUCTION

IEEE 802.11 wireless LANs (WLANs) are one of the fastest growing network technologies in the wireless communication field. Today, most of our personal digital assistants (PDAs) and laptops include by default a WiFi interface. At the same time, we have been overrun with all kinds of multimedia applications. Since more and more places are covered by hotspots, this will allow travelers at airports or at rail stations to use their PDAs and watch television broadcasts or newsflashes. Multicasting video instead of streaming individually each video flow results in a much more efficient use of the shared wireless medium. Whereas all these new applications are very likely to appear soon with upcoming WiMAX or DVB-H enabled devices, the IEEE 802.11 standard does not comply with multicast data requirements. In particular, the current MAC layer sends multicast packets in open-loop as broadcast packets, i.e., without any possible acknowledgements.

This open-loop transmission mechanism causes three main problems. First, without feedback mechanism, it is not possible to adapt the contention window according to the network state, as it is done with regular point-to-point connections. Consequently, multicast flows achieve a higher priority[1] than concurrent unicast flows and the network may become severely congested and could collapse. Second, it is not possible to adapt the physical (PHY) transmission rate to the channel characteristics, so the packets are broadcast over the wireless medium at one of the rates included in the basic rates set. Third, there is no way to retransmit lost packets at the MAC layer, so the transmission is more lossy than for unicast flows, which degrades performance of the multicast application.

One of the alternatives to improve the standard multicast approach is the leader-based multicast mechanism [8][10]. In a nutshell, this solution proposes to select one of the receivers to send acknowledgement frames back to the sender. As with regular unicast transmissions, the multicast sender can use a PHY rate selection mechanism such as ARF [6]. Furthermore, lost packets can be retransmitted as it is the case for unicast flows. Moreover, the leader-based approach provides fairness with other concurrent unicast flows because the same algorithm is used to adapt the contention window.

As we discuss in the following section, all the previous solutions proposed to improve performance on multicast over WLANs are only based on simulation models. Therefore, we believe that it is crucial to check if these assumptions are realistic with measurements made on current 802.11 devices. In this paper, we measure the packet loss correlation

[1]Note that we do not advocate here for a strict fairness between unicast and multicast transmissions, we only complain about the absence of ways to prevent multicast flows to swamp all the network resources.

MSWiM'06, October 2–6, 2006, Torremolinos, Malaga, Spain.

between multicast receiving stations. We also compare characteristics of multicast flows and unicast flows to evaluate the gain obtained with a leader-based [8, 10] controlled multicast session.

The contributions of this paper are twofold. First we analyze with measurements the impact of multicast transmissions on current IEEE 802.11 WLANs. Second, we use those measurements to help in designing the most efficient leader-based mechanism and we compare its performance with the legacy based multicast scheme.

All the experimentations shown in this paper are done in our 802.11 testbed using two tools available in the public domain. The first tool is a packet capture and pre-processing tool named Kismet [20]. The second one is a packet processing software called WisMon [28] that we developed as a general-purpose tool to analyze wireless networks. WisMon includes many important features which are currently lacking in the available measurement tools.

The rest of the paper is organized as follows. Section 2 presents the related works. Section 3 discusses the different tools that are currently available for measurements and packet log analysis and provides a short overview of the WisMon tool we have designed. Section 4 presents our experimental setup and the different scenarios of experimentation. Then, Section 5 describes all the measurements made and analyzes them. Finally, Section 6 concludes the paper and describes future directions.

2. RELATED WORKS

Recently, several solutions have been proposed to improve performance of multicast transmissions on WLANs. In the context of multicast video transmission over WLANs, Majumdar et al. [12] have proposed a layered encoding scheme with hybrid ARQ/FEC error correction. They have addressed theoretically the optimization of video transmission in a multiple user case. But they assumed a non realistic model based of uniform distributed packet errors. Moreover, their proposed scheme is not able to adapt the video transmission mechanism to the heterogeneity of receivers.

Choi et al. [1] have proposed to solve the unfairness problem mentioned above by dynamically adapting the contention window size according to the number of competing stations in the WLAN. Then, simulations are done and performance is evaluated using an original multicast fairness index. However, their simulations assume a perfect network with no transmission errors. In practice, the contention window size depends on both collision errors and transmission errors, so their performance results are not realistic.

Kuri et al. [8] have proposed different protocols to provide reliable multicast transmission over IEEE 802.11 WLANs. These protocols modify the MAC layer to enable the RTS/CTS option in multicast mode. They also provide solutions in order that only one receiver (called the leader) responds with a CTS or an ACK. However, these solutions are only convenient for low mobility wireless stations, i.e., when there is no need to change the leader. Gupta et al. [3] have proposed another solution to this problem that works for mobile stations for both infrastructure and ad-hoc 802.11 networks. They use dual busy tones to simulate NACKs or Negative CTS (NCTS). However, this solution is not practical because it requires two wireless network interfaces, one for transmitting and receiving the busy tones, and the other one for transmitting and receiving data packets.

More recently, Lauppe et al. [10] have proposed several MAC layer improvements for layered video transmission on 802.11 WLANs. Layered video is used to handle heterogeneity of receivers. For each video group, a leader is selected to send ACK frames. ACKs are used as in the point-to-point case to detect loss, select a convenient PHY transmission mode using the CLARA algorithm [5], and adapt the contention window size. SARC [14] is used to cluster receivers with similar characteristics and to control dynamically the sending rate and FEC level for each video layer. They have done simulations to analyze performance of the overall mechanism and assume a combined Rayleigh/packet erasure channel model. Whereas this model is more realistic than a Gaussian model, it does not take into account shadowing effects. Furthermore, to accurately evaluate the SARC clustering algorithm, a better channel model is required to take into account the effects of long term correlation. Unfortunately, to the best of our knowledge no study exist on correlated loss behavior between multiple wireless stations.

Concerning wireless traffic measurements, most of previous studies have focused primarily on fairness and channel usage for point-to-point connections or for best-effort traffic. These results can be observed in the following references.

Gopal et al. [2] have studied the behavior of simultaneous TCP flows in a wireless environment. The authors describe the differences observed between real measurements and simulations, and the appearance of unexpected situations which were not taken into account on the simulator. The authors conclude that it is indispensable to measure what is simulated, and furthermore, the need to standardize testbeds.

Ng et al. [11] have considered fairness on a 802.11e experimental platform. The study is limited to the analysis of TCP traffic under different conditions, one of them uses UDP traffic to saturate the medium [11].

Lacan et al. [9] have used real traces to build a packet loss profile as a primary resource. Then these traces were modified in order to differentiate spatial correlation errors from temporal correlation errors while keeping statistical validity. These traces have been obtained without background traffic and contain only losses due to noise and interference – packet losses caused by collisions have not been not considered.

Finally, Kotz et al. [7] discussed the common pitfalls of wireless network studies and provided some design rules to develop more accurate models. The objective of this study was to improve models based on simulations taking into account realistic propagation characteristics. Moreover they demonstrated the feasibility of the proposals with wireless measurements.

3. WIRELESS MEASUREMENT TOOLS

In this section, we provide a brief overview of measurement tools available in the public domain. We focus on open-source tools, whose functionalities and results can be verified. Then, we describe the WisMon tool we have developed in further details.

To analyze TCP streams, three main software programs are available: Tcptrace[25] is a command-line tool used to obtain statistics from TCP traces. It also includes a realtime option. Tcpflow[24] is a tool that can generate offline reports on different TCP flows. Tstat[26] can produce statistical data from network traces. These three tools are networkoriented only, i.e., they cannot be used to analyse MAC or PHY information.

For low-level packet-capture, libpcap[21] is the most widely used software, but it does not include any functionalities to process the data. The code is stable and can be used on different link layers. The last version of this tool includes 802.11 header support.

For multilevel packet analysis, ethereal[16] is the most complete tool. It provides a Graphical User Interface (GUI) for offline processing. This tool is packet-oriented and statistics can not be obtained online. Tethereal[16] is the terminal version of this software. However, it lacks very important functionalities such as the ability to automatically build histograms from MAC/PHY headers statistics. From the very long list of interpreted protocol headers and filters, the wireless prism and radiotap packet formats were included recently [18].

The wireless-specific acquisition software is dominated by *wardriving* tools, which are generally used for sniffing the 802.11 environment when searching for available Access Point (APs). Among them, Kismet[20] has the richest set of features and is the most flexible. However, it fulfills only its original purpose, which is wireless network discovery and monitoring. Although it can be used with more than one packet probe, there is no native data synchronization function available.

3.1 WisMon

Since no specific tool covered all our needs, we designed WisMon[28], a graphical software environment for data acquisition and automatic generation of per-station statistics logs.

In a nutshell, WisMon works as follows. We use a multiple coordinated probe scheme in order to capture all the traffic on a WLAN, because none of the probes is likely to receive all the transmitted packets [15]. Moreover, time is synchronized from a common reference available to all the stations[2], and the received packets are re-timestamped using this new time reference. Then, the captured data from all the probes are merged with a time-constraint criteria in order to remove duplicated packets, i.e., packets received by more than one sniffer. This list of packets is then filtered and processed through an on-line *packet classification engine*. In this manner, we can obtain near real-time histograms to analyze the behavior of WLANs. Several parameters can be obtained on a per-station basis such as received power level, inbound traffic, outbound traffic, number of retransmissions, PHY transmission mode and percentage of bandwidth used.

There are two functional modes for WisMon: offline and online. The offline mode can be used for example to study long-term patterns of some parameters, whereas the online mode is used to monitor parameters during the experiments. In the latter case, it is possible to detect anomalies in real time on a WLAN. Characteristics of current stations connected to the WLAN can be analyzed in real-time, allowing to focus the analysis of a particular station when necessary, see Figure 1.

WisMon connects as a client to a customized version of Kismet[20] to obtain the raw packet header capture.

A configuration file can be used to select and customize the sniffers to connect to, and to generate packet traces from

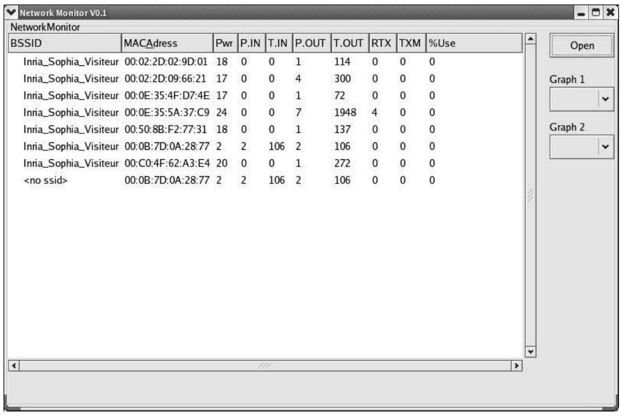

Figure 1: Screen capture of station list window from WisMon

the *packet classification engine*, as shown in Figure 2.

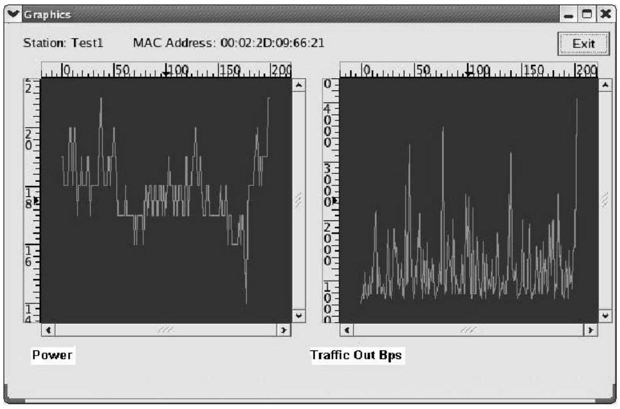

Figure 2: Screen capture of the real-time graphics from WisMon

WisMon is built using a client-server architecture, in order to separate the heavy packet processing functionality from the lightweight GUI client. The whole system is built as an open source software, which results in a very convenient and customizable tool.

4. EXPERIMENTAL SETUP

In this section, we describe our wireless testbed and the placement of wireless stations and the AP for the following experiments.

4.1 Hardware and Software

In our wireless testbed, both stations and the AP are composed of standard laptops with off-the-shelf wireless boards. This solution allows us to instrument the AP driver in order to differentiate packet lost at the sending queue from packets lost due to collisions or bad channel conditions. Furthermore, our previous experiments made with three different commercial APs (Netgear WG602, Netgear ME102 and Linksys WAP55AG) showed that these APs periodically deauthenticate stations when the network is congested. This problem biases statistics because a re-authentication requires up to 7 seconds (with no traffic exchanged), and results in

[2]The time reference is obtained using the Beacon frames which are periodically sent by the AP.

long bursts of artificial packet loss. Furthermore, we have instrumented the AP in order to provide statistics of its sending queue. In this manner, we are able to differentiate collisions from packet lost before transmission.

It is important to note that during the experiments all the stations are fixed and do not enter the sleep or power-down modes. When a station changes to power-down mode, the wireless board sends a message to the AP to start buffering the packets until it recovers full activity.

A Linux-based system has been deployed on each laptop because it brings more flexibility[3]to the WLAN boards.

As a complement to WisMon, we have developed the following toolchain. Tethereal captures and builds logs for each station. Vlc[27] is used to send video packets in multicast. These packets are then filtered, and the text output is parsed using a filter which extracts for each packet, the Prism header information and the RTP[13] sequence number. Finally, a script creates a file containing the list of lost packets, which is then post-processed to obtain the packet loss correlation values. Further details on the experimental setup follow.

Hardware:

- 6 laptops (STA1, STA2, STA3, STA4, STA5, AP). We use Dell Latitude D800 (1Gb RAM, Pentium M 1.7GHz) and COMPAQ EvoN800c (256Mb RAM, Pentium M 1.7GHz) with IEEE 802.11b Proxim Orinoco Gold wireless cards (Atheros AR5212 chipset).

Software:

- Operating System: Linux kernel version 2.6.8.1 installed in all the laptops.
- WLAN Board Driver: Madwifi[22] driver.
- Streaming generator and client: vlc[27] (VideoLan Client) media player version 0.8.1, used as a video server. RTP is selected to send video either in multicast to a group of stations or in unicast to a specific vlc client.
- TCP and UDP background traffic generator: iperf version 2.3.5

Packet Capture software:

- kismet-2004-10-R1[20] modified for time synchronization. These modifications are available as a patch in [28]
- WisMon[28] version 0.1-R4.

4.2 Physical Setup

In our wireless testbed, 5 probes (STA1-STA5) are distributed in the receiving range of the AP to collect the traffic from different places. In this way, we obtain a variety of signal-to-noise ratios (SNR) and receiving power levels. The mean receiving power values (measured from the collected data) for each station are shown in Table 1, whereas the position of each station is drawn in Figure 3.

Three stations (STA1,2,4) are located in the same office as the AP (i.e., Office B). STA1 is the nearest station from the AP. It receives a very high signal from the AP. STA2 is located in a corner at Office B and in a place where the receiving signal is lower than the one of STA1. At this position, there is a high probability to find reflected signals from nearby structures. STA4 is farther from the AP than STA2 and is placed at a very good reception spot.

STA3 and STA5 are placed in a contiguous office called Office A. STA5, located at a corner of Office A, has better reception conditions than STA3.

From Table 1, we can note the expected power values related to the distance of the stations to the AP.

To generate different levels of channel load, we use iperf to generate concurrent UDP or TCP traffic from each station.

Legacy multicast transmissions use the default transmission mode of the madwifi [22] driver which is equal to 1Mbps, whereas ARF [6] is used to select the PHY mode (from 1 to 11Mbps) for both unicast and leader-based transmissions.

[3]Note that the promiscuous mode is not always available on commercial Windows drivers. Furthermore, when this mode is available, sometimes management frames are discarded by the driver [17].

Table 1: Mean receiving power value for each station

Station	Power Value	Group
STA1	-45dBm	near
STA2	-51dBm	near
STA3	-64dBm	far
STA4	-52dBm	near
STA5	-61dBm	far

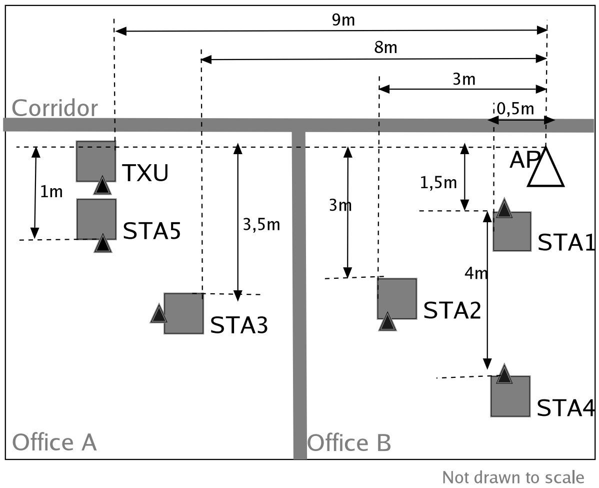

Figure 3: Distribution of the AP and wireless stations. There are two groups: near (STA1-STA2-STA4) and far (STA3-STA5)

4.3 Leader-based implementation issues

There are several ways to implement the leader-based approach. The most direct way is to modify the legacy multicast mechanism as follows. One of the receivers in the multicast group is elected to send acknowledgment (ACK) frames. The Duration/ID field of the MAC header of each multicast data frame has to be modified. In particular, the virtual carrier-sense mechanism provided by the MAC should take into account the extra delay for multicast acknowledgements. The latter modification is important to make in order to prevent possible collisions between multi-

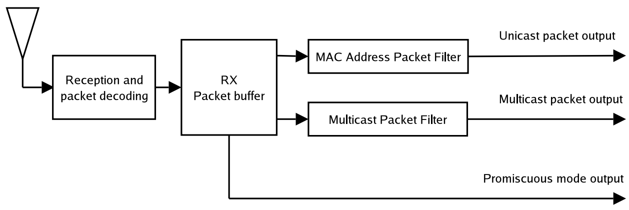

Figure 4: Receiver side of a 802.11 WLAN board

cast ACK and other frames.

However, this solution cannot be used on our current WLAN devices. Indeed, generating ACK frames requires very precise timing synchronization and is implemented within the hardware of the wireless devices to comply with this requirement. A simplified schematic of the receiver side of a 802.11 wireless card is shown on Figure 4.

The first block synthesizes the RF and decoding stages. When a packet arrives, it is directed to the packet buffer. Packets are then selected depending on the MAC address destination field on the multicast filter and the unicast filter. The multicast filter selects the packets corresponding to the multicast groups the receiver has subscribed to. The unicast filter selects the packets addressed to the card.

Fortunately, it is possible to implement the leader-based approach with current available hardware and make changes only to the software driver. For example, it is possible to fake multicast transmissions using the promiscuous mode on all receiving stations while the AP sends unicast frames to the leader-station. In this manner, all packets sent to the leader are also received by other stations. At the application level, packets must be processed to remove all the headers manually, since this method overrides the TCP/IP stack.

To minimize processing overhead, receivers can apply a filter at the kernel level to only let packets, sent by the AP to the leader (at the port corresponding to the multimedia session), reach the upper layers.

The promiscuous mode must not be passive, to allow other sources of traffic to access the medium. For example, it is possible to configure the madwifi driver to create two virtual devices: a standard driver to connect to the wireless network and a promiscuous mode driver to obtain the packets directly from the buffer.

The proposed solutions rely on the chipset capabilities and configurability, which vary depending on brand and model.

5. EXPERIMENTS AND RESULTS

In this section, we first show how unfair the IEEE 802.11 legacy multicast is with other concurrent unicast flows in current WLANs. Then, we make various experiments in order to evaluate the leader-based approach and compare its performance with the standard multicast transmission mechanism.

5.1 Why legacy multicast does not work

As discussed in the introduction, the legacy IEEE 802.11 multicast is an open-loop transmission mechanism, so, it is not possible to retransmit lost packets or select the best PHY rate mode according to the channel conditions. But the most severe problem is than contrary to unicast flows, legacy multicast flows cannot adapt their probability to access the channel according to the network load. This leads

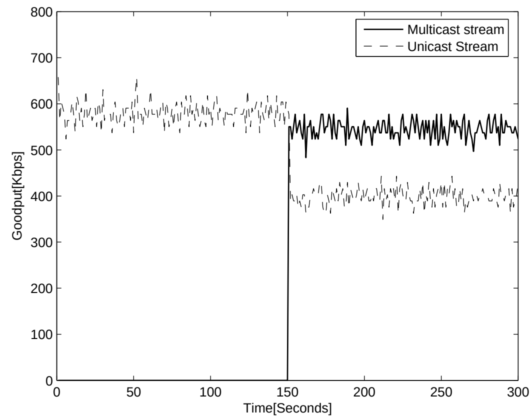

Figure 5: Goodput multicast - unicast unfairness without background traffic

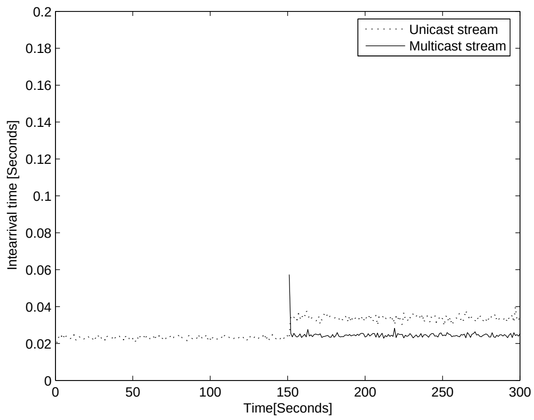

Figure 6: Interarrival time multicast - unicast unfairness without background traffic

to severe unfairness between multicast and unicast flows and can even cause network collapse.

To illustrate this problem, we have compared the characteristics of two identical UDP/CBR flows transmitted simultaneously in unicast and in multicast modes from STA5 to the AP. We use iperf to generate both flows with CBR=500kbps and using the default overall packet size of 1678 bytes[4].

Figures 5 and 6 show respectively the goodput and the interarrival time between two packets sent in the two different modes with unloaded traffic conditions, i.e., without background traffic.

At time $t = 0s$, the unicast flow is started at STA5 and achieves an average goodput of 600kbps [5]. The mean interarrival time between two packets is about $25ms$. At time $t = 150s$, the multicast flow is started on STA5. We observe that the multicast flow obtains the same performance than those observed by the unicast flow in the first $150s$. How-

[4]This includes the Prism header, (144 bytes) the 802.11 header, (30 bytes) and the CRC (4 bytes).

[5]Note also that the CBR traffic of 500kbps does not include the RTP/UDP/IP/MAC/PHY headers overhead.

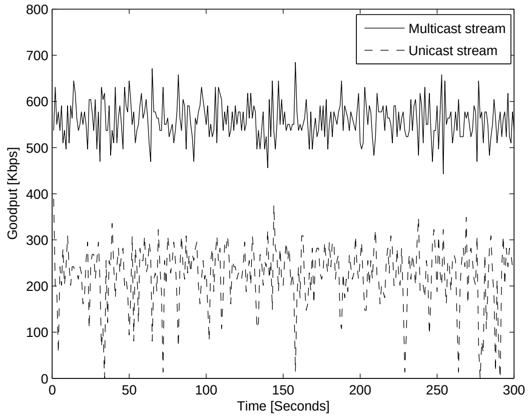

Figure 7: Goodput multicast - unicast unfairness with background traffic

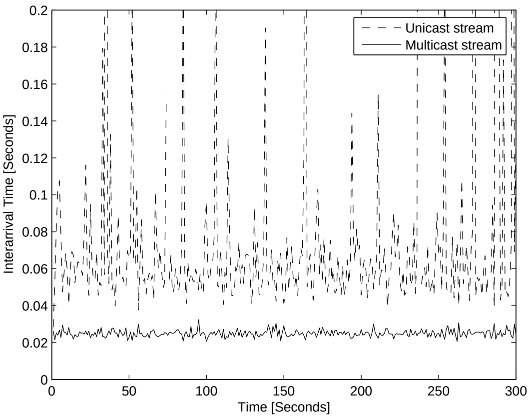

Figure 8: Interarrival time multicast - unicast unfairness with background traffic

ever, the performance of the unicast flow suddenly drops of about 30%, i.e. 30% less goodput and 30% more delay between two successive arrivals of packets.

Figure 7 and Figure 8 show the same experiment but in presence of high background traffic and when both unicast and multicast flows start at time $t = 0s$. To generate highly loaded network conditions, four uplink saturated[6] UDP unicast sources are added on each remaining station (STA1 to STA4).

As expected, we observe a higher goodput variability for both unicast and multicast flows. But this time, the difference of performance obtained between these two flows is larger than before. The average goodput for the multicast flow is about the same than in the previous experiment (i.e. without background traffic), whereas for the unicast flow, the mean goodput drops for about 70% of its original value. Indeed, the quality of the unicast stream severely decreases whereas it remains roughly the same for the multicast stream.

[6]A saturated UDP source is a packet generator station which has always a packet waiting on the queue to be sent.

5.2 Comparison between legacy multicast and the leader-based mechanisms

The remainder of the paper focus on the comparison between the 802.11 legacy multicast and the leader-based approach. We consider a video streaming application in which a vlc video source (located in the same LAN than the AP) sends a VBR video to a group of 5 receivers (STA1-STA5). In order to analyze all types of packet loss, we have run experiments for 2 different network conditions: without background traffic and in presence of congestion. Therefore, four experiments, as detailed in Table 2, have been run to generate all the figures shown in the rest of the paper.

Table 2: Table of Experiments

Exp.	Back. traffic	PHY mode	Tx method
1	No	1 to 11Mbps	Multicast
2	Yes	1 Mbps	Multicast
3	No	1 to 11Mbps	Leader-based
4	Yes	1 Mbps	Leader-based

We use a standard dvd movie to generate realistic video traffic. The video stream is encoded by vlc with mpeg2v and it is configured to send a VBR video flow with RTP/UDP encapsulation. Notice that although vlc has been configured to send a mean rate of 512kbps[7], the actual average sending rate is roughly the double. The background traffic is generated with iperf [19] as follows. For the first experiment, 10 UDP unicast flows directed to the AP are started per station, with a requested bandwidth of 10Mbps. This ensures that there is always a packet waiting to be sent in the driver queue. For the second experiment, 10 TCP unicast flows directed to the AP are started per station, which results in a lower but more realistic load than the former experiment.

5.2.1 *Packet loss correlation between stations*

In the leader-based approach, one of the receivers (called the leader) has to generate acknowledgement frames to the multicast source. The performance of such an approach highly depends on the algorithm used to select the leader and on the packet loss correlation between the stations. So, it is crucial to study the packet loss correlation between all the receivers.

When more than one station transmits at the same time, there is a high probability to observe a packet loss in all the stations (i.e., high packet loss correlation). We associate this event to a collision. When a collision is present, the packet is definitely loss when the legacy multicast mechanism is used. However, the leader-based solution should obtain better performance, because all the stations will benefit from the packet retransmission. The other uncorrelated losses can be assigned to background noise or interference, possibly the result of unadapted PHY sending rate for some of the receivers.

Figure 9 shows the packet loss profile for both the legacy multicast and the leader-based approach, with and without background traffic. This figure is composed of 4 sets of statistics corresponding to the 4 experiments detailed in Table 2. For each experiment, we plot 3 columns corresponding

[7]Note also that the mean video rate of 512kbps does not include the RTP/UDP/IP/MAC/PHY headers overhead.

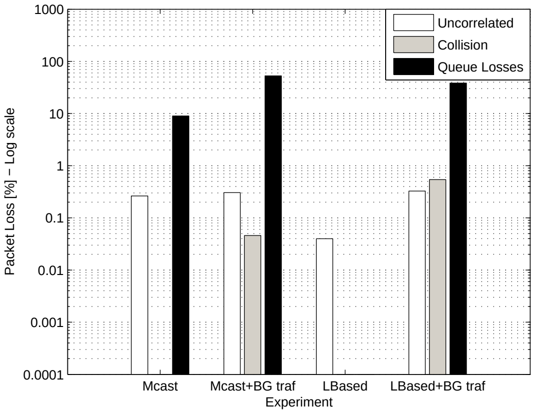

Figure 9: Multicast Packet loss correlation: Uncorrelated packet losses (independent), Collisions (same packets lost by all the stations) and Queue Losses (AP packet drops) for each of the 4 experiments.

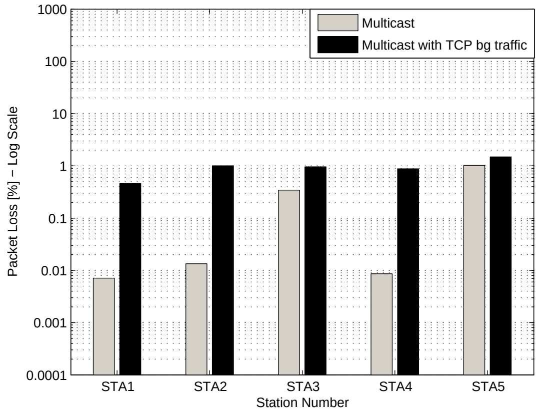

Figure 10: Per-Station Packet loss without background traffic

to the different types of packet loss: uncorrelated losses due to interference or noise, collisions, and packets lost at the AP sending queue. Five tests were done under the same conditions for each experiment. The first column comprises the mean value of uncorrelated packet losses between stations, the second column represents the mean number of collisions, and the third column shows the mean number of queue packet drops.

The first columns (in white) for these 4 experiments correspond to the uncorrelated packet loss statistics, including correlated packet losses for 2, 3 and 4 stations. Let us first consider the case when the network is unloaded. We can note that without background traffic, most of the effectively transmitted packets reach the destination: the uncorrelated packet loss is 0.03% for the leader-based approach and reaches about 0.3% for the standard multicast mechanism. This shows that most of the uncorrelated losses in the leader-based approach case are recovered by retransmissions. The leader corresponds to the worst receiver, and consequently experiences most of the losses. This can be observed on Figure 10

The second columns (in grey) of Figure 9 stand for the number of correlated losses between all the stations which are associated to collisions. There are almost no collision for all the experiments made without background traffic. However, in presence of high level of background traffic, the collisions are about 10 times more important in the leader-based approach than for the legacy multicast. The difference is due to the fact that the legacy multicast gets a higher priority than the background traffic, as it is discusses in Section 5.1.

The third columns (in black) of Figure 9 reflects the number of packets lost at the AP sending queue. In the unloaded case, the legacy multicast flow experiences a high level of packet loss (about 9%), while no packets are dropped with the leader-based solution. This difference is due to the different PHY rates used to send packets in both schemes. With legacy multicast, the PHY transmission mode, fixed at 1Mbps, is not able to support the VBR video stream. On the other hand, the leader-based solution allows to adapt the PHY sending rate to the channel conditions of the leader, which is usually the receiver that experiences the worst channel conditions. When a higher transmission rate is used, the AP queue length decreases. Because it is faster to transmit packets, no more packet loss is observed at the AP. We can observe that the ratio of packets lost at the AP sending queue is about 10% less in the leader-based approach than for the legacy multicast scheme. On both cases the main factor which influences queue growth is the transmission mode. When legacy multicast is used, the AP queue grows because the PHY transmission mode is only 1Mbps. When the leader-based solution is used, 11Mbps transmission mode with ARF is used. This allows for a shorter transmission time which leads to queue size reduction.

Figure 10 reflects the individual packet loss for each station when the network is unloaded.

Per-station packet loss is analyzed because it is an important criteria to select the leader. The station with the largest packet loss is likely to be the best candidate, since it will ask for the highest number of retransmissions. We can also observe that packet loss criteria is a better criterion to select the leader than the RSSI mean value. This is reflected from the comparison between Table 1 and the individual packet loss from Figure 10. In our experiments, STA5 obtains the highest value of packet loss. Using the RSSI value criteria, the selected station would have been STA3, instead of STA5.

5.2.2 How to select the leader

The choice of the leader has a high impact on determine the performance of the leader-based mechanism. In our experiments, we have decided to select the receiver that experiences the worst channel conditions, i.e., STA5. But for a larger group of receivers, we can imagine to select the leader differently, in order not to penalize all the other receivers. In case the set of receivers is very heterogenous, it may also be possible to cluster the receivers in groups that experience similar packet loss as it is proposed in in the SARC [14] algorithm for multicast transmission over the Internet.

To evaluate such an approach, it is important to consider spatial correlation between receivers. The spatial packet loss

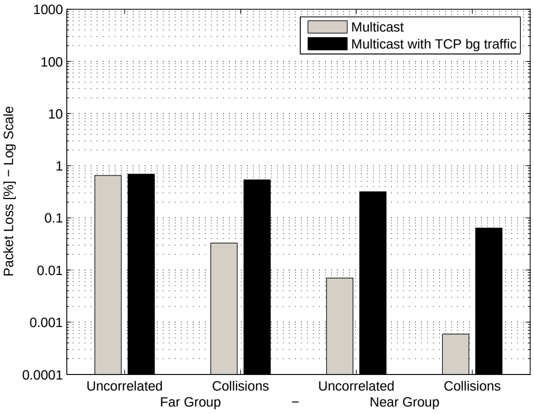

Figure 11: Multicast packet loss correlation for each group of stations

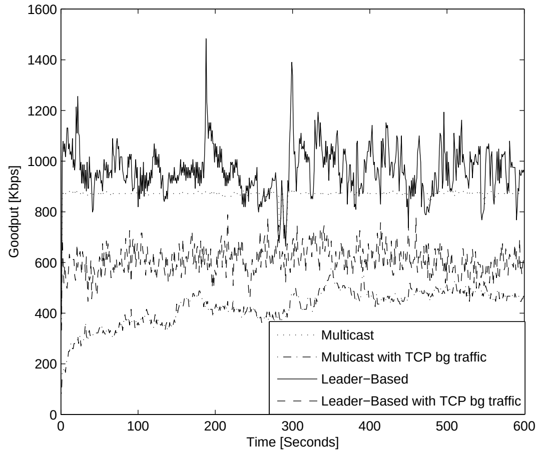

Figure 12: Video goodput for legacy multicast and leader-based mechanisms

correlation stands for the amount of packet loss experienced simultaneously by a group of receiving stations. If we observe high packet loss correlation, a leader-based solution for each multicast group could achieve good performance. Let us now also assume that an adaptive sending mechanism is implemented in order to prevent packet loss at the AP sending queue.

In our experiment, two groups of stations have been identified, as presented in Table 1. The criteria to cluster the stations are the distance and obstacles between the stations and the AP. Far group stations are located more than 6 meters from the AP and they are also behind a wall. Near group stations reside at the same room as the AP.

Figure 11 shows the correlated and uncorrelated packet loss for the two groups of stations using the legacy multicast transmission mechanism, with and without background traffic.

We observe 0.7% of uncorrelated packet losses and about 0.5% of correlated packet losses in the far group. The near group has 0.3% of uncorrelated packet loss and 0.1% of correlated packet losses. In this case, an application-level FEC mechanism can obtain better performance, as proposed by [9]. FEC mechanisms are efficient in presence of isolated losses or short bursts of packet loss. This is usually what we observe in our experimental results once we remove packets lost at the AP sending queue.

5.2.3 Performance analysis at the application level

In this section we compare the performance obtained at the application level for both mechanisms.

To study video streaming performance, the most important parameters to consider are packet loss and goodput[8].

Packet loss for the legacy multicast and the leader-based mechanisms can be observed in Figure 9 – it has already been analyzed in Section 5.2.1 but for a different purpose. Now, we study the impact of packet loss on the video receivers. When the network is unloaded, the overall packet loss for the legacy multicast is more than 9% while it is very low (about 0.04% before possible retransmission) for the leader-based approach. In presence on high level congestion, packet loss is higher than 50% for the legacy multicast and it is about 40% for the leader-based approach.

Now, let us compare the corresponding goodput performance for both mechanisms in Figure 12.

Without the presence of background traffic, we observe that the leader-based goodput reflects the VBR characteristics of the video sent by vlc. The trace follows all the fluctuations of the video traffic used for the experiment.

On the other hand, the legacy multicast mechanism, using the default PHY transmission mode, is not able to send as much throughput and gets a constant goodput of about 800kbps[9]. We can observe the legacy multicast mechanism cannot cope with the video bandwidth requirements, which generates packet drops.

In presence of high level of congestion, the goodput at the application level is about 40% less for the leader-based approach and half less for the legacy multicast solution – in the same proportion than what we have observed for the packet loss parameter.

It is important to notice that the bulk of packet loss appears at the AP sending queue. So, the MAC retransmission mechanism used for the leader-based approach will not help much in improving the video quality. With such high level congestion, it is preferable to stop straight transmitting video streams.

We can imagine several ways to prevent such a situation. The video sender should be reactive to packet loss observed (at the RTP level) and should implement a mechanism to adapt its transmission accordingly. The sending queue at the AP could be monitored and a signal should be sent before an overflow occurs, or it could transmit a load indicator in each beacon frame. Another way to estimate channel load is to monitor the average contention window which is a good indicator of channel load.

6. CONCLUSION AND FUTURE WORK

We show with measurements that the legacy multicast transmission scheme is unusable due to its open-loop structure and can significantly degrades performance of other concurrent flows. Then, we evaluate the performance of the

[8]Latency and jitter parameters have more impact for interactive applications such as VoIP or videoconferencing.

[9]In this case, the multicast source is saturated.

recently proposed leader-based mechanism and compare it with the standard multicast solution. Our results clearly show that the new approach outperforms the legacy multicast mechanism while preventing multicast flows to swap all the resources of the WLAN. Such a measurement study is crucially missing in the literature today. For example, the analysis of packet loss correlation is very important to consider while selecting the leader station.

There is certainly more work to be done to thoroughly evaluate the leader-based mechanism. In future work, we will seek to extend the study in presence of mobile stations. In particular, the algorithm to select the leader should be able to dynamically track the receiver that experiences the worst channel conditions. We will also consider the case where some of the stations use the power safe mode, which can modify the characteristics of the multicast transmission.

We hope that our work will provide incentives to implement efficient multicast transmission mechanisms on future IEEE 802.11 cards, and possibly patches for our current devices.

Acknowledgements

This work has been partially supported by the French Ministry of Research RNRT Project "DIVINE" and within the framework of the "Saint Exupery"grant, Ministerio de Educacion de la Republica Argentina. The authors wish to thank Arnaud Legout for providing valuable comments in a preliminary version of the paper.

7. REFERENCES

[1] N. Choi, J. Ryu, Y. Seok, Y. Choi, and T. Kwon, "Unicast-Friendly Multicast in IEEE 802.11 Wireless LANs", IEEE CCNC, Jan 2006.

[2] S. Gopal, D. Raychaudhuri, "Experimental Evaluation of the TCP Simultaneous Send Problem in 802.11 Wireless Local Area Networks", ACM SIGCOMM E-WIND workshop, August 2005.

[3] S. K. S. Gupta, V. Shankar, and S. Lalwani, Reliable Multicast MAC Protocol for Wireless LANs, wIEEE International Conference on Communications, Vol. 1, 2003, pp. 93-97.

[4] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, "Performance anomaly of 802.11b", IEEE Infocom, April 2003.

[5] C. Hoffmann, M.H. Manshaei, and T. Turletti, "CLARA: Closed-Loop Adaptive Rate Allocation for IEEE 802.11 WLANs", IEEE WirelessCom'05, Hawai, USA, June 13-16, 2005.

[6] A. Kamerman and L. Monteban, "WaveLAN II: A high-performance wireless LAN for the unlicensed band", Bell Labs technical Journal, Tech. Rep., Summer 1997.

[7] D. Kotz, C. Newport, R. S. Gray, J. Liu, Y. Yuan, and Chip Elliott, "Experimental evaluation of wireless simulation assumptions", ACM MSWiM, October 2004.

[8] J. Kuri and S.K. Kasera, "Reliable multicast in multi-access wireless LANs", Wireless Networks, Vol. 7, No. 4, pp. 359-369, July 2001.

[9] J. Lacan and T. Perennou, "Evaluation of Error Control Mechanisms for 802.11b Multicast Transmissions", WinMee'06, April 2006.

[10] W. Lauppe, "Multicast Video over Wireless LANs", Internship report,
'Ecole Polytechnique Promotion 2002, July 2005.

[11] A. Ng, D.J.Leith, and D.W. Malone, "Experimental Evaluation of TCP Performance and Fairness in an 802.11e Test-bed", ACM SIGCOMM E-WIND workshop, August 2005.

[12] A. Majumdar, D.G. Sachs, IV. Kozintsez, K. Ramchandran, and M. Yeung, "Multicast and unicast real-time video streaming over wireless LANs , IEEE Transactions on CSVT, Vol. 12, No. 6, June 2002.

[13] H. Schulzrinne et al., "RTP: A Transport Protocol for Real-Time Applications", RFC-3550, July 2003.

[14] J. Vieron, T. Turletti, K. Salamatian, and C. Guillemot, "Source and channel adaptive rate control for multicast layered video transmission based on a clustering algorithm", in EURASIP Journal on Applied Signal Processing, Vol. 2004, No. 2, pp. 158-175, February 2004.

[15] J. Yeo, M. Youssef, A. Agrawala, "Characterizing the IEEE 802.11 Traffic: The Wireless Side", CS-TR-4570, March 2004.

[16] [web page] Ethereal, tethereal: http://www.ethereal.com [Accessed 15 May 2006]

[17] Ethereal Users List [web page]: http://www.ethereal.com/lists/ethereal-users/200405/msg00305.html [Accessed 15 May 2006]

[18] Ethereal supported filters[web page]: http://www.ethereal.com/docs/dfref/ [Accessed 15 May 2006]

[19] iperf [web page]: http://dast.nlanr.net/Projects/Iperf/ [Accessed 15 May 2006]

[20] Kismet sniffer [web page]: http://www.kismetwireless.net [Accessed 15 May 2006]

[21] libpcap [web page]: http://www.tcpdump.org [Accessed 15 May 2006]

[22] Madwifi driver [web page]: http://www.madwifi.org/ [Accessed 15 May 2006]

[23] tcpdump [web page]: http://www.tcpdump.org [Accessed 15 May 2006]

[24] tcpflow [web page]: http://www.circlemud.org/ jelson/software/tcpflow/ [Accessed 15 May 2006]

[25] tcptrace [web page]: http://jarok.cs.ohiou.edu/software/tcptrace/tcptrace.html [Accessed 15 May 2006]

[26] tstat [web page]: TCP statistic and analysis tool: http://tstat.tlc.polito.it/ [Accessed 15 May 2006]

[27] VideoLan Client (VLC) [web page]: http://www.videolan.org/vlc/ [Accessed 15 May 2006]

[28] WisMon [web page]: http://www-sop.inria.fr/planete/software/WisMon/ [Accessed 15 May 2006]