A Collision Detection Method for Multicast Transmissions in CSMA/CA Networks

Thomas Nilsson, Greger Wikstrand and Jerry Eriksson
Department of Computing Science
Umeå University
SE-901 87 Umeå, Sweden
{nilsson, greger, jerry}@cs.umu.se
fax: +46 786 61 26, phone: +46 786 9789
April 4, 2006

Summary

Multicast traffic in an IEEE 802.11 network has low reliability unlike unicast which is protected in three ways: 1) by an automatic repeat request protocol (ARQ), 2) by backing off in case of collisions, and 3) by adapting the link. These mechanisms are not available for multicast because of the inherent feedback problem arising in a shared communication channel.

This paper proposes and evaluates an algorithm called early multicast collision detection (EMCD) for increasing the reliability of multicast. Senders using EMCD will introduce an early pause in the transmission, perform a clear channel assessment (CCA), and if a collision is detected abort the transmission and schedule a retransmission. The time for the pause will be determined randomly for each station from a fixed discretized interval.
A probabilistic analysis is provided showing that EMCD is more efficient than ordinary multicast and can be made even more efficient by tuning parameters, for instance the length of the interval in which the pause is performed.

Simulations were performed to go beyond the scope of the analysis in terms of non-saturated conditions and mixed unicast/multicast situations. The simulations show that EMCD increases the reliability for multicast in the presence of unicast senders.

Index Terms—Quality of Service, CSMA/CA, Medium Access Control, 802.11, Collision Detection, Reliable Multicast, Performance Evaluation
Abstract

Compared to unicast traffic, multicast is not protected by any ARQ mechanism in 802.11 networks: collisions with other multicast and unicast transmissions are not detected and senders will not adapt to the contention situation by backing off. This results in an unreliable service for multicast transmissions. We propose early multicast collision detection (EMCD), an algorithm with the purpose of increasing the reliability of multicast transmissions in the MAC layer of an IEEE 802.11 network. A multicast sender using it will introduce an early pause in a transmission, perform a clear channel assessment (CCA), and if a collision is detected abort the transmission after a fixed time and schedule a retransmission. This allows for detecting collisions with both multicast and unicast transmissions but also adapting to the contention situation. A probabilistic analysis is provided showing that EMCD is more efficient than ordinary multicast and can be made even more efficient by tuning parameters. Simulations show that EMCD leads to increased reliability for multicast transmissions.

1 Introduction

It is difficult to provide the same kind of reliable service for multicast traffic in an IEEE 802.11 network as for unicast traffic [1, p. 29]. In an IEEE 802.11 network, unicast traffic is protected in three ways: 1) by an automatic repeat request protocol (ARQ), 2) by backing–off more in case of congestion, and 3) by adapting the link. These, or other reliability–enhancing mechanisms, are not available for multicast traffic. ARQ is problematic because of the ACK-implosion problem, i.e. multiple stations will send an ACK. Without ARQ, the back-off mechanism cannot be applied after a collision. Link adaptation is based on frame error rate, which is not available for the same reason.
Thus, such packets are more likely to be lost.

One further reason for multicast packet loss is collisions between access points (AP) in overlapping cells on the same channel and in the same extended service set (ESS). Any multicast stream must be transmitted by each access point in an ESS [1, p. 83] leading to a risk for synchronized, and therefore colliding traffic. Also, stations are explicitly forbidden to eavesdrop on access points with which they are not associated [1, p. 44] thus making it less efficient to use channel reservation techniques. A configuration like that can be conceived e.g. in an arena or an exhibition hall with free space propagation where a large numbers of access points have been distributed to maximize coverage and association capacity.

This work was done in the context of the Arena project (http://www.cdt.luth.se/projects/arena/). At a sport event, spectators were provided with an interactive match program in wireless terminals. Among the features of the application were the ability to playback live video and personalized replays in the phones. The efficient way to reach the crowds in such a context would be to use multicast for the multimedia information. In the case of multicast multimedia this will require minimizing packet losses to provide smooth playback and minimizing or bounding delays so that playback does not lag too far behind the live event. In particular the system must be robust against unicast interference as there will be no way to restrict such traffic on an unlicensed frequency band. We expect spectators to bring their own equipment from home, so any solution based on changes to their equipment is hardly feasible.

For clarity, we provide the following definition of MAC–layer multicast. It is based on the definition given by Sun et al. but adds the requirement of simultaneity [2]. Thus we do not consider the process in which a sta-
tion transmits the same multicast packet to several stations by sending it repeatedly, once for each recipient as multicast.

**Definition 1.** Multicast is the process in which a station simultaneously transmits the same data packet to some of its neighbors. A *multicast transmission* is a transmission from one sender, the AP in an IEEE 802.11 infrastructured network, to a group of zero or more recipients. Each multicast group has a special MAC-address recognized by its members.

Improving the reliability of multicast traffic is important since multicast traffic, e.g. streaming video, reaches at least as many users as unicast traffic. Thus, poor reliability for multicast traffic can be expected to have a higher negative impact on user satisfaction than the converse case for unicast traffic.

Not all multicast traffic has the same semantics. Sometimes it is important that all recipients receive all frames in a timely and orderly fashion and that the sender knows that they have done so, e.g. configuration and control. In our scenario, it is sufficient that most recipients receive most of the packets. Streaming video can tolerate and compensate for data loss in several different ways.

An algorithm, which would enable collision detection for multicast frames, was introduced in 1984 by Lo and Mouftah [3, 4] under the name Time Split Collision Detection or CSMA/TCD and again in 1986 by Rom [5]. We call it the “the Rom Algorithm”. It is the purpose of this paper to re-evaluate their algorithm(s) in terms of their suitability for solving the problems outlined above. We investigate the efficiency of the modified algorithm and the possibility of using it in an IEEE 802.11 network.

The rest of this paper is organized as follows. In Section 2 some other collision detection and avoidance techniques are described and categorized. In Section 3 our modified Rom Algorithm, called Early Multicast Collision
Detection (EMCD), is described along with the modifications required for applying it to an IEEE 802.11a network. We argue for relevant parameters to use and describe various collision scenarios. EMCD was evaluated through probabilistic analysis and simulations. The analysis is presented in Section 4 and the simulation results are presented in Section 5. Section 6 contains our conclusions.

2 Related work

Much work has been done in the area of channel reservation and collision resolution in MAC protocols for unicast traffic, e.g. [6], [7] and [8]. These solutions are not directly applicable to multicast traffic where multiple receivers are addressed and normal ARQ is not feasible. Below, we review some that have been developed for multicast. Basically, the problem with lost multicast traffic can be resolved in two ways: avoiding or detecting packet loss.

2.1 Channel Reservation and Collision Detection in IEEE 802.11

The medium access control (MAC) scheme used in the IEEE 802.11 standard is called the Distributed Coordination Function (DCF). It is a distributed contention based access method based on the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) algorithm. Carrier sensing, known as Clear Channel Assessment (CCA) in IEEE 802.11, is used to determine the status of the medium prior to attempting to access the channel. If the channel is determined to be busy or after a collision a station chooses a back-off counter (time to wait until retransmission) randomly from a Contention Window (CW), which is doubled after each unsuccessful attempt. The back-
off counter is decremented by one for each time slot the medium is idle, and as soon as the medium becomes busy the countdown is postponed until the medium has been idle during a DIFS.

Subsequent frame transmissions are separated by short time gaps on the channel called Inter Frame Spaces (IFS), used to differentiate control and data frames. The Short IFS (SIFS) is used when a station have seized the medium and need to keep it for the completion of a frame exchange sequence, e.g. between the data frame and the corresponding ACK frame. Data frames are separated by a DCF IFS (DIFS) which is longer than a SIFS.

There is an additional inter frame space called Extended Inter frame Space (EIFS) used by a receiving station to defer access to the channel after an unsuccessful reception of a packet. The EIFS is intended to provide sufficient time for another station, that did experience a correct reception of the frame, to transmit an ACK to the sender. As a result of this, the length of the EIFS is much longer than other inter frame spaces.

An optional way of accessing the channel, by an exchange of control frames prior to the data transmission, is included in DCF. The sender starts by sending a Request To Send (RTS) and the receiver responds with a Clear To Send (CTS), after first waiting a SIFS. The channel is then reserved and the pending data frame may be sent free of collisions. These control frames contain the duration of the complete frame exchange sequence which allows other stations, in range of the sender or the receiver or both, to defer access during the completion of the frame exchange sequence. The stations only in range of the receiver are referred to as hidden terminals (hidden from the sender).
2.2 Collision Avoidance Techniques

With the exception of Blackburst [9], channel reservation techniques for wireless MAC-layer multicast in IEEE 802.11 or other CSMA/CA protocols that we have found in the literature are based on variations on the RTS/CTS scheme described above. The main difference between the proposed methods is the number of stations which respond with a CTS, one station or all stations.

In the techniques with one CTS responder, it can either be a fixed and designated responder as in Robust Multicast [10], or the stations can take turns in a round-robin fashion as in BMW [11]. In the latter method, the CTS station will also piggy-back information on which frames it has received earlier. The first method suffers from the fact that the designated station might not be representative. The second solution might lead to frames being retransmitted after a long time if the round-robin queue is long.

In the techniques where all stations respond in turn the responses can either be sequential or simultaneous. In one method, all recipients send a CTS in an order determined by the multicast sender and announced in the RTS packet [2]. This method obviously has a large overhead and will scale poorly. In another scheme, each station waits a random interval in case someone else responds before sending their own CTS [12]. If the interval is too long (≤ DIFS) the stations will not have time to respond before other transmissions start. If it is too short it becomes very probable that several stations transmit at once.

Two methods where all stations send their CTS at the same time have been proposed to enable the multicast sender to decode the transmissions. In one case, the strongest (closest) sender is assumed to be sufficiently stronger than all other senders together so that the multicast sender can decode the
CTS [13]. Others argue that this is highly unlikely and instead propose using linearly independent codes for each sender letting the recipient decode all of them simultaneously [14]. This solution might require extensive changes in the physical layer.

A hybrid method has been proposed where a designated station sends the CTS but where stations who do not “agree” will send a “negative” CTS (NCTS) in order to cancel out the CTS [12]. This solution requires that all stations agree.

2.3 Collision Detection Techniques

Collision detection techniques can be further subdivided into in-transmission carrier sensing and ARQ approaches.

Kuri and Kasera also proposed an analogue hybrid scheme for ARQ [12]. In it, a designated station transmits an ACK and any station which has not received the multicast frame but detects the ACK will transmit a “negative” ACK (NAK) to collide with the ACK. This approach requires all stations to receive a given multicast packet correctly or not at all, i.e. this approach does not tolerate any packet degradation in marginal stations. A large number of multicast receivers with varying channel qualities will deteriorate the performance of this approach since it is enough that one receiver experiences a packet loss for a retransmission to be triggered.

Sheu et al. proposed a highly reliable broadcast scheme for ad hoc networks [15]. They modify the 802.11 MAC scheme to incorporate broadcast acknowledgments by dividing the DIFS time, following a broadcast transmission, into several mini-slots and require each station to contend for transmitting short acknowledgment frames. This ARQ approach does not scale to the number of multicast receivers and is limited by DIFS.
3 Early Multicast Collision Detection

We have adopted the algorithm proposed by Rom [5] for the purpose of detecting collisions experienced by multicast senders in an IEEE 802.11 network and thus enabling retransmissions of lost packets. This will increase the reliability for the otherwise unprotected multicast traffic but also minimize the overhead associated with collisions.

The Rom algorithm differs from non-persistent CSMA in its ability to detect collisions during the transmission. The main difference from CSMA is that a sender transmitting a packet pauses during the transmission and performs an additional carrier sense operation to detect other potential transmitters. A sender detecting a collision does not abort the transmission immediately but continues for a predetermined period, referred to as the Collision Detection Interval (CDI), to assure that a later scheduled pause does not miss the collision. If the channel is sensed idle during the CDI the transmission continues as normal.

Lo and Mouftah [4] and Rom [5] have proposed similar algorithms for collision detection for the CSMA family of protocols. Using the categories in Section 2 these, algorithms and EMCD are collision detection techniques. The work presented by Rom [5] is a more elaborate algorithm in the sense that the pausing sub interval is chosen from a uniform distribution, whereas Lo and Mouftah assume a non-slotted CSMA with the same pausing sub interval for all senders.

[Table 1 about here.]

3.1 Protocol Description

EMCD operates in three phases, see Algorithm 1. (A glossary of our no-
Algorithm 1 Early Multicast Collision Detection

\[
\text{success} = \text{false}, \quad \text{bo} = 0, \quad \text{retries} = 0
\]

\textbf{while} \ \neg \ \text{success} \ \text{AND} \ \text{retries} \leq R_{\text{max}} \ \textbf{do}

- Perform backoff/collision avoidance
- Perform Vanguard Transmission \( T_v \), duration from (3.1.2)

  \textbf{if} \ \text{CCA}() = \text{busy} \ \textbf{then}
  
  - retries = retries + 1
  - jam until end of CDI
  - bo \in U \{0, CW_{\text{min}}\}

  \textbf{else}

  - Perform main transmission \( T_m \)
  - success = true

\textbf{end if}

\textbf{end while}

The first phase is the vanguard transmission, the second phase is the carrier sensing phase, and the third phase is the main or jamming transmission. Since stations can transmit data at varying rates, frames of the same size may require different transmission durations. For this reason, EMCD is based on timing intervals rather than packet size. This is different from the Rom algorithm which implicitly assumes a fixed transmission rate for all transmitters [5].

We have chosen not to double the contention window for EMCD stations in order to prioritize the multicast traffic carried in the downlink. This is motivated by the unbalanced contention between up and downlink reported in [16]. The access point, alone contends for all the downlink traffic, whereas each station individually contends for its uplink traffic. This results in an unbalanced contention since the access point have no higher priority than a normal station in accessing the channel. Hence, unicast stations contending for the uplink can easily starve the downlink traffic under high load. Under saturated conditions with a total of \( N \) contending senders including the access point, the access point will only receive \( 1/N \) of the total capacity, i.e.
the same as all the other stations.

The retransmission mechanism used in IEEE 802.11 uses a short and long retry limit for short control packets (CTS and RTS) and longer data packets respectively. Recommended values are 4 for data packets and 7 for control packets [1]. The retransmission strategy is important to multimedia applications, that are more sensitive to delay and jitter. Retransmitting a data packet carrying video information may not make sense if the play out deadline has passed. There is a clear tradeoff between tolerance to packet losses and delay. We are assuming a lower retransmission limit \( R_{max} \) for multicast packet to minimize delays, see Section 5. Furthermore, as mentioned earlier EMCD is not doubling the contention window to prioritize the multicast traffic, this will give the multicast senders higher capacity and lower delays.

3.1.1 Collision Detection Interval

The length of the CDI, i.e. \( T_{CDI} \), affects the performance of EMCD in three ways. First, \( T_{CDI} \) determines the maximum number of unique transmission times that are selectable by the multicast senders. If the CDI is very short there will only be a few unique transmission times and hence the probability for deriving the same transmission time will be greater. Second, \( T_{CDI} \) has an effect on the detection probability for collisions with unicast senders. If the derived vanguard transmission is longer than the unicast transmission, the multicast sender will not detect the collision. Third, a long detection period will add additional delay and more overhead to the retransmission, mainly because the collision is detected later.

So the CDI should be as short as possible to detect a majority of the unicast transmissions, but still large enough to yield several unique trans-
mission times.

A static solution might be envisaged where every multicast sender would be assigned a $T_{v,i} \neq T_{v,j}$ for any stations $i \neq j$ to be used for every vanguard transmission; no collisions would go undetected. However, a static solution would require configuration of the multicast senders, which is contrary to the basic idea of decentralization in IEEE 802.11. It will not be considered in this paper.

### 3.1.2 Duration of the Vanguard Transmission

The duration of the vanguard transmission is derived from the expressions below:

$$T_v = T_{min} + X T_{\Delta},$$

$$X \in U \left\{ 0, \frac{T_{CDI} - T_{min}}{T_{\Delta}} \right\}. \quad (1)$$

$T_{min}$ is the minimum allowed or possible transmission time. $T_{\Delta}$ is the smallest time difference that can be detected between two vanguard transmission from different senders by the CCA and $T_{CDI}$ is the maximum duration of the vanguard transmission plus CIFS (see below). Figure 1 illustrates how three senders have selected different $T_v$. All EMCD stations in an ESS, i.e. stations including access points that implement and use EMCD, should use the same value for $T_{CDI}$.

[Figure 1 about here.]

### 3.1.3 Duration of the Carrier Sensing Phase

After the vanguard transmission, the station performs a CCA to detect other transmissions, see Algorithm 1. We refer to the interval in which the CCA
is performed as the Collision Inter frame Space (CIFS). Selecting a short CIFS minimizes overhead.

A lower bound on the CIFS is defined by the time required to perform a correct CCA (ordinarily $< 4\mu s$ [17, pp. 27 & 40]) the signal propagation time, and the time required to switch from transmitting to receiving and back again, i.e. the receive/transmit turnaround time. The first term is partially hardware dependent, the second term depends of the physical configuration of the arena, and the third term is hardware dependent.

An upper bound on the CIFS is defined by the SIFS, PIFS and DIFS timing intervals (16, 25 and 34 respectively in IEEE 802.11 [1, p. 85][17, p. 40]). It must be shorter than PIFS and DIFS of them to prevent other stations from detecting the channel idle, and consequently begin their transmissions in between $T_v$ and $T_m$. So, in an 802.11a WLAN the CIFS can be in the range $11 \leq \text{CIFS} \leq 25$.

### 3.2 Collision Scenarios

Figure 2 illustrates a collision between two multicast senders (1 and 2) starting their transmissions at the same time. Sender 2 will detect the collision before Sender 1, and continue by jamming the medium until the end of the CDI to assure that Sender 1 also detects the collision. At the end of the CDI they will end their transmissions and reschedule the collided packets according to Algorithm 1. The access point generating the shortest backoff time will win the next contention phase (Sender 2 in Figure 2).

[Figure 2 about here.]

In Figure 3 a multicast sender (1) and a unicast sender (2) start their transmissions simultaneously resulting in a collision. The carrier sense operation performed by station 1 will detect the collision. Station 2 will detect
the collision when it does not receive an ACK. Both stations will backoff before retransmitting their collided packets. If the packet had been shorter than $T_{\text{min}}$ the collision would not have been detected.

3.3 Compatibility

It is sufficient to implement EMCD in the multicast senders. Multicast recipients need no further information than what is required to reassemble the two parts of the packet. The multicast senders could make that process entirely transparent by refragmenting the packet at a higher protocol level. That would also circumvent the injunctions in the 802.11 standard against fragmenting multicast packets [1, p. 71] and making earlier fragments of a packet shorter than later fragments [1, p. 93].

3.4 Protocol Overhead

The extra carrier sense operation in EMCD imposes an additional overhead on transmissions. In Table 2 the protocol overhead is shown for unicast, multicast and multicast using EMCD, under the assumption that the data payload is 1496 bytes and the rate is 6 Mbps. Note that only the packet overhead is considered in the table and not the back-off time, DIFS etc. Multicast packets are split into two transmissions by the EMCD algorithm and the overhead per packet is doubled. The overhead for EMCD is slightly larger than for unicast.
4 Analytical Evaluation

The Rom algorithm was evaluated analytically in the original paper [5]. The analysis uses an infinite population of users which as an aggregate form a Poisson arrival process. In general the time to transmit a packet is shorter than the arrival rate so the channel is far from saturation [18].

Here it is assumed that there is a finite population of \( k \) stations. Each station operates under saturation conditions, i.e. it always has a packet ready for transmission. We further assume that there are only EMCD stations. We also assume that no packets are lost because of noise and that the EIFS mechanism works perfectly.

The first two assumptions are fairly standard in this type of analysis, cf. [19]. The following three assumptions are made for simplicity of analysis, the impact of breaking them are considered in the simulation section later.

4.1 Channel Utilization

Consider a transmission cycle consisting of zero or more unsuccessful transmission attempts followed by a successful transmission. The channel utilization \( \rho \) will be the time required for the transmission of useful data in a cycle divided by the length of the cycle. (A glossary of our notation is provided in Table 1.) Let \( E[U] \), \( E[I] \) and \( E[B] \) be the expected useful, idle and busy times of a cycle then [18, 4, 19]

\[
\rho = \frac{E[U]}{E[I] + E[B]}. \tag{2}
\]

\( E[U] \) is easily obtained from the average packet length \( m \) in units of time. The expected idle and busy times are given in (13) and (11) respectively. Let \( n \) denote the number of slots in the CDI and \( k \) the number of stations.
Single Station Backoff  Under the assumption of independent backoff between stations the backoff of a single station can be described using a Markov–chain. By solving for the stationary probability distribution it is possible to obtain the probability that a single station has a backoff counter of zero and will transmit. It was given as $\tau = 2/(W + 1)$ by Bianchi [20, eq. 8], where $W$ is $CW+1$. We shall use a slightly different notation $l = W - 1$, and write

$$\tau(l) = \frac{2}{l + 2}.$$  \hfill (3)

Multiple Stations Backoff  For Bianchi, it was sufficient to determine the probabilities that there was a transmission from one station and if so if that station was the only one transmitting. In our case we must know how many stations collide to be able to derive the probability that a collision will be detected.

The probability that $i$ out of $k$ stations have finished backing off at the same time will be given by the binomial distribution with $\tau(l)$ as the success probability and we write

$$p(i, k, l) = \begin{cases} \binom{k}{i} \tau(l)^i (1 - \tau(l))^{k-i} & i \leq k, \\ 0 & i > k. \end{cases}$$  \hfill (4)

'Winners’ in Each Step  Recall that the EIFS affects what happens after a collision. Non–colliding stations will defer until a successful transmission occurs or to the end of the EIFS. Let $e$ be the length of the EIFS in slots and $P(r, i)$ be the probability that $i$ stations are left before contention phase $r$ begins. First, consider the case where the EIFS is longer than the contention window, i.e. $e \geq l$, then the probability $w(r, i)$ for $i$ ‘winning’ stations is
given by

\[ w(r, i) = \sum_{j=1}^{k} \frac{p(i, j, l) P(r, j)}{1 - p(0, j, l)}. \]  

(5)

Under our assumptions we start with \( k \) stations in the first step. The number of stations left is basically the same as the number of winning stations in the preceding steps except for the following: if there is a single winning station in the earlier step then there will be no stations left in the following step and we can write

\[ P(r, 1 \ldots k) = \begin{cases} 
(0, \ldots, 0, 1) & r = 1, \\
(0, w(r-1, 2), \ldots, w(r-1, k)) & r > 1. 
\end{cases} \]  

(6)

If \( e < l \) two cases can occur, either at least one station finishes backing off before the end of the EIFS or the EIFS ends with no stations having finished backing off. The probability that a station \( i \) will have a backoff counter \( bo_i \) higher than the EIFS timeout is

\[ P(bo_i > EIFS) = \frac{l - e}{l} = 1 - \frac{e}{l}. \]

The probability that none of the stations have finished backing off before the end of the EIFS is

\[ P(\min_{i=1}^{k'} bo_i \geq EIFS) = \left( \frac{l - e}{l} \right)^{k'} \]  

(7)

where \( k' \) is the number of initially participating stations, \( 1 \leq k' \leq k \). So the
probability that at least one station finishes backing off is

\[ P(\min_{i=1}^{k'} b_i < \text{EIFS}) = 1 - \left( \frac{l - e}{l} \right)^{k'} . \]  

(8)

Considering the two cases, the probability that there will be \( i \) ‘winning’ stations in step \( r \) is given by combining (5), (7) and (8). After some trivial simplifications we have

\[ w(r, i) = \sum_{j=1}^{k} \left[ \frac{p(i, j, l)}{1 - p(0, j, l)} P(r, j) \left( 1 - \left( \frac{l - e}{l} \right)^j \right) \right] + \frac{p(i, k, l)}{1 - p(0, k, l)} \sum_{j=1}^{k} \left[ P(r, j) \left( \frac{l - e}{l} \right)^j \right] . \]  

(9)

**Busy Times** In each transmission attempt, there are three possible outcomes: Only one station transmits and the transmission is considered to be successful \((s)\). Two or more stations transmits and either the collision is detected \((CD)\) or it is not detected \((CND)\). The busy times are derived from Algorithm 1 and the IEEE 802.11a standard [17].

The busy time will be the same if the transmission is successful or not and will be given by \( T_s = T_{CND} = m + 155 \), see Table 2. If the collision is detected the busy time will be \( T_{CDI} \).

The probability of not detecting a collision is the probability that all stations will choose the same slot to pause in and is

\[ P(\text{detect}) = n^{1-k} \]

if there are \( k \) transmitting stations. The risk of not detecting a collision will decrease with increasing \( n \) and \( k \). The expected busy for a single
transmission attempt is

\[ E[\text{busy}] = n^{1-k}T_{CND} + (1-n^{1-k})T_{CDI}. \] (10)

The expected busy time for all attempts in a cycle is

\[ E[B] = \sum_{r=1}^{\infty} \sum_{i=1}^{k} w(r,i)(n^{1-i}T_{CND} + (1-n^{1-i})T_{CDI}) \] (11)

If \( n = 0 \), i.e. ordinary multicast, then the right-hand factor will reduce to \( T_{CND} \) which is obviously larger than \( n^{1-i}T_{CND} + (1-n^{1-i})T_{CDI} \) for any positive \( n \) and \( i \).

**Idle Time** Remember that the expected value of a discrete stochastic variable is the sum of the probabilities of each outcome times its numerical value. The channel will be idle until at least one station has finished backing off. The probability that at least 1 station has finished backing off is \( 1 - p(0, k, l) \) from (5). The expected number of backoff steps before that happens is given by the geometric distribution

\[ E[\text{backoff}] = \frac{p(0, i, l)}{1-p(0, i, l)}. \] (12)

Each transmission attempt will be preceded by a DIFS and with a certain probability an EIFS. Hence the expected idle time in a cycle is

\[ E[I] = \sum_{r=1}^{\infty} \sum_{i=1}^{k} q(r,i)(DIFS + et_{slot} \left( \frac{l-e}{l} \right)^i + \frac{p(0, i, l)t_{slot}}{1-p(0, i, l)}). \] (13)

The channel utilization is easily found by substituting (11) and (13) in (2).
4.2 Results

The analysis was validated by comparison to simulations in GloMoSim see Section 5 for details. Figure 4 shows that the analytical results are a good match for the simulated results as long as the number of stations is small.

The lowest curves, for $n = 1$, shows what performance could be expected from normal multicast but with some additional overhead. Channel utilization decreases asymptotically towards a value around .35. The upper two pairs of curves show an interesting behavior in that they first decrease then increase slightly. The effect is more pronounced in the simulated curves. This is an effect of the backoff no longer being independent for each station when there are a large number of stations as explained in Section 4.3.

[Figure 4 about here.]

The optimal number of steps $n_{opt}$ as a function of the number of stations and the effect on channel utilization is presented in Figure 5. The figure shows two things: first, the optimal number of slots decreases with the number of stations and second, the choice of $n$ has smaller impact for lower numbers of stations.

[Figure 5 about here.]

4.3 A Note on Asymptotic Behavior

As the number of stations $k$ grow in relation to the number of available backoff slots $l+1$ the preceding analysis will become increasingly inaccurate. It is based on the assumption that almost all channel allocations will start with a competition for the medium during at least one backoff time–slot.

What will happen instead is that we will end up with a number of stations $1 \ldots k$ with $bo = 0$. All of them will try to transmit (and fail). After that
they will choose a new backoff value so that on average only 1/(l+1) stations will remain with bo = 0. This will reoccur until only zero or one stations are left. In the first case, all stations will backoff once. In the second case a successful transmission will occur.

5 Simulation Results

We extended the analytical evaluation from the previous section with simulations in order to examine factors which are beyond the scope of the analysis. The simulations were performed in the discrete event simulator GloMoSim [21].

We are mainly interested in reliability and efficiency. Efficiency is often measured as throughput which is hard to define. The aggregated throughput for multicast cannot be easily calculated because there are multiple recipients and some of them may receive the packet while other do not. Consequently, it is difficult to know if a multicast packet should be included in the aggregated throughput or not. For instance, Tourhilles considered only a single broadcast receiver [10]. Another way is to consider the average reception. We consider the average packet delivery ratio which is more a measure of reliability than efficiency.

The following metrics are used: delay and ratio of assumed success to measure efficiency and packet delivery ratio to measure reliability. The delay is the time from when a packet arrives at the MAC layer in the sender until it is successfully received. The average packet delivery ratio is defined as the average ratio of multicast packets received by the multicast group to the unique packets offered for transmission. Ratio of assumed success is defined as (#main transmissions)/(#vanguard transmissions). When a main transmission begins the transmission is assumed successful. More than
one vanguard transmission per main transmission will result in a ratio of assumed success smaller than 1.

5.1 Traffic and Radio Model

In each simulated scenario a specific number of access points are evenly distributed onto a quadratic area with sides 200 meters long. The specific number of stations are varied for the different simulation scenarios and the position of each station is randomly set within the simulation area.

We assume that there is no mobility. Each station is associated with the closest access point. The association is kept during the whole simulation. All stations in a BSS are members of a single multicast group.

Both the multicast and unicast traffic assumes a constant bit rate model with a packet size of 1496 bytes.

Two different scenarios are simulated. In the first scenario the multicast senders have a fixed bit rate of 500 kbps and the aggregated load from unicast senders is gradually increased by introducing more senders until the saturated condition is achieved. In this scenario, each unicast sender adds a packet rate of 4 packets per second to the aggregated load. In the second scenario with saturated conditions each sender constantly has a new packet to send.

For the modeling the path loss we have used a noise accumulating two–ray path loss model and for fading we assume a Rician distribution with $K = 5$. This represents a strong line of sight path between the transmitter and receiver, resembling an open arena environment. The output power is set to 15 dBm. All access points are assumed to use the same frequency channel resulting in overlapping basic service sets. The data rate for all transmissions is set to 6 Mbps.
5.2 Variable load scenario

In the first simulation scenario EMCD is evaluated under variable load. In the simulation there are four and nine overlapping cells (access points) with both multicast and unicast senders.

[Figure 6 about here.]

In Figure 6, the average packet delivery ratio is plotted against the number of unicast senders. For standard multicast (802.11) the packet delivery ratio decreases rapidly as the offered load increases, when more unicast senders join the network. When the medium becomes saturated, for around 70 stations, the rate of decrease becomes smaller mainly as a result of much higher \( CW \) values for the unicast senders. EMCD is capable of maintaining a relatively high packet delivery ratio for an increasing load. The upper curves are for four and nine access points and two values of \( n \). The effect of increasing \( n \) is more evident in the case of nine access points where more collisions are between the access points and more slots will increase the detection probability. Moreover, the saturation points is reached more quickly for EMCD since retransmissions are performed that increase the load.

In Figure 7 the ratio of assumed success is shown for an increasing number of unicast senders. Increasing the number of unicast senders and/or access points require more vanguard transmissions per main transmission due to collisions. The fraction of collisions experienced between multicast senders are higher for low and high loads. For lower loads, few unicast senders are present and a majority of the collisions are between multicast senders. For high loads, the unicast senders have much higher values of \( CW \) then the multicast senders (having \( CW_{\text{min}} \)) resulting in more collisions between the multicast senders. This is why the effect of increasing \( n \) is more
distinct for these two load levels.

[Figure 7 about here.]

5.3 Saturated load scenario

In the second scenario, a fully saturated network with four access points, the aim is to illustrate the tradeoff between packet losses and delay when changing some of the parameters.

[Figure 8 about here.]

In Figure 8 the average packet delivery ratio is shown for three different retransmission limits $R_{\text{max}}$ and two sizes of $n$. The delivery ratio is increasing for higher values of $R_{\text{max}}$. The effect of changing the number of slots $n$ depends on the fraction of unicast senders in the system. When the fraction of unicast senders are low, the majority of collisions will be between multicast senders and if $n$ is increased some gain in performance is achieved. However, this effect is less significant when the fraction of unicast senders are higher. Only a small percentage of the collisions will be between multicast senders and increasing $n$ has smaller effect.

When $R_{\text{max}} = 4$ and only a few slots exists the packet delivery ratio may actually increase when more unicast senders enters the network. This is because all collisions with unicast senders will be detected, leading to more retries and so the success probability will increase.

[Figure 9 about here.]

The average delay is shown in Figure 9. The delay is evidently increasing for higher values of $R_{\text{max}}$. Changing $n$ has much smaller impact on the delay, especially for smaller values of $R_{\text{max}}$. The two curves at the bottom
are almost indistinguishable while there is more difference in the two upper curves.

Clearly, there is a tradeoff between delay and packet delivery ratio which is determined by $R_{\text{max}}$. In the case of a streaming application the choice of $R_{\text{max}}$ may be important to maximize the utility of the service.

6 Conclusion

We have adapted the Rom algorithm to an IEEE 802.11 network with the purpose of detecting and recovering from multicast packet losses due to collisions. We call the algorithm Early Multicast Collision Detection (EMCD). We have used probabilistic analysis and simulations to study the performance of EMCD.

The main contributions of this paper are a) that we have adapted the Rom algorithm to contemporary CSMA/CA networks and MAC layer multicast, b) that we have provided a probabilistic analysis of its performance under saturated load conditions and c) that we have provided simulation results for the algorithm under both saturated and unsaturated conditions in mixed networks.

Some of the key findings are that EMCD significantly reduces undetected packet losses caused by collisions and reduces the time wasted on collisions (see (11)). The net effect is higher efficiency and reliability.

Only a relatively low number of slots are needed to achieve optimal channel utilization. With a large number of stations, $n = 7$ is optimal. With fewer stations $n$ is higher but the system is also less sensitive to changes in $n$. Adding unicast to the mix changes the situation slightly, on one hand fewer slots are needed to detect a collision as long as there is at least one unicast station involved, on the other hand ending the transmission at $T_{\text{CDI}}$ has
less effect as the unicast transmission will continue anyway. An acceptable choice in many situations could be $n = 7$.

References


Thomas Nilsson is a Ph.D. student at Umeå University, Sweden. He focuses on resource optimization and Quality of Service in wireless networks. Thomas received his Master of Science degree in computing science in 2002 and his licenicate degree in 2005, both from Umeå University. He was born in Umeå, Sweden in 1978. Thomas Nilsson is a student member of the IEEE.

Greger Wikstrand is a doctoral student at Umeå University. He was born in Uppsala, Sweden in 1972. He obtained the Master of Science in Engineering Physics degree from Uppsala University in 1998 and the Licenciate of Engineering degree in Computing Science from Umeå University in 2003. He was a doctoral student at Linköping University during 1998 and 1999. In 2000 he was with Procter & Gamble in Stockholm, Sweden. During 2001–2002 he worked as a research project manager at Ericsson AB. Greger Wikstrand is a student member of the ACM and the IEEE.

Jerry Eriksson is a senior lecturer at Umeå university. He obtained the Master of Science in Computing Science from Umeå University in 1990 and the Ph.D. degree from the same university in 1995. His doctoral dissertation (1995) treats the optimization and regularization of nonlinear least squares
problems. He worked at Ericsson from 1997 to 2000. His current research interests involves telecommunication and resource optimization.
List of Figures

1. Different $T_v$ generated by three senders. ............................................. 33
2. Collision between two multicast senders is detected by EMCD. Retransmissions are scheduled. ......................................................... 34
3. Collision between a multicast and a unicast transmission. Both stations detect it and schedule retransmissions. .......................... 35
4. Channel utilization $\rho$ as a function of the number of stations $k$ with CW $l = 15$, simulated and analytical. The relative mean square error is $\approx 2.1\%$. ......................................................... 36
5. Selecting the optimal number of slots $n$. .............................................. 37
6. The average packet delivery ratio for EMCD and standard 802.11 for four and nine access points. ......................................................... 38
7. The ratio of assumed success versus an increasing number of unicast sender for four and nine access points. ............................... 39
8. The average delivery ratio for three different values of $R_{\text{max}}$ and two values of $n$. ................................................................. 40
9. The average delay for three different values of $R_{\text{max}}$ and two values of $n$. ................................................................. 41
Figure 1: Different $T_v$ generated by three senders.
Figure 2: Collision between two multicast senders is detected by EMCD. Retransmissions are scheduled.
Figure 3: Collision between a multicast and a unicast transmission. Both stations detect it and schedule retransmissions.
Figure 4: Channel utilization $\rho$ as a function of the number of stations $k$ with CW $l = 15$, simulated and analytical. The relative mean square error is $\approx 2.1\%$. 
Figure 5: Selecting the optimal number of slots $n$. 
Figure 6: The average packet delivery ratio for EMCD and standard 802.11 for four and nine access points.
Figure 7: The ratio of assumed success versus an increasing number of unicast sender for four and nine access points.
Figure 8: The average delivery ratio for three different values of $R_{\text{max}}$ and two values of $n$. 
Figure 9: The average delay for three different values of $R_{\text{max}}$ and two values of $n$. 
List of Tables

1  Glossary ................................................................. 43
2  Protocol overhead in $\mu$s for multicast using EMCD, multicast and unicast. The payload length is 1496 bytes and the rate is 6 Mbps. ................................................................. 44
\begin{itemize}
\item $\rho$ \quad channel utilization
\item $l$ \quad contention window size
\item $k$ \quad number of stations
\item $n$ \quad number of slots
\item $\tau$ \quad probability for a station to have $bo = 0$
\item $bo$ \quad current backoff of a station
\item $p(i, k, l)$ \quad probability that $i$ stations finish backoff simultaneously
\item $w(r, i)$ \quad probability that $i$ stations backoff simultaneously in the $r$:th attempt
\item $P(r, i)$ \quad probability of $i$ contending stations in the $r$:th attempt
\item $E[U]$ \quad expected useful time
\item $E[B]$ \quad expected busy time
\item $E[I]$ \quad expected idle time
\item $CW_{\text{min}}$ \quad minimum backoff window size
\item $T_{\text{CDI}}$ \quad duration of the collision detection interval
\item $T_m$ \quad duration of main transmission
\item $T_{\text{min}}$ \quad duration of shortest possible transmission
\item $T_s$ \quad duration of a successful transmission
\item $T_{\text{CND}}$ \quad duration of an unsuccessful transmission
\item $T_v$ \quad duration of vanguard transmission
\item $T_\Delta$ \quad smallest detectable time difference
\item CCA \quad clear channel assessment
\item CIFS \quad collision inter frame space
\item SIFS \quad short inter frame space
\item DIFS \quad DCF inter frame space
\item EIFS \quad extended inter frame space
\end{itemize}

Table 1: Glossary
<table>
<thead>
<tr>
<th></th>
<th>EMCD</th>
<th>Multicast</th>
<th>Unicast</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preamble</td>
<td>40</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>ACK</td>
<td>-</td>
<td>-</td>
<td>44</td>
</tr>
<tr>
<td>SIFS</td>
<td>-</td>
<td>-</td>
<td>16</td>
</tr>
<tr>
<td>CIFS</td>
<td>11</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Data</td>
<td>1996</td>
<td>1996</td>
<td>1996</td>
</tr>
<tr>
<td>Mac header</td>
<td>104</td>
<td>52</td>
<td>52</td>
</tr>
<tr>
<td>Total µs</td>
<td>2151</td>
<td>2068</td>
<td>2128</td>
</tr>
<tr>
<td>%</td>
<td>7.2</td>
<td>3.55</td>
<td>6.22</td>
</tr>
</tbody>
</table>

Table 2: Protocol overhead in µs for multicast using EMCD, multicast and unicast. The payload length is 1496 bytes and the rate is 6 Mbps.