Collision Avoidance and Resolution Multiple Access for Multichannel Wireless Networks

RODRIGO GARCÉS
rgarces@metricom.com
Metricom Inc.
980 University Avenue
Los Gatos, CA 95032

J.J. GARCIA-LUNA-ACEVES
Computer Engineering Department
University of California
Santa Cruz, California 95064
Networking and Security Center
Sun Microsystems Laboratories
Palo Alto, California 94303

Abstract—We introduce and analyze CARMA-MC (for Collision Avoidance and Resolution Multiple Access MultiChannel), a new stable channel access protocol for multihop wireless networks with multiple channels. CARMA-MC relies on the assignment of a unique channel and a unique identifier to each node to support correct deterministic collision resolution in the presence of hidden terminals. CARMA-MC dynamically divides the channel of each node into cycles of variable length; each cycle consists of one or more receiving periods and a transmission period. During the receiving period, stations with one or more packets to send compete for the right to acquire the floor of a particular receiver's channel using a deterministic tree-splitting algorithm. Each receiving period consists of collision resolution steps. A single round of collision resolution (i.e., a success, and idle or a collision of control packets) is allowed in each contention step. The receiving period is initiated by the receiver and takes place in the channel assigned to the receiver station. The channel utilization and packet delays are studied analytically and by simulation.

I. INTRODUCTION

Collisions in a packet-radio network can be cause by *direct* or by *secondary* interference. Direct interference occurs when two neighboring nodes transmit to each other at the same time. Secondary interference occurs when two or more stations unaware of each other's existence transmit to the same receiver at the same time or when a station is transmitting to its neighbor and a third stations transmission to some other station causes an interference. This problem was first introduced by Tobagi and Kleinrock [15] and is known in the literature as the hidden terminal problem. Several approaches have been proposed in the past to resolve the hidden terminal problem, and collision-avoidance protocols have recently received considerable attention (e.g., [13], [6]). In a collision-avoidance protocol, sender and receiver collaborate trying to avoid data packets from colliding with other packets at the receiver. However, as traffic load increases in the network, the collision of collision-avoidance control packets increases and throughput in the system drops. A way to stabilize the operation of contention-based protocols is by means of collision resolution mechanisms.

Several stable MAC protocols have been proposed in the past based on tree-splitting algorithms for collision resolution (e.g., [4], [7], [20]). Those protocols in which data packets are used to resolve collisions achieve throughput below 0.6 [22] for a single channel and fully connected networks. Several MAC

protocols have been proposed that implement collision resolution using either control packets that are much smaller than data packets, or are based on the ability of the transmitter to abort transmission rapidly after detecting collision (e.g., [2], [8], [14]). Among those stable MAC protocols that achieve high throughput, some build a separate queue for the transmission of data packets, in addition to the stack or queue of the control packets used for collision resolution. However, the stable collision resolution approaches reported to date operate in fully-connected networks or networks based on central base stations.

On the other hand, several reservation based protocols have been proposed (e.g., [1], [10], [12], [14]) which provide stability at high load levels, and efficient service at low load levels. Resource auction protocols, i.e., [1], [14], require a significant amount of overhead for each auction period and are difficult to implement. On the other hand, PRMA [10] is relatively easy to implement but uses a fixed frame length which can lead to starvation if the number of active stations is large. These protocols all require a base station, and do not operate in a network with hidden terminals. The limitation of these schemes is that most of them require the use of a base station which is a single point of failure.

This paper presents an approach to utilizing collision resolution in multihop wireless networks by taking advantage of unique channel (or code) assignments to network nodes. In the past, multichannel networks have been constructed using multiple transceivers operating on separate fixed channels [21]. Such devices were expensive to construct. However, current transceiver technology (e.g., Metricom Inc. new generation network devices), enables radio devices with as many as 260 channels to be controlled by a single DSP, enabling radios to switch from one channel to the other within 1 μsec. This allows multichannel networks to be constructed inexpensively using a single device at each station. In addition, using multiple channels renders better delay characteristics than single-channel networks [16], [17], [19] and have better fault tolerance against fading and noise [5], [17]. Early work in protocol design for multichannel networks used CSMA or ALOHA protocols in slotted multiple channels [18]. A reservation protocol over multiple channels is investigated in [16] for satellite communication systems. A sequential multichannel system which uses

1. REPORT DATE **2000**	2. REPORT TYPE	3. DATES COVERED **00-00-2000 to 00-00-2000**
4. TITLE AND SUBTITLE **Collision Avoidance and Resolution Multiple Access for Multichannel Wireless Networks**		5a. CONTRACT NUMBER
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of California at Santa Cruz,Department of Computer Engineering,Santa Cruz,CA,95064**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S)
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**		
13. SUPPLEMENTARY NOTES		
14. ABSTRACT		
15. SUBJECT TERMS		

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**8**	

CSMA/CA on each channel to dynamically assign stations to channels is presented in [3]. Analysis of multi-hop multichannel networks using CDMA in sparse networks with receiver-based, transmitter-based, pairwise-based, and common channel assignment is presented in [11].

In this paper we introduce a new stable receiver-initiated multiple access protocol with collision resolution called **C**ollision **A**voidance and **R**esolution **M**ultiple **A**ccess **M**ulti**C**hannel (CARMA-MC) protocol. CARMA-MC operates in a multi-channel network in which hidden terminals may exist. It assumes that each network node is assigned a unique identifier and a unique channel, at least within the two-hop neighborhood of any network node. CARMA-MC is a receiver initiated protocol that dynamically divides the channel assigned to each receiver into receiving and transmitting periods. The transmission period has a maximum-length duration and each receiving period consists of collision resolution steps, i.e., of success, idle or collision of control packets. The protocol maintains a stack for the transmission of control packets used in collision resolution.

During the receiving period CARMA-MC uses a deterministic tree-splitting algorithm and an RTR/RTS/CTS exchange. A receiving period is initiated by the receiver sending a ready-to-receive (RTR) signal and takes place in the channel assigned to the receiver station. During contention intervals a station attempts to acquire the floor by sending an RTS to the intended receiver who, in turn, sends a CTS if the received RTS is error-free. RTSs are sent according to a deterministic tree-splitting algorithm that resolves all the requests that arrive during the same receiving interval. A packet is transmitted from the station that has acquired the floor by successfully completing a collision-resolution round. The control packets used in each contention step are much smaller than a single data packet. Because CARMA-MC uses a deterministic collision-resolution mechanism, average delays incurred in the network are bounded and are a function of the number of one-hop neighbors of a receiver. In stark contrast to prior approaches to collision resolution, CARMA-MC operates correctly in multihop networks with hidden terminals.

The rest of this paper is organized as follows. Section II describes CARMA-MC in detail. In Section III we present the worst case packet delay and channel utilization. Section IV compares the analytical results with the simulation. The analytical results are very close to the results obtained by the simulation, and this validates the approximations made in the analysis. Our results confirm that CARMA-MC is stable at any load level, and that it provides high throughput and bounded delays by first avoiding collisions of data packets and then efficiently resolving collisions of control packets.

II. CARMA-MC

A. Definitions and Assumptions

In CARMA-MC, the channel access time is divided into receiving and transmitting periods. Each station has a unique receiving channel that is used by its one-hop neighbors to transmit packets. A station is allowed to transmit packets in any of the unique receiving channels assigned to its one-hop neighbors. Therefore, stations switch from their default receiving frequency to any of the frequencies assigned to its one-hop neighbors.

We assume that there exists a mechanism that ensures that no station is assigned the same receiving frequency as any of its two-hop neighbors. Each station has knowledge of the assigned frequencies for all its one-hop neighbors. The assignment of frequencies can be achieved by applying the distributed assignment of codes algorithm proposed in [9]. This algorithm assigns a different frequency to each station within a two-hops neighborhood, provided that the number of frequencies available for assignment is at least $n \geq d_{max}^2 - d_{max} + 2$, where d_{max} is the maximum number of one-hop neighbors that any station can have. This mechanism eliminates co-channel interference and avoids the hidden terminal problem.

Besides the assignment of a unique channel, each station is also assigned a unique identifier (ID), that is known by all its one-hop neighbors. This can be achieved by exchanging the ID information at the same time that the receiving frequencies are assigned, i.e., applying the distributed assignment algorithm.

Stations in CARMA-MC are half duplex; they can be senders or receivers. A station in the sender state participates in a collision-resolution interval (CRI) based on the deterministic tree-splitting algorithm introduced in [8]. The deterministic tree-splitting algorithm resolve collisions among competing senders. The CRI evolves in terms of collision-resolution steps, where the size of a CRI is bounded and is a function of the number of senders (see [8] for more details). We assume a *ternary feedback* model, i.e., there are three types of collision-resolution steps: *collision*, *success*, and *idle*. Collision-resolution steps follows a handshake procedure meant to eliminate collisions among data packets. This procedure is known in the literature as "floor acquisition" [6]. In a single-channel network, floor acquisition entails allowing one and only one station at a time to send data packets without collisions. To achieve this, a station that wishes to send one or multiple data packets must send a request-to-send packet (RTS) to an intended destination and receive a clear-to-send packet (CTS) from it, before it is allowed to transmit any data. RTSs are required to last a minimum amount of time that is a function of the channel propagation time.

In CARMA-MC a station in the sender state is allowed to participate in the CRI in the unique receiving frequency assigned to its one-hop intended destination.

It is the responsibility of the receiver to initiate and gear the collision-resolution interval (CRI). The receiver can be visualize as the master of its one-hop neighbors. The receiver uses the unique ID to resolve contentions among transmitters. Like previous efficient MAC protocols based on tree-splitting algorithms, the receiver maintains a stack and two variables, $LowID$ and $HiID$. $LowID$ and $HiID$ are respectively the lowest and highest ID numbers of the stations that are initially allowed to transmit. Together, they define the allowed ID interval, $(LowID, HiID)$. The allowed ID interval is broadcasted by the receiver to all its one-hop neighbors in a ready-to-receive packet (RTR) at the beginning of every collision resolution step.

While a station is a receiver, it transmits an RTR at the beginning of each collision-resolution step and transmits CTSs in response to RTSs. It receives RTSs and data packets from its one-hop neighbors. All packet exchanges take place in the receivers assigned channel.

On the other hand, a station in the sender state, waits on the intended receiver's channel for RTRs and CTSs and transmits RTSs and data packets on the the same channel.

B. Information Maintained and Exchanged

Information is maintained, exchanged, and broadcasted by the receiver station. Each station is assigned a unique identifier and a unique receiving channel within the two-hop neighborhood of any network node. The receiver station maintains a stack and two variables $LowID$ and $HiID$. Recall that $LowID$ is the lowest ID number that is allowed to send an RTS and it is initially set to 1. On the other hand, $HiID$ is the highest ID number that is allowed to send an RTS and it is initially set to the largest number of one-hop stations allowed in the network. $LowID$ and $HiID$ define the allowed ID-number interval, $(LowID, HiID)$, that can send RTSs. If the ID of a station is not within the allowed ID interval, then the station is not allowed to send its RTS. The RTSs and CTSs specify the IDs of the sender and of the intended receiver. Finally, the stack is the storage mechanism for ID intervals that are waiting for permission to send an RTS.

Throughout the paper we assume that each station knows the maximum number of one-hop stations allowed in the network and the maximum propagation delay.

C. Basic Operation

If a station does not have a data packet to send, it returns to its receiving channel broadcasting an RTR initiating a new CRI. The allowed ID interval is broadcasted by the receiver to all its one-hop neighbors in a ready-to-receive packet (RTR). An RTR is also transmitted by the receiver at the beginning of every contention step, allowing new stations to know the state of the CRI. The allowed ID interval as well the unique receiver ID are embedded within the RTR.

Once a station initiates a new CRI it remains as a receiver until the end of the current CRI. If at the end of the current CRI the station has a data packet to send then it switches to the channel of its intended receiver and becomes a sender, participating in the CRI of its intended receiver. On the other hand, if at the end of the CRI the station does not have a packet to send, then it remains as a receiver initiating a new CRI. Any station engaged in a CRI as a sender must remain in the receiver's channel for at least the duration of a maximum CRI.

As shown in Fig. 1, the channel access time in CARMA-MC is divided into cycles, consisting of receiving periods and transmission periods. The CRI in receiving period is a sequence of collision-resolution steps, each initiated by an RTR. The one-hop stations participating in the CRI are assumed to constantly monitor the state of the channel while they are not transmitting. It is assumed that all one-hop neighbors are at most τ seconds apart from each other. The duration of a contention step varies according to the type of the collision-resolution step. It is possible for a channel to have only receiving periods. This is the case if the station owner of the channel does not have data packets to send. Such a station remains in the receiving state initiate new CRI, as is the case for the station in channel 2 (as shown in Fig. 1).
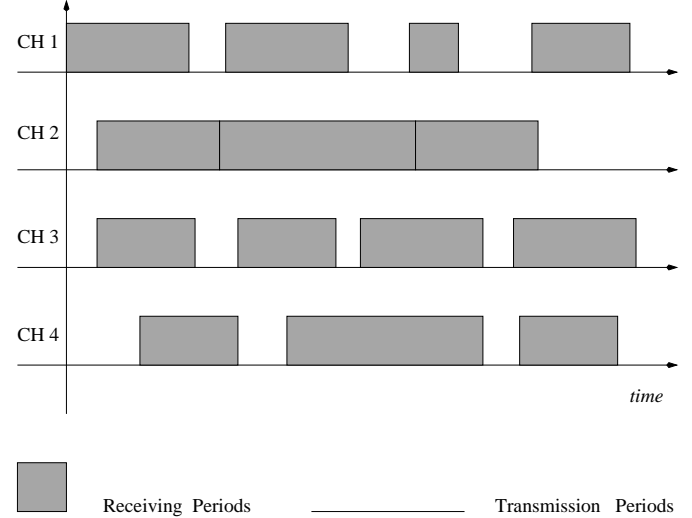

Fig. 1. Each channel is composed of receiving periods and transmission periods.

If a station has no local packet pending, then the station can initiate the receiving period by transmitting an RTR on the channel assigned to it. The station becomes a receiver station for its one-hop neighbors by transmitting an RTR at the beginning of each contention-resolution step, i.e., the station makes a transition to the receiver state. Each RTR can be visualize as a small packet indicating to other stations that the station transmitting the RTR is ready to receive a request for the floor. The RTR also contains information regarding the allowed ID interval of stations that compete to acquire the floor. Since only the receiver station sends an RTR in its corresponding channel, collisions of RTR can never occur. The receiver state is the default state for all active stations.

A station in the receiver state with a local packet pending makes a transition to the sender state, if the current CRI is completed, i.e., the station must resolve all the contentions for the current CRI in its channel. Only then, it scans the destination channel for at most the maximum duration of a CRI. If an RTR is heard during this interval and the stations ID is within the allowed ID interval, then the station competes to acquire the floor extending its duration in the intended receivers channel until it acquires the floor.

A station acquires the floor by sending an RTS. The station follows a non-persistent CSMA strategy for the transmission of RTSs. More precisely, the RTS is directed to the receiver in the channel where the RTR was sensed. The sender of an RTS waits and listens for one maximum round-trip time, plus the time needed for the destination's CTS to arrive. If the CTS is not corrupted and is received within the time limit, then the station acquires the floor and transmits its data packet or train of data packets. If the CTS is not received, all stations monitoring the channel detect a collision. Because the handshake occurs on the receiver's channel, a lost CTS on one of the contending stations due to errors or fading, leads to a loss of feedback but not inconsistent feedback. Notice that a station wishing to acquire the floor spends at most the equivalent of two maximum CRI durations. The maximum CRI duration is determined by the maximum number of one-hop neighbors allowed. The rule

is that if within the first maximum CRI duration the station does not hear a valid RTR, the station transitions back to the receiver state and remains in this state until the end of a CRI.

Although each station transmits an RTS only after the RTR is received and only if the station is within the allowed ID interval, collisions of RTSs may still occur due to propagation delays. RTSs are vulnerable to collisions for time periods equal to the propagation delays between the senders of RTSs. CARMA-MC uses a deterministic tree-splitting algorithm [8] to resolve collisions of RTSs, which are detected by the absence of a CTS. Each step of the CRI is initiated by an RTR. There are three types of steps, idle, collision, and success. An idle steps has a duration of $\rho + 2\tau$, a collision step has a duration of $\rho + \gamma + 3\tau$, and a success step has a duration of $\rho + 2\gamma + \delta + 4\tau$, where ρ is the size of an RTR, γ is the size of an RTS or a CTS, δ is the size of a data packet, and τ is the maximum channel delay.

D. Collision Resolution Interval

CARMA-MC uses a deterministic tree-splitting algorithm to resolve collisions. This algorithm was presented and analyzed in detail in [8].

Whenever a station has a local packet to send and is not currently the receiver in a CRI, the following operations are executed. First, the station scans the channel of its intended receiver for an RTR. RTRs inform contenders that a resolution step can take place, allowing multiple contenders to respond. If the RTR is detected, the station sends an RTS and waits for one maximum round-trip time plus the time needed for the destination to send a CTS. If the CTS is received within the allocated time, then the sender acquires the floor and can start sending collision-free packets. On the other hand, if the CTS is not received within the allocated time, then the station sender of the RTS and all the other stations competing for the floor detect a collision. In this case, the collision-resolution algorithm is started with the first round of RTS collisions. As soon as a collision is detected, the receiver station divides the allowed ID interval $(LowID, HiID)$ into two ID intervals. The first ID interval, called the backoff interval, is $(LowID, \lceil \frac{HiID+LowID}{2} \rceil - 1)$, while the second ID interval, which is the new allowed ID interval, is $(\lceil \frac{HiID+LowID}{2} \rceil, HiID)$. The receiver updates its stack by executing a PUSH stack command, where the key being pushed is the backoff interval. After this is done, the station updates $LowID$ and $HiID$ with the values from the new allowed ID interval broadcasting this information on the next RTR. This marks the beginning of the next collision-resolution step. This operation is repeated each time there is a collision.

Recall that each step of the collision-resolution algorithm, and consequently each contention step, can be either:

• Case 1–*Idle*: There is no station in the RTS state whose ID is within the allowed ID interval. The channel remains idle for the duration of one propagation delay. The stack and the variables $LowID$ and $HiID$ are updated. The receiver station executes a POP command in the stack updating the allowed ID interval with the new values.

• Case 2–*Success*: There is a single station with an RTS to send whose ID lies within the allowed ID interval. In this case, a single station is able to complete an RTS/CTS handshake successfully, acquiring the floor and transmitting its data packet.

• Case 3–*Collision*: There are two or more stations with RTSs to send whose IDs are within the allowed ID interval. In this case, each of these stations sends an RTS creating a collision. The stations in the allowed ID interval are again split into two new ID intervals and the stack and the variables for the receiver station are updated.

Fig. 2 shows two stations contending for the floor. The receiver initiates the CRI. Each CRI is composed of a sequence of collision-resolution steps, each initiated by an RTR.

III. CHANNEL UTILIZATION AND PACKET DELAY

In this section we derive a lower bound on the average utilization of the channel as well as an upper bound on the average delay that a station experiences in transmitting a data packet. Recall that, the channel utilization is defined as the amount of time that the channel is used to transmit data packets divided by the total time.

In order to simplify our analysis, we assume that each station has at most d_{max} one-hop neighbors and that each station has a different receiving frequency or channel than its two-hop neighbors. Furthermore, we assume that all the channels have similar behaviors and this enable us to study the utilization of a generic channel. Observe that, if $d_{max} > 2$ then the number of distinct receiving channels is $n = d_{max}^2 - d_{max} + 2$.

Let ch_r denote the receiving channel, f_r the frequency associated with channel ch_r and n_r the receiving station. Then, station n_r is the intended receiver for at most d_{max} one-hop neighbors. At the same time, any of the d_{max} one-hop neighbors is a potential receiver for station n_r. We also let n_x denote the intended receiver of n_r regardless of which of the d_{max} one-hop neighbors station n_r chooses to send its data packet. Thus, f_x is the transmitting frequency of station n_r and ch_x is the channel associated with frequency f_x.

Finally, we define $T(n, d_{max})$ as the average size of the maximum duration of a CRI for a receiver with at most d_{max} one-hop neighbors and n unique channels within the two-hop neighborhood of the network node and at the same time the maximum number of nodes in the deterministic tree. In order to derive $T(n, d_{max})$, we make use of the analytical results obtained in [8] for the number of collision-resolution steps.

For all $n \geq d_{max} > 1$, where n is the maximum number of stations and d_{max} is the maximum number of stations participating in the CRI, the average number of collision steps required to resolve all d_{max} collisions is

$$
\begin{aligned}
C(n, d_{max}) &= 1 + \sum_{i=\mu}^{\nu} \frac{\binom{\alpha}{d_{max}-i}\binom{\beta}{i}}{\binom{n}{d_{max}}} C(\alpha, d_{max} - i) \\
&+ \sum_{i=\mu}^{\nu} \frac{\binom{\alpha}{d_{max}-i}\binom{\beta}{i}}{\binom{n}{d_{max}}} C(\beta, i)
\end{aligned}
\tag{1}
$$

while the average number of idle steps required to resolve all d_{max} collisions is

$$
Z(n, d_{max}) = \sum_{i=\mu}^{\nu} \frac{\binom{\alpha}{d_{max}-i}\binom{\beta}{i}}{\binom{n}{d_{max}}} Z(\alpha, d_{max} - i)
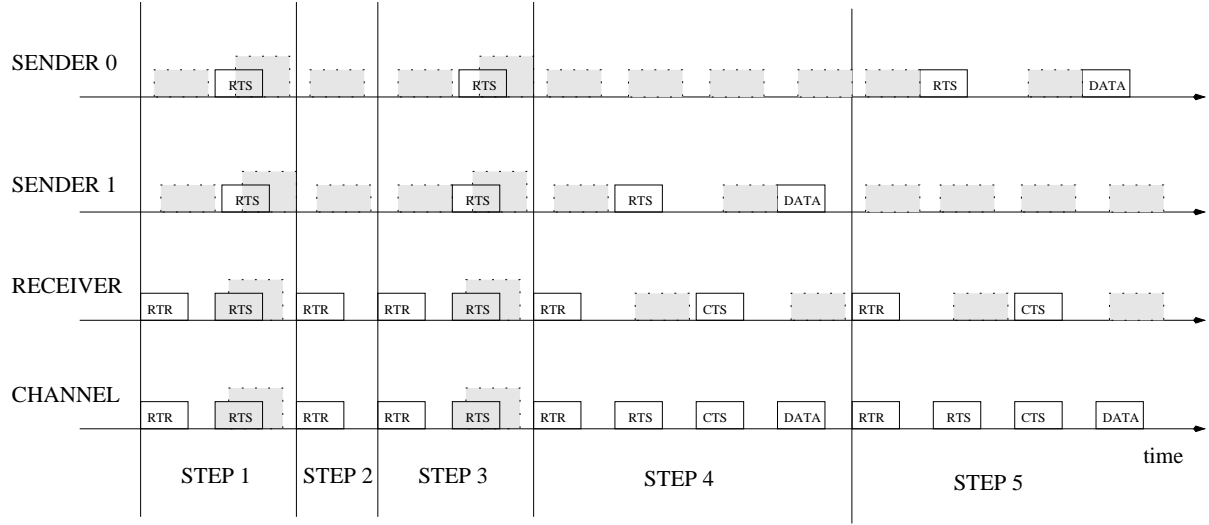$$

Fig. 2. Receiving mode: The intended receiver sends an RTR to initiate the CRI. Notice that each collision-resolution step is initiated by an RTR packet.

$$+ \sum_{i=\mu}^{\nu} \frac{\binom{\alpha}{d_{max}-i}\binom{\beta}{i}}{\binom{n}{d_{max}}} Z(\beta, i) \qquad (2)$$

where

$$\alpha = \lceil n/2 \rceil; \ \beta = n - \alpha = n - \lceil n/2 \rceil$$

$$\mu = \begin{cases} 0 & \text{if } d_{max} \leq \alpha \\ d_{max} - \alpha & \text{if } d_{max} > \alpha \end{cases}$$

$$\nu = \begin{cases} d_{max} & \text{if } d_{max} \leq \beta \\ \beta & \text{if } d_{max} > \beta \end{cases}$$

The number of success steps is $S(n, d_{max}) = d_{max}$. $T(n, d_{max})$ can be calculated by multiplying the number of steps by the duration of a step for each of the three types of collision-resolution steps. An idle steps has a duration of $\rho + 2\tau$, a collision step has a duration of $\rho + \gamma + 3\tau$, and a success step has a duration of $\rho + 2\gamma + \delta + 4\tau$, where ρ is the size of an RTR, γ is the size of an RTS or a CTS, δ is the size of a data packet, and τ is the maximum channel delay. Therefore,

$$\begin{aligned} T(n, d_{max}) &= S(n, d_{max})(\rho + 2\gamma + \delta + 4\tau) \\ &+ Z(n, d_{max}m)(\rho + 2\tau) \\ &+ C(n, d_{max})(\rho + \gamma + 3\tau) \end{aligned} \qquad (3)$$

Recall that a channel is composed of receiving intervals followed by transmitting intervals. Once a station is in the receiving mode (i.e., is a receiver) it remains in that mode until the end of the current CRI. Only then, the station can switch to the transmitting mode if its transmitting queue contains data packets that need to be sent. We further assume that when a station acquires the floor it can only transmit one data packet.

We start by computing in Theorem 1 an upper bound on the average delay that a station experiences in transmitting a packet. According to the CARMA-MC protocol, a station trying to transmit a packet can wait at most $T(n, d_{max})$ time for an RTR. If during this maximum period of time it does not hear an RTR, it must return to its channel and initiate a CRI. On the
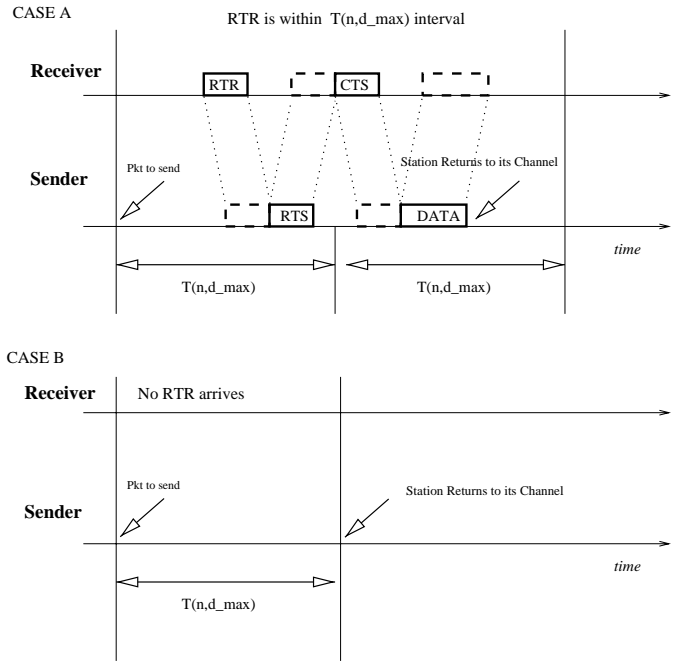

Fig. 3. Transmission period for CARMA-MC. In case A, the RTR arrives within the the time interval $T(n, d_{max})$, therefore, the sender contents for the floor. In case B, the RTR does not arrived within the allowable interval, therefore, the sender must return to its channel and transition to the receiving mode.

other hand, if it hears an RTR it will transmit its packet within an additional $T(n, d_{max})$ interval. This implies that the duration of the longest possible transmission period is $2T(n, d_{max})$. Thus, for our protocol the best possible receiver is the one that spends all its time in the receiving mode, i.e., its transmission queue is always empty, and a station requesting the floor in the channel of such receiver will always succeed in the request within the time interval $T(n, d_{max})$. On the other hand, the worst possible receiver is the one that spends most of its time in the transmitting mode and the least amount of time in the receiving mode. That is, the worst possible receiver will constantly have

idle steps (i.e., non of its neighbors will request the floor) and receive a data packet in its transmission queue when found in the receiver state. This will force the receiver to switch to the sender state and then to remain in this state for as long as possible. Now, since the duration of an idle step is $\rho + 2\tau$ where ρ is the size of the RTR packet and τ is the maximum channels delay, and the duration of the longest possible transmission period is $2T(n, d_{max})$, each time a sender targets the worst possible receiver its probability of success is

$$P_s = \frac{T(n, d_{max})}{2T(n, d_{max}) + \rho + 2\tau} < 0.5 \qquad (4)$$

and its probability of failure is

$$P_f = \frac{T(n, d_{max}) + \rho + 2\tau}{2T(n, d_{max}) + \rho + 2\tau} > 0.5 \qquad (5)$$

In general we can approximate this event by assuming an independent Bernoulli trial with probability q. Each time the station tries to send a data packet, the packet will be successful with probability q. This can be approximated by a geometric distribution function whose expected value is $E[Y] = \frac{1-q}{q}$. Recall that a geometric random variable counts the number of failures before a success occurs. Thus, for the worst possible receiver we have $q = 1/2$ and $E[Y] = 1$. We are now ready to prove the following theorem.

Theorem 1: Consider a system where each station has at most $d_{max} > 1$ one-hop neighbors. Then the average channel delay experienced by a station trying to deliver a data packet can be upper bounded by

$$D \leq 5T(n, d_{max}) \qquad (6)$$

where $T(n, d_{max})$ is the maximum possible duration of a CRI.
Proof: To derive Eq. (6) we assume that for our station n_r the intended receiver is always the worst possible receiver that we can think of. Thus, each time the station n_r tries to transmit a packet it fails on average half of the time. According to our model this is true if within $T(n, d_{max})$ the station does not hear an RTR. After a failed transmission the station must return to its channel and switch from the transmitting state to the receiving state initiating a new CRI as a receiver. After the CRI ends the station transitions back to the transmitting state and tries once more to acquire the floor on the channel assigned to its intended receiver.

Now, since each failed transmission has a total cost of $T(n, d_{max})$ (the time waiting for an RTR) plus $E[R]$, the expected duration of the CRI (in which the station is a receiver), the expected duration of a CRI can be upper bounded as follows

$$E[R] \leq \sum_{k=0}^{d_{max}} p(k)T(n, k) \leq T(n, d_{max}) \qquad (7)$$

where $p(k)$ is the probability of having a CRI with k stations competing for the floor, and $T(n, k)$ is the corresponding duration to resolve all k initial collisions.

The same procedure is repeated each time the station fails to deliver its data packet and must retry after one CRI as a receiver. We can approximate the retry event by assuming an independent

Bernoulli trial with probability q. Each time the station tries to send a data packet, the packet will be successful with probability q. Therefore, it can be represented by a geometric distribution function whose expected value is

$$E[Y] = \sum_{y=0}^{\infty} y(1-q)^y q = \frac{1-q}{q} \qquad (8)$$

which is equivalent to the expected number of failures before a success. Thus, the average time spend by station n_r transmitting a data packet can be written as

$$
\begin{aligned}
T_{xmit} &\leq \sum_{y=0}^{\infty}(1-q)^y q(y(T(n, d_{max}) + E[R]) + \\
&\quad 2T(n, d_{max}) \\
&= (T(n, d_{max}) + E[R])E[Y] + 2T(n, d_{max}) \\
&\leq 2T(n, d_{max})(1 + E[Y]) \qquad (9)
\end{aligned}
$$

Now observe that the data packet had to be originated in the last CRI in which the station was in the receiver state. If this was not the case the station would have remained in the receiver state and not switched to the sender state. Therefore, we need to add to T_{xmit} the expected duration of the last CRI (i.e., $E[R] \leq T(n, d_{max})$) before the station switched from the receiver state to the sender state. The thesis then follows from the fact that for the worst possible receiver we have $q = 0.5$ and $E[Y] = 1$. ■

In the next result we derive a lower bound on the average channel utilization.
Theorem 2: Consider a system where each station has at most $d_{max} > 1$ one-hop neighbors. Then the average channel utilization is bounded by

$$U \geq \frac{(E[X] + E[Y] + 1)E[k]\delta}{E[R](E[X] + E[Y] + 1) + T(n, d_{max})(E[Y] + 2)} \qquad (10)$$

where $E[X]$ is the number of CRI in which station n_r remains in the receiver state before it leaves its channel to transmit a data packet of its own on another channel; $E[Y]$ is the number of attempts that station n_r must make before transmitting its data packet; $E[k]$ is the expected number of data packets received by station n_r per CRI; $E[R]$ is the average duration of one CRI as a receiver; $T(n, d_{max})$ is the maximum duration of a CRI and δ is the size of one data packet.
Proof: The frequency with which station n_r tries to send a data packet is determined by its packet data rate. Once again we can approximate this procedure with a geometric distribution function. At the end of each CRI a coin with probability q is tossed to decide whether or not during the previous CRI station n_r had a packet to send and thus must switch from the receiver state to the transmitting state. Hence, the expected number of CRIs occurring before station n_r has a packet to send is

$$E[X] = \frac{1-q}{q} \qquad (11)$$

Now, let k be the number of packets received during a CRI, $p(k)$ be the probability of having a CRI with k initial collisions

and $T(n,k)$ be the time needed to resolve all k collisions. It is not difficult to see that the expected duration of a CRI can be upper bounded by

$$E[R] \leq \sum_{k=0}^{d_{max}} p(k)T(n,k) \qquad (12)$$

and that the expected number of packets received in a CRI is

$$E[k] = \sum_{k=0}^{d_{max}} p(k)k. \qquad (13)$$

Since each station has at most d_{max} one-hop neighbors, we have that $k \leq d_{max}$. Observe that the total duration T_{total} in a channel is composed of receiving periods and transmitting periods that appear as idle periods in the channel. Furthermore, since the number of receiving periods before station n_r has a data packet to transmit on another channel is determined by the expected value for a geometric random variable, i.e., $E[X]$, it follows that the expected arrival time for a packet is $E[R]E[X]$. Once the packet has arrived, station n_r finishes as receiver the current CRI adding one more $E[R]$ period to the total duration. Then it switches to the sender state.

Now, station n_r makes several trials before it is able to deliver its data packet. As explained in Theorem 1, the trials are also governed by a geometric distribution. Each failure has a cost of $T(n,d_{max})$ plus $E[R]$, and the number of failures before the data packet can be delivered successfully is $E[Y]$. Once station n_r hears an RTR in the channel of the intended receiver, it has at most $2T(n,d_{max})$ to content successfully and deliver its packet. Therefore, the total period in the channel can be written as

$$\begin{aligned} T_{total} \quad \leq \quad & E[X]E[R] + E[R] + \\ & (T(n,d_{max}) + E[R])E[Y] + 2T(n,d_{max}) \end{aligned}$$
$$(14)$$

where $E[X]$ is the number of CRI in between two transmissions, $E[R]$ is the expected duration of each CRI and $E[k]$ is the expected number of data packets transmitted in channel ch_r per CRI. On average the amount of time that channel ch_r is being used to transmit data packets is

$$(E[X] + E[Y] + 1)E[k]\delta \qquad (15)$$

The thesis then immediately follows by dividing Eq. (15) by Eq. (14). ∎

IV. Simulation

In order to verify the analytical results obtained in the previous section, simulations with different loads were performed. For the experiments we used a total of 100 stations each surrounded by at most $d_{max} = 4$ neighbors. Therefore, the maximum number of channels was set to $n = d_{max}^2 - d_{max} + 2 = 14$. Packets were created at each station using independent Poisson generators. The simulations were repeated a number of times running on average one hour per trial. We assumed a high-speed network ($1\,Mbps$) in which Ethernet data packets (512 bytes) are transmitted. Furthermore, we have used the same distance

between stations and defined the diameter of the network to be 1 mile. Under these assumptions, the propagation delay of the channels is $5.4\mu s$. In order to accommodate the use of IP addresses for destination and source, the minimum size of RTSs and CTSs was chosen to be 20 bytes. The RTR control packet was assumed to be 10 bytes. The maximum expected theoretical packet delay was $5T(n, d_{max}) = 5T(14,4) \leq 98ms$.
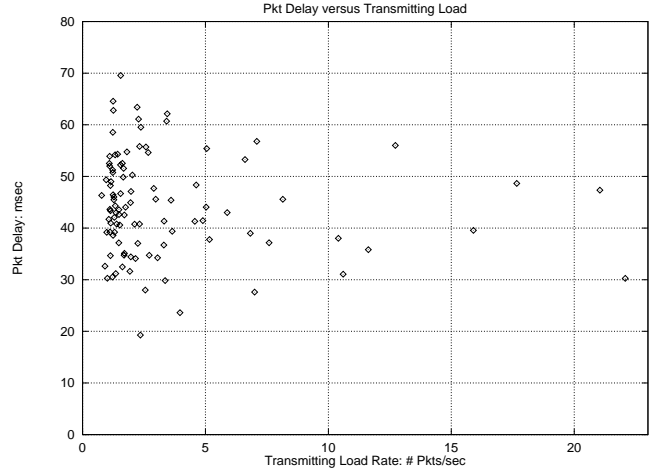

Fig. 4. Channel delay for CARMA-MC

Figure 4 shows packet delay as a function of the number of data packets being transmitted per second on a channel. The load applies to the receiving channel, i.e., the number of packets that the receiver receives per second. The delay is measure in msec and is the interval of time from the arrival of the packet to its queue until the packet was successfully delivered. The packet delay for each individual trial was always below the theoretical upper bound predicted by Eq. (6).

In figure 5 we have compared the utilization derived from our analysis to that obtained in the simulation as a function of the receivers load. To obtain the utilization of the channel we counted the intervals of time when data packets were transmitted on the channel divided by the total simulation time. The simulation results validate our analytical results.

V. Conclusions

CARMA-MC implements collision avoidance and resolution that works correctly in multihop wireless networks. CARMA-MC is a receiver-initiated protocol based on a deterministic tree-splitting algorithm; to operate, it requires each node to be assigned a unique identifier and a channel that must be unique within the two-hop neighborhood of the node. CARMA-MC resolves three mayor sources of packet failure. First, busy interference is resolved since the intended receiver initiates the communication and remains as a receiver until the end of the CRI. A station will never send a data packet to a receiver that is not ready to receive it. Second, co-channel interference is not an issue for CARMA-MC since no station two hops away
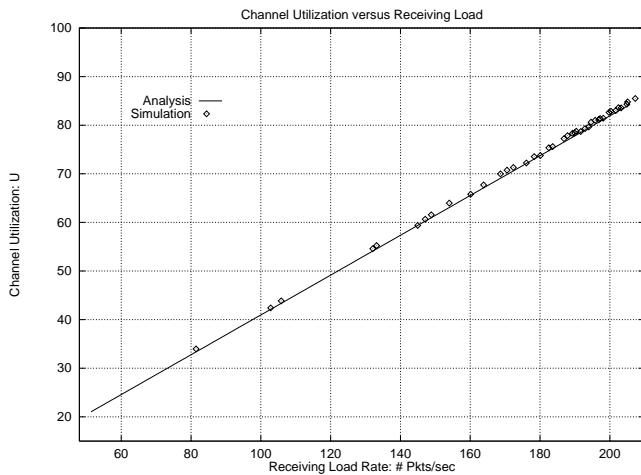
Fig. 5. Channel utilization for CARMA-MC

will have the same receiving frequency. Third, contention interference is resolved by using the deterministic tree-splitting algorithm. Our analysis shows that the average channel delay incurred in CARMA-MC is bounded and is a function of the maximum number of one-hop neighbors; our simulation validate the simplifying assumptions that we made on our analysis.

REFERENCES

[1] N. Amitay, "Resource auction multiple access (RAMA): Efficient method for fast resource assignment in decentralized wireless PCS." *Electronic Letters*, April 9, 1992, V28 N8:799-801.

[2] J. Boudenant, B. Feydel and P. Rolin, "Lynx: An IEEE802.3 compatible deterministic protocol," *Proc. IEEE INFOCOM '87*, 1987.

[3] R. L. Brewster and A. M. Glass, "Simulation model of sequential multichannel network and its throughput determination," *Electronic Letters*, vol. 25, no. 15, pp. 941-942, 1989.

[4] J.I. Capetanakis, "Tree algorithm for packet broadcasting channel," *IEEE Trans. Inform. Theory*, vol.IT-25, pp. 505-515, Sept. 1979.

[5] K. C. Chua, "Performance analysis of multichannel CSMA/CD network with noisy channel," *IEEE ICC'91*, 1991.

[6] C. L. Fullmer and J.J. Garcia-Luna-Aceves, "Floor Acquisition Multiple Access for Packet-Radio Networks," *Proc. ACM SIGCOMM 95*, Cambridge, MA, August 30-September 1, 1995.

[7] R. G. Gallager, "Conflict resolution in random access broadcast networks," *Proc. AFOSR Workshop Commun. Theory Appl.*, Provincetown, MA. Sept. 1978.

[8] R. Garces and J.J. Garcia-Luna-Aceves, "Collision Avoidance and Resolution Multiple Access (CARMA)," *Cluster Computing*, vol. 1, no. 2, pp. 197–212, 1998.

[9] J.J. Garcia-Luna-Aceves and J. Raju, "Distributed Assignment of Codes for Multihop Packet-Radio Networks," *Proc. IEEE MILCOM'97*, Monterey, California, November 2–5, 1997.

[10] D.J. Goodman, R.A. Valenzuela, K.T. Gayliard and B. Ramamurthy, "Packet Reservation Multiple Access for Local Wireless Communications" *IEEE Transactions on Communications*, August, 1989.

[11] L. Hu, "Local throughput performance of packet radio networks with transmitting power control," *IEEE ICC'91*, 1991.

[12] Dong Geun Jeong, Chong-Ho Choi and Wha Sook Jeon "Design and Performance Evaluation of a New Medium Access Control Protocol for Local Wireless Data Communications" *IEEE/ACM Transactions on Networking*, December 1995.

[13] P. Karn, "MACA - a new channel access method for packet radio," *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, pp. 134–40, ARRL, 1990.

[14] M. J . Karol and I. Chih-Lin, "A protocol for fast resource assignment in wireless PCS," *IEEE Transactions on Vehicular Technology*, vol. 43, no. 3, pp. 727-32, IEEE, 1994.

[15] L. Kleinrock and F. A. Tobagi, "Packet switching in radio channels: Part I - carrier sense multiple-access modes and their throughput-delay characteristics," *IEEE Trans. Commun.*, vol. COM-23, no. 12, pp. 1400–1416, 1975.

[16] V. C. M. Leung, "Multichannel reservation protocol for packet multiple-access communications," *Electronic Letters*, vol. 26, no. 20, pp. 1637-1638, 1990.

[17] C. G. Lof, "Packet delay for CSMA and Multi-channel ALOHA multicast schemes in WLANs with fading and co-channel interference," *IEEE Int. Conference on Universal Personal Communications*, 1996.

[18] A. M. Marsan and D. Roffinella, "Multichannel local area network protocols," *IEEE JSAC*, vol. SAC-1, no. 5, pp. 885-897, 1983.

[19] A. M. Marsan and F. Neri, "A simulation study of delay in multichannel CSMA/CD protocols," *IEEE Trans. Commun.*, vol. COM-39, no. 11, 1991.

[20] J. L. Massey, "Collision resolution algorithm and random access communications," in *Multiuser Communication Systems, G. Longo Ed.*. New York: Spriger-Verlag, 1981. pp. 73-137.

[21] N. Shacham and P. King, "Architecture and performance of multichannel multihop packet radio network," *IEEE JSAC*, vol. SAC-5, no. 6, pp. 1013-1025, 1987.

[22] B.S. Tsybakov, and N.B. Likhanov, "Upper bound on the capacity of random multiple access system," *Problems of Information Transmission*, vol. 23, no. 3, pp. 224-236, July-Sept. 1987.