An Energy Measurement Based Collision Resolution Protocol

Sanjeev Khanna, Saswati Sarkar, Insik Shin

Abstract—

A central problem in wireless packet networks is the efficient resolution of collisions among nodes with independent transmission schedules. Natural metrics for measuring the efficiency of any collision resolution scheme include total time taken for collision resolution, average participation time of each transmitter during contention resolution, average number of transmissions made by a transmitter, and average power consumption per a transmitter. While it is straightforward to design a centralized protocol that optimizes along each of these metrics, the challenge is in designing distributed protocols that are efficient along each one of these metrics. No existing approach performs well along all these metrics simultaneously. In this paper, we present EMCRR, a novel randomized distributed collision resolution protocol that is based on power control and energy measurement techniques. We study the performance of EMCRR both analytically and empirically. Our analytical and empirical results closely match each other and show that EMCRR compares very favorably to all existing schemes.

I. Introduction

WIRELESS nodes exchange information through low bandwidth multiple access radio spectrum. Users in transmission range of each other can not transmit messages simultaneously. At the same time users not in the same neighborhood can use the transmission medium simultaneously. A key challenge in wireless systems is to resolve this space dependent contention efficiently. Space dependent contention can be mitigated if users have locally unique transmission channels. For this purpose, distinct frequencies(codes) are assigned to contending users for FDMA(CDMA) based multiple access scheme. However, use of multiple channels can not eliminate the space dependent contention. This is because wireless nodes have single radios and one radio can receive one message or transmit one message, but can not perform multiple receptions and transmissions simultaneously. So, if multiple messages reach a node at the same time, all of them are wiped out and also a message is rejected if it reaches the node, while the node is in the transmission mode. Both events are termed "collision." A user often has no knowledge of its neighbors transmission schedules, and decides its own transmission schedule independently. As a result, collisions are inevitable in any packet-radio network with distributed control. All wireless transmission protocols need to recover from such collisions. The objective of this paper is to provide a new collision resolution approach using power control and energy measurement techniques.

S. Khanna, Dept. of Computer & Information Science, University of Pennsylvania, PA, USA. E-mail: sanjeev@cis.upenn.edu .

S. Sarkar, Dept. of Electrical Engineering, University of Pennsylvania, PA, USA. E-mail: swati@ee.upenn.edu .

I. Shin, Dept. of Computer & Information Science, University of Pennsylvania, PA, USA. E-mail: ishin@cis.upenn.edu .

A. Performance Metrics

We first identify the desirable properties of a collision resolution protocol in context of the current wireless network requirements. Clearly, for any practical implementation, a collision resolution protocol needs to be inherently distributed, i.e., nodes take contention decisions without any centralized coordination. Next, a good collision resolution protocol should resolve contentions within a short time, so that significant bandwidth is not wasted in collision resolution. Short collision resolution intervals also reduce power consumption of participating nodes. This is important as wireless nodes often operate on limited power sources. So a natural performance metric for a collision resolution approach is the *expected total time consumed in the collision resolution process.* Another important evaluation metric would be the *average participation time of a transmitter.* If a number of nodes contend for transmission to a single receiver, then the receiver needs to be involved in the entire resolution process. However, the participation time of the transmitting nodes depend on the particular resolution approach. The participation time of a node is the total time a node needs to transmit, receive or monitor the channel for a possible reception. Individual transmitters may be active in only a small fraction of the total contention time. In the remaining time, a particular transmitter can either remain idle and save power, or participate in other transmissions in another channel. Thus the participation time of a transmitter determines its bandwidth and power consumption in the resolution process. Another metric is the *average number of transmissions per transmitter.* Transmissions consume the maximum energy and thus it is important to reduce the actual number of transmissions in the process. The last metric is *average power consumption per transmitter.* The better it is, the less power a transmitter consumes for its collision resolution. From the receiver's viewpoint, it is important to reduce the total resolution time, whereas for the transmitters, it is more useful to decrease the individual transmitter participation times, the average number of transmissions, and the average power consumption. Thus, all of these quantities are important performance metrics of a collision resolution protocol. Also, relative performance of protocols can be significantly different for different metrics, and hence it is important to evaluate the performance of a protocol w.r.t. all of these metrics. In addition, collision resolution approaches should be simple as wireless nodes typically have limited memory and processor power as well.

B. Existing Collision Resolution Approaches

We first briefly review the existing collision resolution schemes. Several wireless medium access protocols including IEEE 802.11 use a random backoff based collision resolution[10]. Whenever there is a collision, a node backs off for a random amount of time, and then re-attempts once again, and the backoff interval increases exponentially with each reattempt. Tree based collision resolution approach has been suggested in [2], [3], [9]. Whenever there is a collision, all contending nodes randomly split into two subsets. The different subsets attempt transmission in different slots. Further splitting occurs if there are further collisions. A deterministic tree splitting approach has been proposed in CARMA-MC of [7]. Here, the splitting occurs in accordance with ID announcements from the receiver. In the collision multiplicity based splitting of CRAI in [6], the receiver estimates the number of colliding nodes from the received energy, and broadcasts this to the transmitters. The transmitters decide the splitting probability based on the collision multiplicity. We will describe these approaches in detail later, and show that the new approach we propose is substantially better than all the existing ones with respect to all the four metrics.

C. Our Collision Resolution Approach

We propose a new collision resolution approach which we call EMCRR (Energy Measurement based Collision Resolution in Rounds). The basic intuition behind our approach can be outlined as follows. Collision resolution takes place in rounds. Whenever there is a collision, a receiver node can estimate the number of colliding nodes from the received power[1]. It broadcasts this estimate back to the colliding nodes. The first collision resolution round length is determined as a function of this estimate. The colliding nodes choose their re-attempt slots randomly in the resolution round. At the end of the first reattempt round, the receiving node broadcasts the slot numbers which had successful re-attempts and the total number of remaining contenders. A colliding node can find out whether its attempt was successful or not from this information. A node that fails in the current round reattempts in the next round. This repeats till each node succeeds.

[1]The total received power can be measured using Standard circuits[12]

D. Performance Comparison

EMCRR attains good performance w.r.t. all four metrics. We prove that the average total collision resolution interval of EMCRR is upper-bounded by $2.78k$ slots, where k is the total number of contenders. So far, the best average total collision resolution time has been attained by the CRAI protocol of [6] and the number is $3.74k$ for k nodes. Simulation results indicate that the collision resolution intervals of [2], [3], [7], [9], [10] exceed that of EMCRR by at least a factor of 1.7 for broad range values of k, and the factor increases with increase in k. We prove that the maximum participation time of a transmitter in EMCRR is upper bounded by $4.36\ln k + 0.075k$ with a high probability. Analytical results are not available for the other approaches. Simulation results indicate that there is an improvement by a factor of 1.5 in the average participation time as compared to all the known approaches[2], [3], [6], [7], [9], [10]. This improvement is evident even for a small value of the number of contenders ($k \geq 4$) and the factor increases with an increase in the number of contenders. EMCRR has fewer average number of transmissions as compared to others for a large number of contenders ($k \geq 64$). The difference increases with the increase in the number of contenders. We also prove analytically that the average number of transmissions of EMCRR is upper bounded by $e + 1$ independent of the number of transmitters, whereas for all other approaches, this number increases with the increase in the number of contenders. We observe through simulations that power consumption is at least 2 times more for all the other approaches as long as the number of contenders is greater than 8 ($k \geq 8$). Power consumption depends on the average participation times, the average number of transmissions and actual values of power consumed in different activities like transmission, reception, channel monitoring and sleeping. The improvement in power consumption is mainly due to the significant reduction in the average participation time.

EMCRR attains substantially shorter collision resolution interval as compared to [2], [3], [7], [9], [10] because nodes can decide their reattempt intervals depending on the actual number of contenders. Initially, the reattempt intervals are chosen from a wide range since the number of contenders is large. This reduces collisions even if there are a large number of contenders. As the collision resolution process continues, more and more contentions are resolved, and the choice range becomes smaller, thus decreasing the total time. The policy in [6] uses the number of contending nodes as well. Even then, the collision resolution interval of EMCRR is lower than that of [6] because of reduced feedback requirements. A significant advantage of EMCRR is that a contending node needs to wake up only twice during each collision resolution round. The individual collision resolution rounds are large, and there are a few rounds in all. In contrast, the previous approaches[2], [3], [6], [7], [9], [10] require a node to monitor the channel all along, till its contention is resolved. Thus a transmitter node devotes much more time in the collision resolution process and the average participation time is high in these earlier approaches. EMCRR has lower average number of transmissions because a transmitter transmits once every round till its contention is resolved (transmitters do not transmit after contention resolution), and the number of rounds is small. We demonstrate these analytically as well as via simulations. Finally, this contention resolution scheme is simple and very general, and may be used in conjunction with any higher layer protocol. We design a transmission protocol using this contention resolution approach.

We have assumed limited multi-user detection and power control capabilities. However, the multi-user detection scheme is just used in estimating the total power received. Also, the power control is required to modulate the transmission power. Most existing receivers have both of these capabilities, and these have been widely used in interference control and performance improvement. Similar assumptions have been made in [6].

Varing Channel Characteristics: In general, the received power is a random variable whose value dependes on the transmitted power, the propagation distance, and the transmission charateristics of wireless channel (the randomness is due to the last factor)[13]. Our collision resolution protocol is robust under varying channel conditions. Analytical bounds similar to those presented in this paper can be shown in this general setting. Due to space constraints, we defer the details to the full version[11].

The rest of the paper is organized as follows. Section II describes our network model, assumptions and the proposed protocol and discusses some salient features of our policy. Section III presents our analytical results. Section IV describes existing collision resolution approaches, and compares our policy with these. We conclude with some directions for future work in Section V.

II. THE EMCRR COLLISION RESOLUTION PROTOCOL

A. Assumptions

We consider a multichannel wireless network where the channel is a frequency(code) for a frequency division multiple access (code division multiple access). We assume that every potential receiver[2] is assigned a locally unique receiving channel and that there is a mechanism ensuring that no wireless node is assigned the same channel as any other node in its two-hop neighborhood. This channel assignment can be achieved using a distributed algorithm proposed in [8]. We also assume that each node is aware of the receiving channels for all its one-hop neighbors. Each node transmits packets to its intended receiver in the receiving channel of the receiver. This mechanism can ensure that only flows sharing the same receiver interfere with each other[7]. Multi-channel wireless networks have been studied in [4], [7].

The channel access falls into two modes: (a)transmission mode and (b)reception mode. While a node tunes in its receiving channel in its reception mode, it tunes in the receiving channel of its intended receiver in its transmission mode. Basically, when a node has no packet to transmit, it is in the reception mode. when a node has packets to transmit, it may be in either mode depending on its mode selection policy. The policy may be based on the communication history, the bandwidth allocation policy, the traffic availability, and etc.

We consider a slotted system with synchronized slot boundaries. This can be attained as the receiver sends synchronization information initiating a new communication with a control message (i.e., an RTR (Ready-To-Receive) message). The transmitters synchronize their clocks with the receiver on receiving this information. Similar synchronization is actually performed in many commercial wireless adhoc standards, including bluetooth[1]. Also, the inherent assumption is that the propagation delays are negligible as compared to packet transmission times. Most often this is the case for single hop wireless links as the propagation delay for the single hop is not significant (unless we assume satellite communication) and the wireless links have low bandwidth. Also, if the collision resolution process takes long time the transmitter clocks may loose the synchronization with the receivers. To prevent this, the receiver can periodically send synchronization information in the control messages during the collision resolution process (i.e., feedback messages). We also assume that wireless links are symmetric.

We assume that a node can measure the signal strength of a packet when it receives the packet. We also assume that each node can adjust its transmission power to a certain degree of strength. Signal strength decays as d^{-r}, where d is the distance traversed and r is normally close to 4. We assume that a good estimate of the path loss index r is known at the receiver. When a transmitter transmits a packet to a receiver with signal strength P_s, the receiver would receive the packet with signal strength P_r, where $P_r = P_s \cdot d^{-r}$ and d is the distance between the trans-

[2]In general, all nodes may not be potential receivers. Consider a sensor network, where sensors communicate their measurements to some predetermined nodes. In this case the sensors do not serve as receivers, and hence do not need designated channels. However, if all nodes are potential receivers, then all nodes need locally unique channels.

mitter and the receiver. Hence, if P_s and r are known to a receiver, the receiver can estimate the distance between the two. When a transmitter knows d and wants its receiver to receive a packet with the signal strength P_r, the transmitter can transmit with the signal strength P_s, where $P_s = P_r/d^{-r}$.

When multiple transmitters transmit to the same receiver simultaneously, all the messages collide at the receiver. We assume that the receiver can measure the total received power on the reception of the corrupt messages. If all the contenders transmit their packets with power scaling such that each packet reaches the receiver with the same signal strength P_r, which is the sum of the powers of all the transmissions. The receiver estimates the number of contenders from the total received power. If k transmitters transmit in all where the received power is kP_r, the receiver can figure out that k transmitters transmitted.

The distance between a transmitter and the receiver may change during the collision resolution process on account of mobility. But, the transmitters can continuously update this estimate from the strength of the control packets (i.e., feedback packets) at the end of each collision resolution round. However, if the distance changes during a collision resolution round, then the power scaling for the RTS in the current round may be affected, and that may affect the estimation of the number of collisions. But, the power scaling will have the right value from the next round onwards.

B. *Protocol Operation Overview*

We first provide an outline of the protocol operation. The protocol has a 5 phase handshake. The phases are (a) a receiver broadcasts an RTR (Ready-To-Receive) control message, (b) a transmitter sends back to the receiver an RTS (Ready-To-Send), (c) the receiver responds to the transmitter by sending a CTS (Clear-To-Send) if the packet is received correctly, otherwise collision resolution is used, (d) the transmitter sends a DATA packet after the resolution of collision, (e) and the receiver finishes the communication by sending an ACK. The overall layout is similar to IEEE 802.11. The contribution of this paper is to provide a novel collision resolution approach. To participate in this communication, each transmitter needs to tune in its receiver's receiving channel and wait for an RTR broadcasted by the receiver. Upon hearing the RTR, each transmitter transmits an RTS. If there is one transmitter transmitting the RTS, the transmitter and the receiver proceed their communication. If there are multiple transmitters transmitting the RTS, the RTS messages collide at the receiver. Through the multi-user detection scheme based on the power control and measurement capabilities, the receiver detects the collision and estimates the number of contenders. Subsequently, the receiver uses this estimate to resolve collisions in one or multiple rounds. The collision resolution process can be described as follows:

Upon detecting the collision, the receiver broadcasts a feedback control packet. The feedback packet contains the number of contenders participating in the collision. Then, the first collision resolution round starts and takes the same number of slots as the number of contenders. If k transmitters participated in the collision, k slots are allocated for resolving this collision. Upon hearing the feedback control packet, all the contenders notice the collision of their previous RTS transmissions and transmit another RTS in a slot randomly chosen in an interval of 0 to $k - 1$.

In the first collision resolution round, some transmitters may succeed and the others may not. At the end of the round, the receiver broadcasts a feedback packet that contains successful slot numbers in the round. Since each transmitter knows in which slot it transmitted in the round, it can figure out whether its previous RTS transmission collided or not through the feedback packet. If there is one or more collisions in the round, another round starts after the feedback packet is broadcast. This repeats until all collisions are resolved. We will describe the operation in details. Also, the protocol has been illustrated in Figure 1.

C. *Receiver-initiated 5 Phase Handshake Protocol*

Our protocol consists of 5 handshakes initiated by a receiver. Among the 5 handshakes, the first three control message exchanges belong to **control message exchange phase** and the last two belong to **data exchange phase.** When a node switches to the reception mode, it broadcasts an RTR in its receiving channel starting the control message exchange phase. Since there is only one receiver in the receiving channel and only the receiver sends an RTR, there is no collision for the RTR. This control message is transmitted with a power P which is uniform throughout the system and contains the clock synchronization information. When a transmitter waiting for an RTR in the channel receives the RTR, it synchronizes its clock with the receiver and also measures the signal strength of this RTR message. Then, the transmitter sends to the receiver an RTS control message that contains the number of packets the transmitter intends to transmit. The transmission power for the RTS message is scaled such that the reception power for this packet after the path loss is P', where P' is agreed apriori throughout the system.

If there is no transmitter in the channel, the receiver does not receive any RTS and concludes that there is no interested transmitter. If there is only one transmitter that transmitted an RTS to the receiver and the RTS is received correctly, then the receiver sends a CTS in the next slot and switches to the data exchange phase. The CTS contains the number of allowed packets. This number is upper bounded by the requested number of packets in the RTS. Like the RTR, there is no collision for the CTS. If a transmitter re-

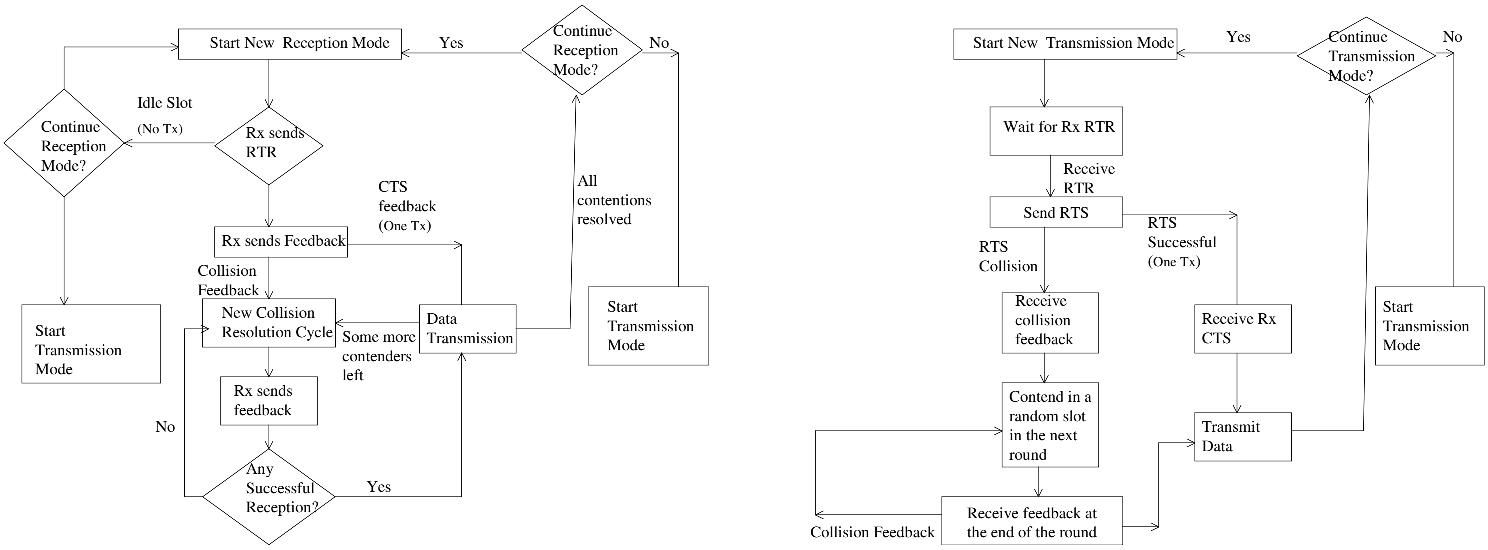

Fig. 1. A flowchart for the the Reception and the Transmission Modes. The left figure shows the Reception mode and the right one shows the Transmission mode. Here, Rx indicates the receiver and Tx indicates the transmitter.

ceives a CTS, it swiches to the data exchange phase and starts transmitting data packets. After transmitting the allowed number of packets, the transmitter either switches to the reception mode, or switches to the transmission mode for a different receiver or remains idle. The receiver either switches to the transmission mode, or continues in the reception mode. In the last case, it issues a fresh RTR starting another control message exchange phase.

If multiple transmitters wish to transmit to the receiver (which is normally the case for many applications), all the corresponding RTS collide at the receiver. The receiver can not decipher any information from the corrupt messages, but can still measure the total received power. It can estimate the number of contenders from this measurement, since on account of power scaling, the received signal strength of all the RTS are the same, and the received signal strength of any one RTS, P' is known apriori. Thus if k transmitters wish to participate, then the received power is kP'. The receiver broadcasts the number of initial contenders, k in the next slot to its one hop neighbors. This is also the length of the first collision resolution round in terms of the number of slots.

D. Collision Resolution Round

Upon detecting an RTS collision right after an RTR broadcast, the receiver broadcasts a feedback control packet that contains the number of contenders, k, participating in the collision. Like an RTR control packet, the feedback packet is transmitted with a power P and contains the clock synchronization. The first collision resolution round starts, and the contenders choose a random integer between 0 and $k-1$ for choosing a slot for retransmission. The probability of choosing any number in this range is equal, $1/k$, and the choice at any contender is independent of that of the others. If a transmitter chooses i, then it retransmits the RTS after waiting for i slots and then waits for a feedback from the receiver. The receiver broadcasts the feedback at the end of the current collision resolution round which is k slot long. During the round, some of the RTS collide, while others are successfully received. The feedback contains the number of contenders participating in the current round. It also contains a list of the slot numbers in which the receiver successfully received RTS and the number of data packets the corresponding transmitters can send. This number of allowed data packets is upper bounded by the request of the individual transmitters. For example let $k = 10$, and the receiver receives RTS successfully in slots numbered $1, 3, 7$ in this round, and the corresponding transmitters are allowed to transmit $4, 2, 3$ packets respectively. Then the feedback packet contains couples $(1, 4)$, $(3, 2)$ and $(7, 3)$. Note that the receiver need not include the identities of the successful transmitters in this packet. A transmitter knows the slot number it attempted, and hence can find out whether or not it succeeded from this feedback. If there are some successful slots, then the receiver and the corresponding transmitters switch to the data exchange phase. We will describe the data exchange phase later. A fresh collision resolution round starts after the data exchange phase until there is no collision in the current round.

E. Data Exchange Phase

The transmitters can compute the data transmission sequences from the information in the feedback packet. For

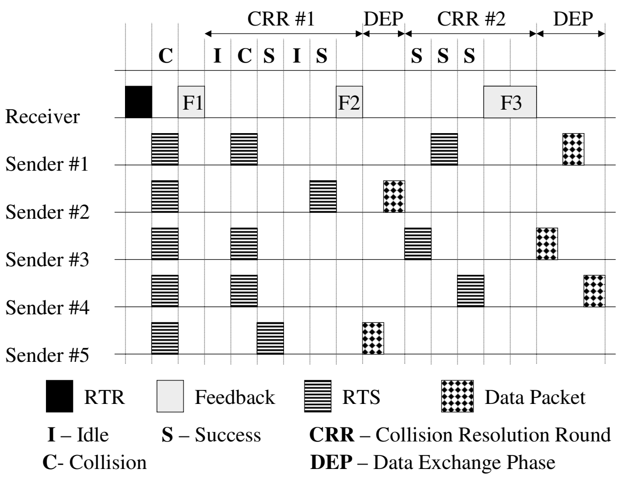

Fig. 2. The figure illustrates the operation of EMCRR with 5 transmitters contending for a single receiver. All 5 RTS collide in the slot after the RTR. feedback F1 indicates this. The first collision round consists of 5 slots. Transmitters 2 and 4 succeed in slots 3 and 5 respectively, while the other 3 collide in the second slot in this round. Slots 1 and 4 are idle. We assume that all nodes requests one packet to transmit. Thus the feedback packet for the first collision resolution round F2 contains [3, (3,1), (5,1)] indicating that there were successful receptions in the third and the fifth slots in this round and the corresponding transmitters can transmit 1 packet each. The successful contenders transmit data packets in the following data transmission phase. The second round consists of 3 slots and all 3 remaining contenders succeed. feedback F3 for this round contains [0, (1,1), (2,1), (3,1)].

example if a transmitter N_j sent RTS successfully in slot j in the current collision resolution round, and there are are i transmitters which succeeded in slots before slot j, and t_l is the number of packets successful node N_l have been permitted to transmit, then transmitter N_j transmits data packets in slots $1 + \sum_{l=1}^{i} t_l$ to $t_j + \sum_{l=1}^{i} t_l$. It knows the numbers i, $t_1, \ldots, t_i$ and t_j from the feedback packet. Note that the remaining contenders (the contenders who failed in the previous round) can also compute the length of the data transmission phase from the information in the feedback packet. If there were i successes in the previous collision resolution round, then the length of the data transmission phase is $\sum_{l=1}^{i} t_l$. Thus these contenders know when a new contention resolution round starts. Figure 2 presents an example illustration of the procedure.

F. Discussion

During a collision resolution round, a contender participates only twice when it transmits an RTS in its randomly chosen slot and when it receives a feedback packet in the last slot of the collision resolution round. Since the contender needs not participate in the other slots to follow the protocol, it can go to sleep or participate in other activities in another channel during the other slots. Thus, clearly the participation time of a transmitter is low in this process. We will illustrate this further by analysis and simulation. However, the receiver needs to be active all through, but this is natural and true with all other known collision resolution procedures.

The feedback packets at the end of every round contains information about the number of remaining contenders and the successful slots. Thus collisions do not induce additional feedback. All the tree based approaches [2], [3], [6], [7], [9] need feedback after every slot and this increases the total time consumed in the collision resolution.

The protocol is completely distributed.

Many of the existing schemes, e.g. [7] assume that every receiver knows the ID range of all potential transmitters in the neighborhood. However, each transmitter needs to know the frequency of its receiver. Our scheme is particularly suited to application where the receivers are a few and reasonably static, and the transmitters are many more and mobile. We will present such an example next. In general, network topology continuously changes on account of mobility. So the receivers need to continuously update the ranges. In EMCRR receivers do not need any such node specific information.

G. Collision Resolution in Wireless Multicast

We conclude this section with an interesting application of EMCRR in wireless multicast. All the multicast group members need to contact the group coordinator in order to join the group, and the group coordinator decides group membership. This is a specific example of scenario where there are a few static receivers (coordinators) and several more possibly mobile transmitters. Consider a multicast group for a real-time broadcast where many people join right before the broadcast begins. In this multicast group, many potential members may contact the group coordinator around the same time. Consequently there is a large number of collisions, which may be resolved efficiently using EMCRR. The group coordinator sends a RTR and initiates a reception cycle. All potential members contact the coordinator in the next slot. The membership is decided in one slot if there is just one member. Otherwise, a collision resolution round is initiated with the collision feedback which mentions the number of contenders. The feedback at the end of every collision resolution round contains the successful slot numbers and the number of remaining contenders as before, and also a variable to indicate whether membership is granted or not. A new collision resolution round immediately follows the current feedback until the collisions are resolved. As we will observe later, we obtain significant improvement in the all performance metrics for a large number of potential group members.

III. ANALYTICAL RESULTS

We obtain analytical bounds on the total collision resolution interval, the average participation time of a transmitter and the average number of transmissions in this section. Let there be k initial contenders and let Z_i, $1 \leq i \leq k$, be a random variable that indicates the number of rounds in which the ith transmitter participates before it succeeds. Note that $Z_i + 1$ is the number of transmissions of transmitter i. Let Z denote $\sum_{i=1}^{k} Z_i$. Also, let k_j denote the number of transmitters that participate in the jth round. Thus the total number of slots that are used in the collision resolution rounds is given by $\sum_j k_j = \sum_{i=1}^{k} Z_i = Z$. This follows because if m transmitters contend in a round then the collision resolution round has m slots. The maximum participation time of any transmitter depends on $Z_{\max} = \max_i Z_i$. This is because a node transmits and receives once every round it contends. We now derive bounds on the values of Z and $Z_{\max}$.

Let X_l be the random variable indicating the number of contenders at the beginning of the lth round. Define $p_j(r)$ as the probability that a transmitter succeeds in the jth round given that $X_j = r$ and given that it has failed in the previous rounds.

$$p_j(r) = \binom{r}{1}(\frac{1}{r})(1 - \frac{1}{r})^{r-1}.$$

Consider the function $f(q) = (1 - 1/q)^{q-1}$. Since $\lim_{q\to\infty} f(q) = 1/e$ and the function $f(q)$ is monotonically decreasing for all $q \geq 1$, we conclude that $f(q) \geq 1/e$, for all $q \geq 1$. It follows that $p_j(r) \geq 1/e$, for all j, r. Let $g_i(r)$ be the probability that transmitter i collides in round r given that it has collided in all previous rounds.

$$\begin{aligned} Pr(Z_i \geq r) &= Pr(\text{Transmitter } i \text{collides in the first} \\ &\quad r \text{ rounds}) \\ &= \Pi_{s=1}^{r} g_i(s) \end{aligned} \quad (1)$$

Denote the event transmitter i colliding in round r as $C_{i,r}$ and the event that transmitter i collides in the first $r-1$ rounds as C_i^{r-1}.

$$\begin{aligned} g_i(r) &= \sum_{j=1}^{k} Pr(C_{ir} \mid X_r = j, C_i^{r-1}) Pr(X_r = j \mid C_i^{r-1}) \\ &= \sum_{j=1}^{k} (1 - p_j(r))\, Pr(X_r = j \mid C_i^{r-1}) \\ &\leq (1 - 1/e) \sum_{j=1}^{k} Pr(X_r = j \mid C_i^{r-1}) \\ &= (1 - 1/e) \end{aligned} \quad (2)$$

From (1) and (2), $Pr(Z_i \geq r) \leq (1 - 1/e)^r$ (3)

$$\mathrm{E}Z_i = \sum_{r=0} Pr(Z_i \geq r) \leq \sum_{r=0} (1 - 1/e)^r = e$$

It follows that $\mathrm{E}Z \leq ke$.

Choosing $r = 4.36(\ln k)$, we conclude from (3) that $Pr[Z_i \geq 4.36(\ln k)] \leq 1/k^2$. Using union bound it follows that $Pr[Z_{\max} \geq 4.36(\ln k)] \leq 1/k$. Thus with probability at least $1 - 1/k$, each transmitter is successful within $4.36\ln k$ rounds. The results are summarized below.

Theorem 1: The expected number of slots used in the collision resolution rounds is upper bounded by ek and with probability at least $1 - 1/k$, all transmitters succeed within $4.36(\ln k)$ rounds.

Note that the feedback at the end of every round consists of successful slot numbers and the number of packets allowed for the corresponding transmitters. We allow 3 bytes for each successful slot, and we assume that every slot can accommodate around 40 bytes (this is the size of the control packets in IEEE 802.11). In all, there are exactly k successful slots. The total number of slots consumed in communicating the feedback is $3k/40$.[3] Also, the resolution process needs 3 slots before the first collision resolution round in order to receive the RTR, send the first RTS and receive the collision feedback. Thus from Theorem 1 the expected duration of the collision resolution round is upper bounded by $(e + 3/40)k + 3$ slots, which is approximately $2.78k$ (the additive constant of 3 can be ignored for reasonably large k).

Note that a transmitter sends a packet once in each round it contends, and from Theorem 1 it can contend for at most $4.36(\ln k)$ rounds with a high probability (probability greater than $1 - 1/k$). Also, a transmitter needs to receive at most $3k/40$ slots of feedback by the previous argument. Again, every transmitter is active for 3 slots before the first collision resolution round. It follows that a transmitter needs to participate in at most $3/40k + 4.36\ln k + 3$ slots with a high probability. Thus the average participation time of a transmitter is upper bounded by $3/40k + 4.36\ln k + 3 + \epsilon$ for all large k and arbitrarily small $\epsilon > 0$. Again, the additive constants can be ignored for large k.

We argued that $Z_i + 1$ is the number of transmissions of transmitter i. Thus the average number of transmissions is $1 + (\mathrm{E}Z/k)$. Since $\mathrm{E}Z \leq ek$, this number is upper bounded by $(e + 1)$ for any k.

[3] Strictly speaking, the total number of feedback slots may be little more than that on account of overheads, and since the number of feedback bytes may not be multiples of slot lengths, and we may need ceiling operations to take this into account. However, such deviations are negligible for all practical values of k.

k	2	4	8	16	32	64	128	256	512	1024	2048
T	0.79	0.82	0.91	0.95	0.98	1.00	0.99	1.00	1.00	1.01	1.00
M	0.80	0.75	0.81	0.80	0.78	0.79	0.84	0.84	0.89	0.92	0.95
A	2.32	2.63	3.04	3.32	3.51	3.62	3.64	3.69	3.70	3.72	3.71

TABLE I

WE SHOW THE RATIO OF THE EXPERIMENTAL RESULTS TO THE ANALYTICAL RESULTS FOR EMCRR. HERE, k IS THE NUMBER OF CONTENDERS, T IS THE AVERAGE TOTAL COLLISION RESOLUTION INTERVAL DIVIDED BY $2.78k + 3$, M IS THE MAXIMUM PARTICIPATION TIME DIVIDED BY $3/40k + 4.36\text{LN}k + 3$, AND A IS THE AVERAGE NUMBER OF TRANSMISSIONS.

IV. SIMULATION RESULTS AND COMPARISON WITH THE EXISTING COLLISION RESOLUTION PROTOCOLS

We present the evaluation of our policy through simulations in this section. We also compare and contrast our approach with the existing ones. We evaluate our policy with respect to the total collision resolution time, the average participation time of the transmitters, the average number of transmissions and the power consumed in the collision resolution process. We study the performance for different number of contenders (k). We assume that the transmitters start contending at the beginning of every reception cycle and continue till their collisions are successfully resolved (we do not allow a contender to join midway or drop out before its collision is resolved). Table I presents the simulation results for EMCRR. Simulation results show that the expected collision resolution interval is indeed upper bounded by $2.78k + 3$ slots, just as the analysis predicts (k is the number of contenders). Simulation results also show that the maximum participation time of a node is indeed upper-bounded by the analytical bound of $(3/40)k + 4.36\ln k + 3$ slots. Finally, the average number of transmissions is upper bounded by $e + 1$ independent of the value of k.

We have analyzed EMCRR assuming that the idle slots have the same length as the control slots. However, in many systems the receiver can detect that a slot is idle and subsequently send a feedback indicating that the slot is over. Depending on the capacity of the wireless medium, idle slots can be shorter than the control slots. In IEEE 802.11b specification that most currently available wireless interfaces follow, the idle slots can have half the length as it takes to transmit a control message (i.e., an RTS)[4][10]. We have experimentally studied the performance of EMCRR w.r.t. that of other approaches in this case as well. In this case, EMCRR still assumes that the idle slots have the same length as the control slots, whereas other systems terminate the idle slots faster. EMCRR is significantly better than the others even for this case. We present comparisons of EMCRR with the existing approaches, IEEE 802.11[5][10], FCFS[2], CRAI[6], CARMA-MC[7]. We observe that only IEEE 802.11 is sensitive to the length of the idle slots.

As a collision resolution scheme, IEEE 802.11 employs a random backoff scheme where each colliding node backs off a random number of slots for its retransmission. The backoff window size increases exponentially upon every retransmission. FCFS, CRAI, and CARMA-MC are all tree splitting approaches where colliding nodes split into two subsets and one of the subsets attempts retransmissions and the other waits for the resolution of the first subset. In tree splitting approaches, all contenders send RTS to the receiver in a slot, and the receiver broadcasts the feedback in the next slot indicating whether the previous slot is idle or has a collision, or has a successful RTS transmission. The difference among the tree splitting approaches is how to split contenders when there is a collision. In FCFS, a randomized tree splitting approach, the contenders split randomly into two subsets. The nodes choose one of the two subsets with equal probability. In CRAI, an energy measurement based tree splitting approach, the feedback indicates the number of collisions in the event of collision, and the splitting probability depends on the number of collisions in the previous slot and can be computed by dynamic programming. In CARMA-MC, a deterministic tree splitting approach, the receiver announces an ID range through the feedback. Only the contenders which have IDs in the announced range send an RTS in the following slot. If there is a collision, the receiver divides the ID range into halves, and the contenders split according to the divided ID ranges.

[4] In IEEE 802.11, a node needs to wait for a random number of (idle) slots to transmit a packet[10]. However, there is no notion of control slot as such. We introduce the notion of control slots into 802.11 model in our simulations for the purpose of even comparison. This minor change does not cause any noticeable difference in performance.

[5] IEEE 802.11 considers a single channel wireless adhoc networks. As a result packets transmitted around the same time by neighbors collide, even though they may not be directed to or from the same node. Such collisions are ruled out in our case as nodes have locally unique designated channels. So, to obtain even comparison, we consider a set of transmitters trying to reach a single receiver. The only ongoing transmission is between these transmitters and the receiver, and there is no other transmission attempt in the vicinity.

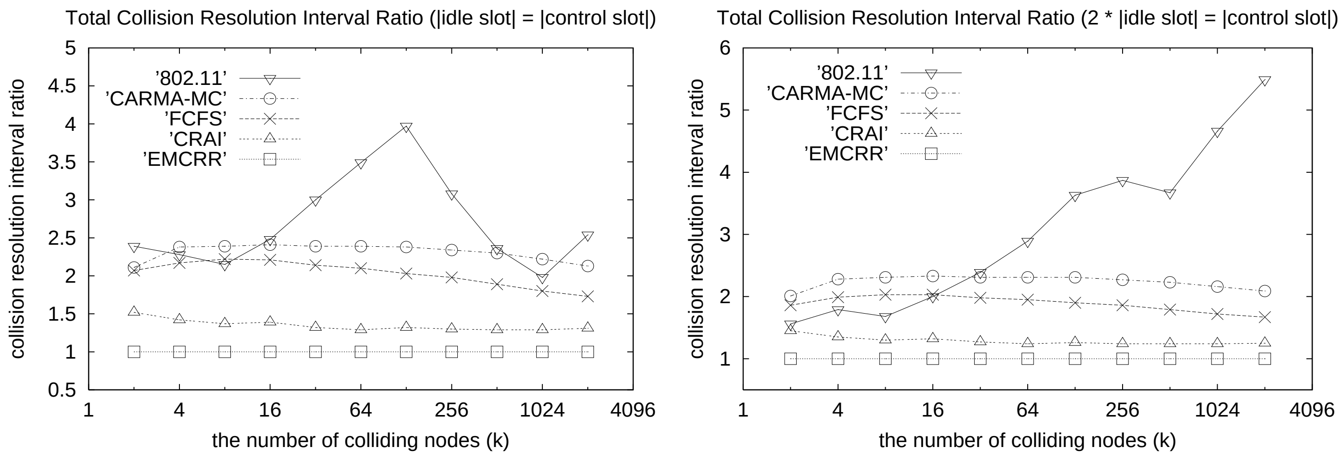

Fig. 3. We plot the ratio between the total collision resolution interval obtained by different protocols and that by EMCRR versus the number of contenders k. The ratio is 1 for EMCRR. We consider two different lengths for the idle slots. In one, idle slot length equals the control slot length, and in another idle slot length is half that of the control slot length.

A. Expected Collision Resolution Interval

We first consider the case when idle slots have the same length as the control slots. Figure 3 indicates that the average collision resolution interval for IEEE 802.11 is always more than 2.3 times that of EMCRR for all k. If length of an idle slot is $1/2$ that of the control slots, then the improvement factor is always more than 1.8 for small values of k. For FCFS [2] the overall collision resolution interval is always more than 1.7 times that of EMCRR for both different lengths of the idle slot. The expected collision resolution time is 2 times more in CARMA-MC[7]. We considered an ID-range of 10000 in CARMA-MC[6]. The significant improvement is due to the successful usage of the number of collisions in each slot. If the number of collisions is low, then the nodes should re-attempt relatively soon, so as to improve the channel utilization, whereas if the number of collisions is high, then the nodes should choose re-attempt intervals from a large range to avoid repeated collisions. We infer this straight away from the number of colliding nodes in the event of a collision and choose the re-attempt slots accordingly. IEEE 802.11 learns this adaptively, by gradually increasing the range of re-attempt intervals with increase in the number of repeated collisions. The splitting approach in FCFS gradually adjusts the attempt probability in accordance with the number of contenders. The collision resolution interval in CARMA-MC depends on the total number of neighbors of any receiver. This number can be very high even though the number of contenders may be small. None of these use the number of colliding nodes in the event of a collision whereas EMCRR uses this information every round to adjust the attempt probabilities. CRAI[6] uses the collision multiplicity in the collision resolution process as well. CRAI has an expected collision resolution interval of $3.74k$ where k is the initial number of contenders as opposed to $2.78k$ of EMCRR[7]. Thus the analytical bounds indicate a 1.34 factor of improvement and the experimental results confirm the analysis. The improvement is because of the reduction in the feedback information. The total feedback in EMCRR is $3/40k$ slots in all, whereas CRAI has $1.87k$ feedback slots on an average. This is because CRAI needs feedback to indicate collision, success or idle slot. However, EMCRR needs feedback only for the successful slots. We would like to mention in this context that CRAI needs to compute the splitting probabilities via dynamic programming which is a computationally intensive operation. This needs to be performed off-line and stored. This increases the storage requirements and renders the policy computationally complex. In contrast, our approach is simple.

B. Expected Participation Time of Transmitters

Figure 4 compares the average participation time of the transmitters. The performance of EMCRR is always 2.3 times better than that of IEEE 802.11 when the idle slots have the same length as the control slots. The improvement factor is at least 1.8 when idle slots have half the length of the control slots. FCFS[2] has at least 1.9 times the average participation time as EMCRR. The factor is greater than 2 for CARMA-MC[7] and 1.5 times for CRAI[6]. The relative performance of EMCRR improves

[6]Note that the actual number of neighbors may be smaller than the ID range. This is because a receiver may not be updated with the IDs of all its current neighbors on account of mobility. So it needs to choose from a large range.

[7]Note that [6] analytically shows that the collision resolution interval is $1.87k$ disregarding the feedback slots completely. However, a feedback is required after every slot in CRAI, and in current wireless networks the feedback slots typically have the same length as the contention slots, and hence the actual number should be $3.74k$.

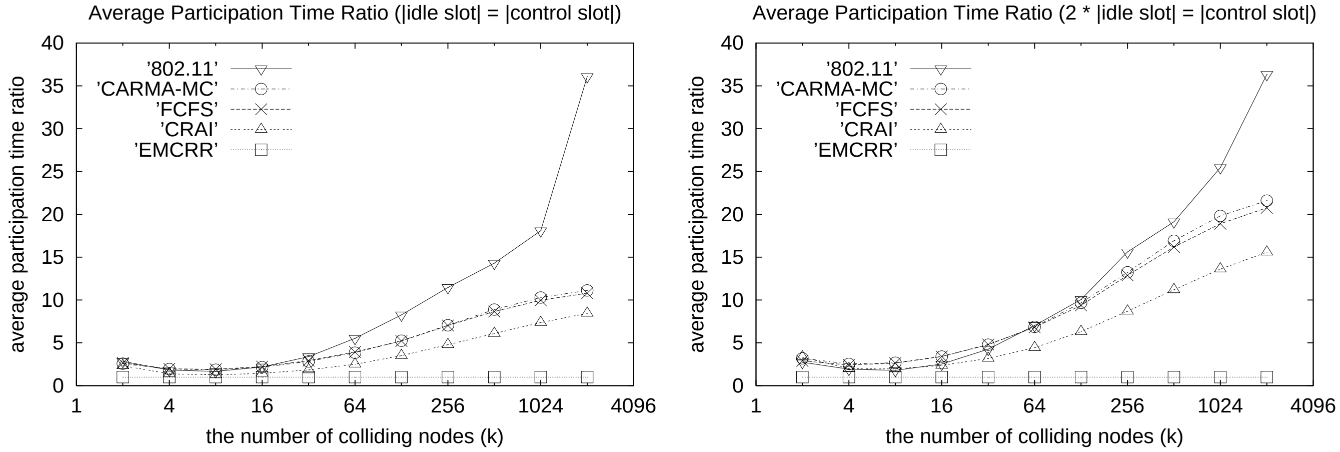

Fig. 4. We plot the ratio between the average participation time obtained by different protocols and that by EMCRR versus the number of contenders k. The ratio is 1 for EMCRR. We consider two different lengths for the idle slots.

even further with an increase in the number of contenders. This happens as in FCFS and CRAI every contending node needs to continuously monitor the receiver feedback in order to decide the splitting which is in turn used in determining whether or not the node contends in the next slot. The feedback is sent after every slot. Thus every node needs to be active at least every other slot. This holds for CARMA-MC as well because every node needs to receive the ID announcements in order to decide whether or not to transmit in the next slot. A transmitter needs to monitor the channel even when it is neither transmitting nor receiving in IEEE 802.11, in order to determine whether the slot is idle or not. This is because a node can attempt after a certain number of idle slots only. For EMCRR a transmitter knows the exact slots in which it would transmit or receive. It transmits in a randomly chosen slot every round and receives feedback at the end of the round. The length of the rounds are also known. Thus the transmitter can sleep in between. This explains the low average participation time of a transmitter in EMCRR.

C. Expected Number of Transmissions

Figure 5 compares the average number of transmissions. IEEE 802.11 has fewer average transmissions than EMCRR for small values of k ($k < 64$) but this number rapidly increases with increase in the number of contenders. EMCRR has fewer average number of transmissions as compared to all the other approaches. Also, for EMCRR, this number is upperbounded by $e + 1$ for all k, whereas the average number of transmissions increase with an increase in the number of contenders for all other approaches.

D. Expected Power Consumption:

Finally, we examine the total power consumed in the different approaches. For this purpose, we assume that a node consumes 280 mA for transmission, 204mA for reception, 177mA for monitoring the channel, and 15mA when it is sleeping[5]. Note that the actual value of the transmission power will depend on the transmission conditions and power control, but we use rough estimates. Figure 6 compares the power consumption. EMCRR improves the power consumption by 9 times as compared to IEEE 802.11 when idle slots have the same length as the control slots, and by 6 times when idle slots have half the length of the control slots. The power consumption is at least 10 times more for CARMA-MC for larger idle slots and at least 6 times more for smaller idle slots. The power consumption is at least 2 times more for FCFS and CRAI for $k \geq 8$ for both lengths of idle slots. For all the approaches the improvement factor increases with an increase in k. The significant improvement is mainly because of much smaller average participation time for EMCRR.

V. Conclusion

We have proposed a new energy measurement based collision resolution approach, called EMCRR. We identified some relevant performance evaluation metrics and evaluated the performance of EMCRR w.r.t. these metrics. Analytical and experimental performance evaluation indicate that EMCRR significantly outperforms all existing approaches. In addition, EMCRR is computationally simple and does not involve sophisticated node functionality. There are several possible directions of future work. First, we would like to investigate the performance of EMCRR when nodes join and leave the collision resolution process in between. Next, we implicitly assumed that the receiver can accurately estimate the number of contending transmitters every time there is a collision. We would like to mention that similar assumptions have been made elsewhere as well[6]. In practice, the estimation is inaccu-

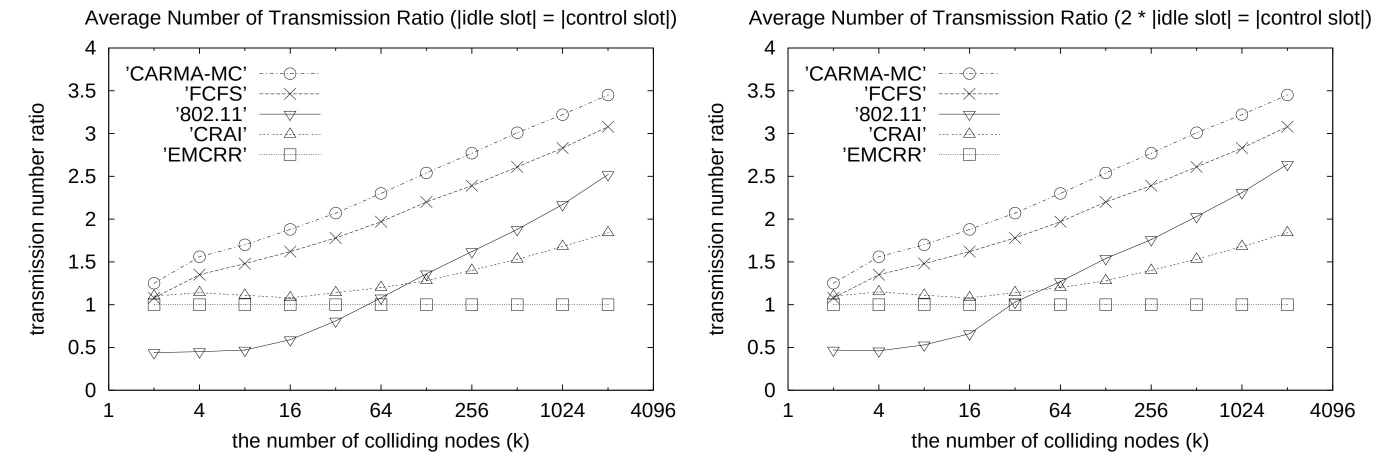

Fig. 5. We plot the ratio between the average transmission number obtained by different protocols and that by EMCRR versus the number of contenders k. The ratio is 1 for EMCRR. We consider two different lengths for the idle slots.

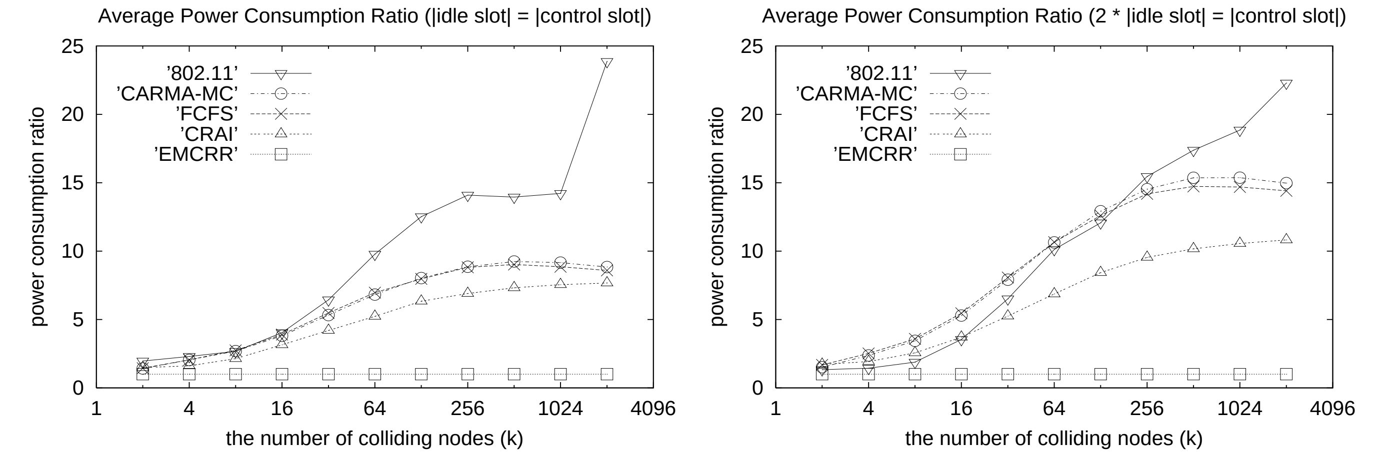

Fig. 6. We plot the ratio between the average power consumption obtained by different protocols and that by EMCRR versus the number of contenders k. The ratio is 1 for EMCRR. We consider two different lengths for the idle slots.

rate on account of noise and imprecise power control. The study of the error resilience of EMCRR remains a topic of future research.

REFERENCES

[1] B. Miller, C. Bisdikian, A. Edlund, *Bluetooth Revealed: The Insider's Guide to an Open Specification for Global Wireless Communications,* Prentice-Hall, 2000.

[2] D. Bertsekas and R. Gallager, *Data Networks*, Englewood Cliffs, NJ:Prentice-Hall, 1987.

[3] I. Capetenakis, "The Multiple Access Broadcast Channel: Protocol and Capacity Considerations", *IEEE Transactions of Information Theory*, pp. $505-515$, 1979.

[4] R. Brewster and A. Glass, "Simulation Model of Sequential Multichannel Network and its Throughput Determination," *Electronic Letters,* vol. 25, No. 15 pp $941-942$, 1989

[5] L. Feeney, "Investigating the Energy Consumption of an IEEE 802.11 Network", *SICS Technical Report SICS-T-99/11-SE, Swedish Institute of Computer Science*, December, 1999.

[6] L. Georgiadis and P. Papantoni-Kazakos, "A Collision Resolution Protocol for Random Access Channels with Energy Detectors", *IEEE Trans. on Communications,* vol. 30, No. 11, 1982.

[7] R. Garces and J.J. Garcia-Luna-Aceves, "Collision Avoidance and Resolution Multiple Access for Multichannel Wireless Networks", *Proceedings of IEEE INFOCOM 2000,* Tel-Aviv, 2000.

[8] J.J. Garcia-Luna-Aceves and J. Raju, "Distributed Assignment of Codes for Multihop Packet-Radio Networks," *Proceedings of IEEE MILCOM '97,* Monterey, CA, November, 1997.

[9] F. Hayes, "An Adaptive Technique for Local Distribution", *IEEE Transactions on Communications,* Vol.26, 1978, pp $1178-1186$.

[10] IEEE, "IEEE Std 802.11b - Wireless LAN Medium Access Control(MAC) and Physical Layer(PHY) specifications: High Speed Physical Layer (PHY) in the 2.4 GHz Band", IEEE Standard 802.11, June 1999.

[11] S. Khanna, S. Sarkar, and I. Shin, "An Energy Measurement Based Collision Resolution Protocol" Tech. report available at http://www.seas.upenn.edu/∼swati/publication.htm

[12] J. G. Proakis, *Digital Communications*, McGraw-Hill, 2000.

[13] T. S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice-Hall, 1996.