Cross layer design for Medium Access Control in CDMA Ad-hoc Networks.

Amit Butala and Lang Tong

Abstract—A medium-access control (MAC) protocol for spread-spectrum ad-hoc networks with Dynamic Channel Allocation (DCA) is presented. DCA can support large systems with a smaller number of channels by dynamically assigning channels only when a node has a packet to transmit. The protocol extends cross layer, with the scheduling at the MAC, and assignment of channels at the physical layer by means of a Query.

It is shown that DCA is collision free under ideal conditions. By assigning channels dynamically, DCA offers improved throughput normalized by available bandwidth.

Analytical results are presented for the performance of the query detection and the throughput for a fully connected network.

Index Terms— MAC, dynamic channel allocation, spread-spectrum, Query, hypothesis detection

I. INTRODUCTION

THERE are several challenges in the design of Medium Access Control (MAC) protocol for code division multiple access (CDMA) ad hoc networks. While it is possible to apply single channel MAC protocols such as MACAW [4], DBTMA [5], and FAMA [6] to a multichannel CDMA network by treating channels independently, such approaches do not exploit the rich diversity of CDMA, nor do they offer an efficient utilization of available spectrum. Specifically, the classical problem of hidden/exposed nodes manifests itself differently in the presence of multiple access channels; multiple data channels and a control channel can co-exist using different spreading codes. If the spreading codes have good cross-correlation properties, contention on one channel does not cause interference on the other channels. The selection of a channel, from a set of channels, to transmit upon, however, is an issue that has not been well addressed in literature.

Spread-spectrum protocols were introduced by Silvester and Sousa [7]. Based on the preassignment of codes, these protocols are identified as Common-Transmitter (CT), Common-Receiver (CR), and Transmitter-Receiver (TR). The CT protocol is the better suited protocol for ad hoc networks since it is less complex and requires a smaller set of spreading codes. In the CT protocol, a node may begin a transmission on the transmitter-assigned code at any time. As there is no feedback on the status of the node, transmissions may be scheduled to nodes unable to receive. Moreover, an a priori assignment of transmit codes is assumed for all nodes in the network. This requires that the number of spreading codes be equal to the number of nodes in the system and necessitates the use of larger than necessary spreading sequences.

MACA-CT [8] improves on the Common-Transmitter scheme of code allocation by the use of a control sequence over the common channel. Medium access is time-slotted. A node sends a Request-to-Send (RTS) at the beginning of a time-slot and is scheduled to transmit data *only if* the intended receiver acknowledges the request with a corresponding Clear-to-Send (CTS). This prevents transmissions to busy nodes. Here too, an apriori assignment of transmit codes is assumed for all the nodes in the network.

In CHMA [9], on the other hand, all the nodes follow a common channel-hopping sequence with each hop duration equal to the amount of time needed for nodes to receive the control packet, either an RTS or a CTS, from a neighbor. The RTS-CTS is followed by data transmission on the same channel, while all other nodes *hop* to another channel. CHMA performs better than the other protocols mentioned earlier under ideal circumstances, but a few factors need to be considered. The hopping channel length has to be at least as long as the length of the packet, which can be a significant penalty as the length of the data-packets increase within relatively small neighborhoods. Longer data-packets may increase the network throughput but require a larger spreading gain to generate the larger number of spreading codes in the channel hopping sequence. The problem of bandwidth utilization remains overlooked.

A common drawback in each of the above protocols is the need for large spreading gains, which imposes a severe penalty on the bandwidth utilization.

A. Contributions

Detection of signals is an integral part of MAC. All control signalling based schemes require the detection of RTS and CTS. Protocols such as DBTMA must detect the presence of busy tones. In the presence of multipath fading, such detections can not be assumed perfect; missed detections and false alarms may have an adverse effect on the protocol performance. Unfortunately, the problem of optimal detection for maximizing MAC throughput has not been considered.

In [12] we proposed a new MAC protocol to tackle the issue of efficient spectral utilization. Referred to as the Dynamic Channel Allocation (DCA), this protocol requires only a fixed number of codes irrespective of the size of the network. Codes are dynamically assigned using a receiver-based request detector.

Amit Butala is with Qualcomm Inc., `abutala@qualcomm.com`.

Lang Tong is with the School of Electrical and Computer Engineering, 384 Rhodes Hall, Cornell University, Ithaca, NY 14853. `ltong@ee.cornell.edu`.

In this paper, an optimal design of the request detector is presented. Assuming a Rayleigh fading model, a Neyman-Pearson detector is used with the detection threshold optimized for throughput. In order to perform such an optimization, a Markov chain analysis is used to obtain the relation between the detector level and normalized throughput.

Such a cross layer design enables us to eliminate the dependance of the spreading gain on the number of nodes in the network and assign channels dynamically.

B. Structure

The structure of the paper is as follows. In section II, we discuss the model assumed for ad hoc networks. Section III elaborates the design of the new protocol and the receiver for dynamic channel allocation using a binary hypothesis model for channel occupancy and a busy tone backoff strategy. In section IV, we build a Markovian representation of a fully connected ad hoc network. Analytical bounds on the throughput of the network are computed and compared with our implementation of the protocol. The results of comparisons between existing multi-channel protocols and DCA are presented in section V. Relevant conclusions and foresights into the modeling of ad hoc networks are summarized in section VI.

II. NETWORK MODEL

Consider a hypothetical multi-hop network as shown in figure 1. We use the protocol model definition for the neighborhood of a node. Thus, each node within a fixed radius (R) of the transmitter is assumed to be contained in its neighborhood and can listen to the transmitter. The relationship is dual; a node is not affected by any transmission that orginates outside its neighborhood. It is assumed that all the nodes transmit with a fixed transmit power.

The network consists of N nodes spatially distributed. Not all nodes are able to communicate with each other. The coverage areas for the nodes are represented by the circles centered at the respective nodes. Clearly, transmissions from A to B have to resolve potential contention with nodes C and E.
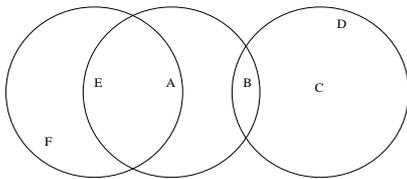

Fig. 1. An ad hoc network

We assume $M + 1$ distinct spreading codes available for transmission where M may be less than N. The codes are designed with good correlation properties [18] such that transmissions using one code do not destroy reception on any of the other codes. As mentioned before, each code identifies a unique channel.

One of the channels is reserved for transmission of control sequences while the other M channels can support the data-packets. Each node makes a choice of transmitting to a node in its neighborhood on any one of the M data-channels. Issues related to routing are not considered. It is assumed that the nodes either know the routing tables apriori or the range of communication involves only neighboring nodes.

Nodes are half duplex and can tune to only one channel at any given time. In addition, nodes also have a frequency generator/receiver that may be used to transmit/receive a monotone on a preset frequency. This is used to specify a busy signal during packet reception.

Transmission time is slotted and the transmissions are packet synchronized. The data is broken up into mini-packets that are transmitted in succession, with each mini-packet requiring one time-slot. The RTS and the CTS packets are assumed to be less than one half mini-packet in length such that a RTS-CTS packet exchange between any two nodes in the network may be completed in a single mini-slot.

A. Normalized Throughput

Since the number of channels in the system that satisfy the constraints on multi-access interference is proportional to the spreading gain, the absolute performance cannot be inferred simply by observing the raw network throughput. The network throughput is expected to increase with an increase of spreading gain and hence we introduce the concept of normalized throughput for comparison of different protocols.

The *network throughput* (Γ) is defined as the average number of packets successfully received in one time-slot over the network when in steady state. The *spreading gain* (G) is the ratio of the chip rate to the symbol rate of a spread-spectrum signal. Then the normalized throughput (η) can be defined as the ratio of the network throughput to the spreading gain,

$$\eta = \frac{\Gamma}{G}. \qquad (1)$$

Multi-access interference can be largely eliminated if the codes are orthogonal to each other. In such codes, such as Walsh codes, the spreading gain is equal to total number of channels available to the system. The normalized throughput would then be the ratio of the network throughput to the total number of channels. This metric is used in all subsequent discussions to compare protocol efficiencies.

III. DYNAMIC CHANNEL ALLOCATION

Fixed channel allocation schemes discussed so far increase the number of channels required in accordance with either the size of the network or the length of the data-packet. A demand-driven dynamic allocation of channels is proposed as one solution for overcoming this constraint. DCA relies on the assignment of one of the available data-channel to the nodes that get scheduled to transmit. Thus, the two basic requirements for packet exchange are, scheduling of packets and allocation of channel.

- *Scheduling:* For a successful transmission, there should be only one transmitter attempting to transmit to a node, and any such transmission must be destined to an idle node. This is effected by the transmission of the RTS-CTS on the control channel. Since the channel is a collision channel

and multiple transmissions on the same channel result in packet collision, the RTS-CTS ensures proper scheduling of the transmissions.

- *Allocation:* Given that two terminals are scheduled, there must be a channel available for transmission that does not interfere with any on-going transmission.

  This is effected by a new procedure called *Querying of channels.*

A. Querying of Channels

The RTS-CTS control packet exchange establishes the scheduling of packets over a particular channel, but it does not ascertain the availability of the channel. A channel is said to be available only if no node in the neighborhood of the intended receiver is transmitting on that channel, and no other node in the neighborhood of the intended transmitter is receiving on that channel. These are, respectively, the conventional exposed terminal and hidden terminal problems that need to be addressed in ad hoc networks. Thus, in our figure for a typical ad hoc network (fig. 1), Node A may transmit to a node B on a specific channel L only if node C (the hidden node) is not transmitting on channel L and node E (the exposed node) is not receiving on channel L.

Overcoming the exposed terminal problem necessitates a response from other nodes in the neighborhood (*i.e.* E) if it is receiving data on the same channel. The hidden terminal problem necessitates a response from B to contention due to transmission from any node in the neighborhood of B that A might be hidden from (*i.e.* C).

The solution is the transmission of a *Query* by the intended transmitter, A. The Query is a known data-packet and thus is a deterministic interference that may be estimated. Once a data-transmission is scheduled using the RTS-CTS exchange, the transmitter sends out the Query on the selected channel.

In response to contention, if any, caused by the Query, the receiver transmits a *busy tone*. The busy tone is a sinusoid sent on an out-of-band frequency, and intimates the transmitter that the channel is in use. A Query is successful only if no busy tone is heard by the transmitter. This represents the case that no exposed terminal is receiving, and no hidden terminal is transmitting, on the selected channel. A node may transmit only if its Query is successful.

With the introduction of the Query, in each time-slot, all the nodes may be classified into the following 4 states:

- *Idle* (or Backlogged) state: Nodes that are not engaged in packet reception or transmission.
- *Query* state: Nodes that get scheduled and are transmitting the Query in the current time-slot.
- *Data* state: Nodes involved in transmission or reception of data packets. Only nodes in the Data state successfully transmit data over the network.
- *Locked* state: An extra state that tracks nodes involved in data-packet collisions. This occurs due to a mis-detection of the Query and shall be been discussed in greater detail in the next section.

B. The Protocol

The Dynamic Channel Allocation protocol is defined below and has been illustrated in Fig. 2, Fig. 3 and Fig. 4.

- Any idle node (*e.g.* A) that has a packet to transmit to any of its immediate neighbors (*e.g.* B) attempts to establish a communication by broadcasting an RTS on the common channel at the beginning of the mini-slot (Fig. 3 and Fig. 4).
- The RTS contains the following information: the destination node (B) identifier, the transmitting node (A) identifier, and the selected channel (**Q**) on which the data will follow. The channel (**Q**) is randomly chosen from the set of available channels.
- If the destination node, B, receives the RTS, it responds immediately, in the same time-slot, with a CTS on the common channel (Fig. 3 and Fig. 4). B transitions from the Idle state to the Query state for the next time-slot and tunes its receiver to the selected channel, **Q**.
- If A does not receive a CTS in the same time-slot, it times out and reverts back to the Idle state. A re-transmission is attempted according to the back-off strategy. If A does receive the CTS, it moves from the Idle state to the Query state. This completes the scheduling.
- In the next mini-slot, A transmits a Query on the selected channel. The Query is successful if no busy tone is generated (Fig. 3).
- The busy tone is generated in two possible cases:
  - By the intended receiver, B, if the Queried channel is already in use (Fig. 2.b)
  - By the contended receiver, D, if the selected channel is already in use (Fig. 2.c)
- If A receives a busy tone on the busy-tone frequency, (cases b or c) it aborts transmission on the channel and reverts to the Idle state (Fig. 4). If A does not receive a busy tone on the busy-tone frequency, it moves to the Data state and begins transmission of the data packet from the time-slot that follows (Fig. 3).
- At the end of the data transmission interval, which is an integral number of mini-slots, both A and B reset to the Idle state. □
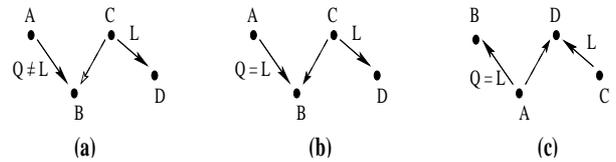

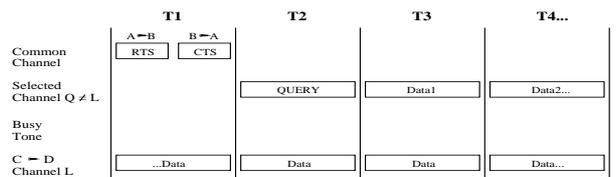Fig. 2. Query for different network states: a) success b) failure c) failure
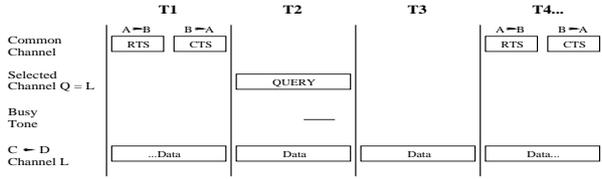

Fig. 3. Successful Querying: case (a)

Fig. 4. Failed Querying: cases (b) and (c)

Lemma: Under the assumption of perfect detection of the Query, there are no data-packet collisions.

Data State: Transmitter
T1. Send an RTS at the beginning of the time-slot. - Wait for CTS. - If no CTS is received, time-out and revert back to Idle.
T2. If CTS is received, transmit a Query on the selected channel in the next time-slot. - A busy tone indicates a busy channel; abort transmission; revert to Idle. - If no busy tone is heard, accept channel and
T3. transmit data-packets on channel.

Idle Nodes
T0. Idle nodes are tuned to common channel.
T1. If an RTS is received, decode the intended receiver. - If the RTS is intended for the particular node, respond with a CTS on the common channel. - Tune to the transmitter specified channel.
T2. Detect Query, - If Query fails, raise busy tone; revert to Idle. - If Query succeeds, switch states to Data
T3. Receive data-packets.

Data State: Receiver
T0. Receivers are tuned to the transmitter specified channels.
T2. Raise busy tone if the presence of Query is detected.
Tn. Revert to Idle when the transmission of data-packet completes.

TABLE I

DCA: ALGORITHM

Proof: If the detection of the Query is perfect, a busy tone is raised only if there is contention either at the Query-receiver or the Data-receiver. This is the case that the selected channel is in use either for reception at an exposed node or for transmission at a hidden node. In both cases the intended transmitter should stop. This is the desired result of the busy tone.

In addition the deterministic nature of the interference by the Query permits data-packet decoding even in the presence of the Query. Thus, under perfect conditions, there is no loss of data-packets due to collisions.   □

C. Detection of the Query

In the presence of noise and multi-access interference, the detection of the Query is not perfect and is contingent on the operating characteristics of the receivers. At every receiver, the interference due to the Query may be missed or a false alarm be raised in response to a Query that does not interfere. This results in a probabilistic model for the acceptance of the Query.

Missed-detection: In the case of a Missed-detection of the Query, there will now bw two (or more) nodes transmitting on the same channel within the vicinity of the receiver. This results in a packet collision at the receiver and it is unable to detect either packet. The receiver and the corresponding transmitters are assumed to be in the *Locked* state. The throughput of node-pairs in the Locked state is zero.

Transition of node-pairs out of the the Locked state would depend on the coding scheme used and the higher layer scheduling. Without imposing any additional constraints, we assume that the pair remains in the Locked state till the end of the current data-packet transmission, after which they reset to the Idle state.

False-alarm: The False-alarm induces less damage, since it merely results in the node (*i.e.* A) aborting the transmission of the data-packet and reverting back to the Idle state. A re-transmission is attempted in accordance with the protocol. This too, would lead to a decrease in the throughput of the network.

The two parameters are related, thus the optimization of the throughput requires an analysis of the receiver operating characteristics. Two types of nodes need to process a Query:

- Data State: Nodes currently in reception.
- Query State: Nodes attempting to tune to the transmitter.

1) Detection of the Query during the Data State: We make the following assumptions:

- Each mini-packet has a fixed packet size of K bits.
- A header of *pilot training* bits ($\kappa \ll K$) is embedded in each data-packet to aid channel estimation and timing synchronization.
- The total number of data-channels is M.
- The channel undergoes slow Rayleigh fading. The amplitude of the fade (A) can be assumed complex circularly Gaussian and constant over one time-slot: $A \sim \mathcal{N}(0, \phi^2)$.

Then, for any particular receiver in the Data state, the received signal can be written as [14],

$$y(t) = \sum_{m=1}^{M} \sum_{k=0}^{K-1} A_m b_m[k] s_m(t - kT - \tau_m) + \eta(t) \qquad (2)$$

where A_m denotes the signal power on the mth channel, $b_m[k]$ denotes the kth bit on the mth channel, $s_m(\cdot)$ is the signature waveform of the mth channel, τ_m is the timing offset of the mth channel and $\eta(t)$ is the additive white gaussian noise (AWGN) at time t.

Let the receiver be tuned into some channel, L. Transmissions on all other channels is secondary interference and under the interference model assumed is treated as AWGN. Transmissions on the same channel, however, cannot be ignored.

By the definition of the protocol, no other data transmission can be on the same channel as long as the receiver is in the Data state. Thus, the primary interference, if it exists, is due to the transmission of a Query. Let the query be transmitted on channel Q which may or may not be the same as L. We use $\delta_{Q,L}$ to denote the interference of the Query at the receiver. Thus,

$$\delta_{Q,L} = \begin{cases} 1 & \text{if Query present in the current slot} \\ & \text{and transmitted on channel } L. \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

Then, the received signal at the output of a matched filter that is synchronized to channel L can be represented as,

$$y[k] = A_L b_L[k] + A_Q \Big[b_Q[k]\rho_{L,Q}(\tau) +$$

$$b_Q[k+1]\rho_{Q,L}(\tau) \Big] \delta_{Q,L} + n[k] \qquad \forall k = 1 \dots K. \quad (4)$$

where, $\rho_{L,Q}$ and $\rho_{Q,L}$ are the cross-correlation between the channels L and Q on the interval over which the bits, $b_Q[k]$ and $b_Q[k+1]$ respectively, overlap bit $b_L(k)$, and $n[k]$ is the filtered output of the secondary interference and the noise in the kth bit.

Detection of the Query is a binary hypothesis testing problem, hence for simplicity of the receiver, we set all the bits in the Query to 1, i.e. $b_Q[k] = 1$. Also, assuming good correlation properties on the channels, the output signal at any receiver in the Data state is,

$$y[k] = A_L b_L[k] + A_Q \delta_{Q,L} + n[k]. \quad (5)$$

Thus, two hypothesis can be formulated as below.
 − The Null Hypothesis (H_0): The Query is not on the same channel ($\delta_{Q,L} = 0$)

$$H_0 : y[k] = A_L b_L[k] + n[k]. \qquad \forall k = 1 \dots \kappa \quad (6)$$

 − The Alternative Hypothesis (H_1): The Query is sent on the same channel as the data-packet ($\delta_{Q,L} = 1$)

$$H_1 : y[k] = A_L b_L[k] + A_Q + n[k]. \qquad \forall k = 1 \dots \kappa \quad (7)$$

We assume that in the presence of a slow block fading channel, a node in the data state has already estimated channel fade on the data link (A_L). The fading on the Query (A_Q) is also a constant but cannot be assumed to be known by the reciever. Then, for the duration of the pilot training symbols in each packet, we can define a metric $\tilde{y}$ as given below.

$$\tilde{y} = \sum_{k=0}^{\kappa} (y[k] - A_L b_L[k])$$

This simplifies our hypothesis, (6) and (7) as given,

$$H_0 : \tilde{y} = \sum_{k=0}^{\kappa} n[k] \implies \tilde{y} \sim \mathcal{N}(0, \kappa\sigma^2)$$

$$H_1 : \tilde{y} = \sum_{k=0}^{\kappa} (A_Q + n[k]) \implies \tilde{y} \sim \mathcal{N}(0, \kappa^2\phi^2 + \kappa\sigma^2)$$

This is a standard energy detector problem. For an α level receiver (i.e. probability of False-alarm $P_\alpha = \alpha$), hypothesis H_1 is selected by the Neyman-Pearson detector if,

$$|\tilde{y}|^2 > \kappa\sigma^2 \left[\mathbf{Q}^{-1}\left(\frac{P_\alpha}{2}\right) \right]^2. \quad (8)$$

If the signal-to-noise ratio (SNR) is known, the power of the detector is given by,

$$P_D = 2\mathbf{Q} \left[\frac{\mathbf{Q}^{-1}\left(\frac{P_\alpha}{2}\right)}{\sqrt{\kappa SNR + 1}} \right]. \quad (9)$$

2) Detection of the Query during the Query State: For a receiver in the Query state, the Queried channel is rejected if any of the neighboring nodes transmit on the same channel synchronized with it. Analogous to eqn. (5), the model of the received signal at the receiver in the Query state is,

$$y[k] = A_Q + A_L b_L[k]\delta_{Q,L} + n[k] \qquad \forall k = 1 \dots K. \quad (10)$$

Over the interval of the Query, the channel A_Q is a constant and known at the receiver. Thus, the signal error over the interval of the pilot training bits is,

$$\tilde{\mathbf{y}} = \mathbf{y} - A_L \mathbf{b}_L.$$

The binary hypothesis thus simplifies as,

$$\begin{aligned} H_0 : \tilde{\mathbf{y}} &= \mathbf{n} \sim \mathcal{N}(0, \sigma^2\mathbf{I}) \\ H_1 : \tilde{\mathbf{y}} &= A_L\mathbf{b}_L + \mathbf{n} \sim \mathcal{N}(0, \phi^2\mathbf{b}_L\mathbf{b}_L^T + \sigma^2\mathbf{I}) \end{aligned} \quad (11)$$

which again is differentiable only in one of the singular value components. This yields exactly the same detector from the previous part.

D. Selection of the Threshold for the Detector

For the Neyman-Pearson detector, the threshold of the detector affects the probability of Missed-detection. We assume that all the nodes use the same detector operating at threshold of detection. Thus, the probability of False-alarm and Missed-detection are constant over the network and known apriori.

The throughput is a function of both the parameters, hence the optimal value of the threshold is the one that maximizes this throughput over the ROC of the detector.

The computation of the throughput of DCA shall be addressed in the next section, but as an illustration, shown above in fig. 5 is the throughput-detector plot for a 20 node network, with 5 data channels and a mean data-packet length of 10 mini-packets. The edge of the plane represents the performance of a network when the signal to noise ratio is 2dB. The optimal point is the point on the corresponding ROC curve at which the maximum value of throughput is reached, and as can be seen, is at approximately $P_\alpha = 0.06$.

IV. ANALYSIS OF DCA

The analysis of a multi-hop network is difficult. Factors such as routing and location paging are dependent on the topology
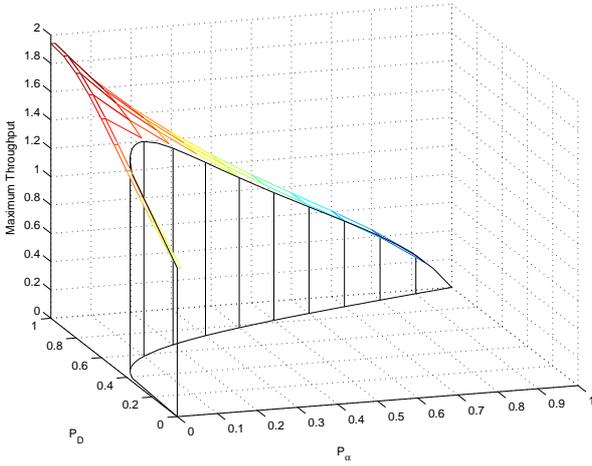
Fig. 5. Throughput of DCA for a fully connected network with 20 nodes, 5 data-channels and a mean data-packet length of 10 mini-packets for different values of the threshold at 2dB.

and hard to model. However, significant insight can be obtained into the performance of an ad hoc network by estimating its performance over a fully connected network. In the next section, we simulate a few representative networks to validate our results.

The throughput of a fully connected single hop network is analysed under the following assumption. Idle nodes have a packet to transmit with a probability p. Backlogged nodes attempt a re-transmission with the same probability p.

The message length of the data packets is assumed to be geometrically distributed. This allows a reduction in the state space by making the model Markovian. If we take q to be the parameter of the geometric distribution, then $P[D = d] = (1-q)q^{d-1}$ and, the average packet length is given by $\hat{D} = 1/(1-q)$.

The system can now be modeled as a discrete time Markov chain, described completely by the number of nodes in each of the four states. (viz, *Idle*, *Query*, *Data*, *Locked*)
Let,

$k =$ number of nodes that transmit an RTS in the current slot.

$l, x =$ number of node pairs in *Query* state (in the current/previous slot)

$m, y =$ number of node pairs in the *Data* state.

$n, z =$ number of node pairs in the *Locked* state.

Also, if N is the total number of nodes in the system, and M is the total number of channels available. Then, the total number of idle nodes (N') during the given time-slot is given by $N' = N - 2l - 2m - 2n$.

Since the system is affected by the detection probability of the Query, we model the performance based on the receiver operating characteristics of the Query detector, namely, the probability of False-alarm (P_α) and the probability of Missed-detection (P_β).

A. The State Transition Probabilities

The Markov chain is completely described by the number of nodes in each state. Given that the total number of nodes in

the network is N, we describe each state by the three identifiers described above, viz. l, m, n. Consider the transition from state *lmn* to state *xyz*

$$
\begin{aligned}
P_{lmn,xyz} &= P(x, y, z | l, m, n) \\
&= P(x | l, m, n, y, z)P(y, z | l, m, n) \\
&= P(x | l, m, n)P(y, z | l, m, n) \quad (12)
\end{aligned}
$$

where step three follows from the knowledge that the number of Nodes in the Query state is determined only by the state of the network in the preceeding slot, or more precisely, only on the number of idle nodes in the previous slot.

1) Computation of $P(x | l, m, n)$: A node pair reaches Query state if the RTS/CTS communication is successful. In a fully connected network, since a maximum of one RTS can be successful in a time slot, the CTS can be granted to be always successful. An RTS is assumed successful if it is transmitted to an idle node. Let this event be denoted by $\mathcal{I}$. Under this assumption, we compute $P(x | l, m, n)$, which represents the probability of an RTS/CTS exchange succeeding in the current time slot. A successful RTS/CTS exchange implies a Query is attempted in the next time-slot, *i.e.* $x = 1$

$$
\begin{aligned}
P(x = 1 | l, m, n) &= P(\text{One RTS is transmitted} \cap \mathcal{I}) \\
&= P(k = 1 \cap \mathcal{I}) \\
&= B(N', p, 1)\frac{N' - 1}{N - 1} \quad (13)
\end{aligned}
$$

where $B(n, p, k) = \binom{n}{k}p^k(1-p)^{n-k}$ is the binomial distribution of selection of k from a set of n when each individual probability of selection is p.

The probability of no success, *i.e.* $x = 0$, is

$$
P(x = 0 | l, m, n) = 1 - P(x = 1 | l, m, n). \quad (14)
$$

2) Computation of $P(y, z | l, m, n)$: Each time-slot can be classified on the basis of the occurrence of four events:

- Query ($\mathcal{Q}$): This corresponds to the event that a node transmits a Query packet over one of the data-channels.
- Interference ($\mathcal{I}$): This corresponds to the event that the Query transmitted is on the same channel as that of one of the Data transmissions.
- Missed-detection ($\mathcal{M}$): This is the event that the Interference of the Query on the Data transmission is missed by the receiving node.
- False-alarm ($\mathcal{F}$): This is the event that any one of the receivers in the Data state raises the busy tone even though the Query is not transmitted on the channel it is tuned to.

We compute the transitions conditioned on the presence of the Query.

If $l = 0$, no Query was sent in the previous time-slot, and no new node-pair starts transmitting. Assume i node pairs in the *Data* state and j node-pairs in the *Locked* state complete

transmission, and thus revert back to *Idle*.

$$P(y, z | l = 0, m, n)$$

$$= \sum_{i=0}^{m} P \left(\begin{array}{c} i \text{ Data pairs} \\ \text{become Idle} \end{array} \right) \sum_{j=0}^{n} P \left(\begin{array}{c} j \text{ Locked pairs} \\ \text{become Idle} \end{array} \right)$$

$$P(y, z | l = 0, m, n, i, j)$$

$$= \sum_{i=0}^{m} \sum_{j=0}^{n} B(m, 1-q, i) B(n, 1-q, j) \delta(y - (m-i)) \delta(z - (n-j))$$

$$= B(m, 1-q, m-y) B(n, 1-q, n-z) \tag{15}$$

where $B(n, p, k)$ is the binomial distribution and $\delta(0) = 1$, and $\delta(x) = 0 \ \ \forall x \neq 0$ represents the acceptable state transitions.

If $l = 1$, a Query was sent in the previous slot. There are 4 outcomes for the Query: Success, Interference-detection, False-alarm or Missed-detection. The probability of successfully establishing a data channel depends on the number of available channels. Let ψ_D node-pairs be added to the *Data* state and ψ_L node-pairs end up in the *Locked* state in the given time-slot.

$$P(y, z | l = 1, m, n)$$

$$= \sum_{i=0}^{m} P \left(\begin{array}{c} i \text{ Data pairs} \\ \text{become Idle} \end{array} \right) \sum_{j=0}^{n} P \left(\begin{array}{c} j \text{ Locked pairs} \\ \text{become Idle} \end{array} \right)$$

$$P(y, z | l = 1, m, n, i, j)$$

$$= \sum_{i=0}^{m} \sum_{j=0}^{n} B(m, 1-q, i) B(n, 1-q, j) \sum_{|\{\psi_D, \psi_L\}|} P(\psi_D, \psi_L)$$

$$\delta(y - (m - i + \psi_D)) \delta(z - (n - j + \psi_L)) \tag{16}$$

In the case of a Missed-detection, there are two nodes transmitting on the same channel within the vicinity of the receiver. This results in a packet collision at the receiver and it is unable to detect either packet. The receiver and the corresponding transmitted are assumed to be in the *Locked* state for the duration of the transmission.

Since our model of the state space does not carry the information about the channel that gets assigned to the transmitter, this case needs to be tackled independent of the knowledge of the number of node-pairs involved in the packet collision.

3) The Upper Bound: Since the nodes are half duplex, there can be no feedback from the receiver to the transmitter. An upper bound can be constructed under the assumption that a 'Genie' informs transmitter involved in a Missed-detection, in which case they immediately stop transmitting. In other words, a Missed-detection causes nodes to move to the Idle state instead of the Locked state. Hence, $z = n = 0$, always.

Only one pair of nodes can be in the Query state in any time slot. On the basis of the occurrence of the above four events, it is clear that $\mathcal{M} \subset \mathcal{I} \subset \mathcal{Q}$. Also, since the network is fully connected, false alarms that are on another channel would also cause the Query to fail. Thus, $\mathcal{F}$ and $\mathcal{Q}$ can be considered as independantly occuring events.

- A Query corresponds to the combination of events,
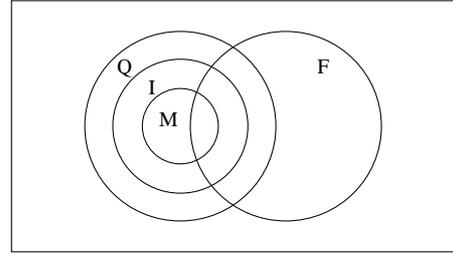$$\{Q \cap F^c \cap I^c\} \Rightarrow (\psi_D = 1, \psi_L = 0).$$

Fig. 6. The event space for DCA with perfect feedback of Missed-detections

- False alarm and Interference detection, both result in the generation of a busy tone and the Query fails. This is the combination of events,
$$\{(Q \cap F^c \cap I \cap M^c) \cup (Q \cap F)\} \Rightarrow (\psi_D = 0, \psi_L = 0).$$
- Missed-detection is the event set,
$$\{Q \cap F^c \cap I \cap M\} \Rightarrow (\psi_D = -1, \psi_L = 0).$$

Conditioned on the arrival of the Query, the probabilities for False-alarm, Missed-detection and Interference are:

$$P_F = P \left\{ \left(\begin{array}{c} \text{False-alarm at at-least} \\ \text{one Data-receiver} \end{array} \right) \bigcup \left(\begin{array}{c} \text{False-alarm at} \\ \text{the Query-receiver} \end{array} \right) \right\}$$

$$= 1 - (1 - P_\alpha)^{m-i+1}$$

$$P_M = P \left\{ \left(\begin{array}{c} \text{Missed-detection at} \\ \text{contended Data-receiver} \end{array} \right) \bigcap \left(\begin{array}{c} \text{Missed-detection at} \\ \text{the Query-receiver} \end{array} \right) \right\}$$

$$= P_\beta P_\beta$$

$$P_I = P \{\text{Non Idle Channel Selected}\}$$

$$= \frac{m - i}{M}$$

Thus we have, from (16),

$$P(y, z | l = 1, m, n)$$

$$= \sum_{i=0}^{m} B(m, 1-q, i) \Big[\delta(y - (m-i+1)) P(\psi_D = 1, \psi_L = 0) +$$

$$\delta(y - (m - i)) P(\psi_D = 0, \psi_L = 0) +$$

$$\delta(y - (m - i - 1)) P(\psi_D = -1, \psi_L = 0) \Big]$$

$$= \sum_{i=0}^{m} B(m, 1-q, i) \Big[\delta(y - (m-i+1))(1 - P_F)(1 - P_I) +$$

$$\delta(y - (m - i)) \Big\{ (1 - P_F) P_I (1 - P_M) + P_F \Big\} +$$

$$\delta(y - (m - i - 1))(1 - P_F) P_I P_M \Big]. \tag{17}$$

4) The Lower Bound: In the absence of feedback from the 'Genie', when a collision occurs the transmitter does not stop transmitting. The node-pair transitions to the Locked state and are unavailable until the transmitter has completed it's transmission.

In addition, since the state space does not carry the information of which channel the Locked transmitter is transmitting upon, we assume that every locked node-pair occupies a different channel. Clearly, this is a very conversative estimate and

provides us with a lower bound for the system in the presence of Missed-detection.

Again there are 4 outcomes for the Query: Success, Interference-detection, False-alarm and Missed-detection. Missed-detection causes transition of node-pairs from the Data state to the Locked state. For receivers in the Locked state, since there are multiple simultaneous transmissions on the same channel, the interference is non-deterministic. The hypothesis detector fails to identify the Query sent on the channel. Hence, receivers in the Locked state do not raise a busy flag, irrespective of the contention.

Thus, depending on whether the Missed-detection was with a node-pair in the Data state or already in the Locked state, $(\psi_D, \psi_L) = (-1, 2)$ *or* $(0, 1)$. Using $\mathcal{I}_T$ to indicate that the interference was with a channel assigned to a Data node-pair and $\mathcal{I}_L$ to indicate interference with a Locked node-pair:
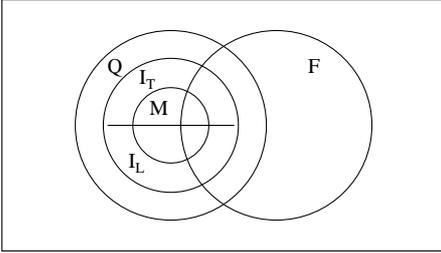

Fig. 7.   The event space for DCA with no feedback

- A Query success corresponds to the events,
$$\{Q \cap F^c \cap I_T^c \cap I_L^c\} \Rightarrow (\psi_D = 1, \psi_L = 0).$$
- False alarm and Interference detection, both result in the generation of a busy tone and the Query fails. This is the combination of events,
$$\{(Q \cap F \cap I_T \cap M^c) \cup (Q \cap F)\} \Rightarrow (\psi_D = 0, \psi_L = 0).$$
- A Missed-detection involving a channel assigned to a node-pair in the transmit state is the event,
$$\{Q \cap F^c \cap I_T \cap M\} \Rightarrow \psi_D = -1, \psi_L = 2).$$
- A Missed-detection when the channel chosen is in the locked state is the event
$$\{Q \cap F^c \cap I_L^c)\} \Rightarrow (\psi_D = 0, \psi_L = 1)$$

Using this, (16) simplifies as follows,

$$P(y, z | l = 1, m, n)$$
$$= \sum_{i=0}^{m} \sum_{j=0}^{n} B(m, 1-q, i) B(n, 1-q, j)$$
$$\left[\delta(y-(m-i+1))\delta(z-(n-j))(1-P_F)(1-P_{I_{T \cup L}}) \right.$$
$$+ \delta(y-(m-i))\delta(z-(n-j))\left\{(1-P_F)P_{I_T}(1-P_M) + P_F\right\}$$
$$+ \delta(y-(m-i-1))\delta(z-(n-j+2))(1-P_F)P_{I_T}P_M$$
$$\left. + \delta(y-(m-i))\delta(z-(n-j+1))(1-P_F)P_{I_L} \right].$$
(18)

B. Throughput

Clearly, the Markov chain is ergodic and thus a steady state distribution exists. Let the probability of being in any state lmn be denoted by S_{lmn}, then the average throughput Γ is equal to the number of nodes-pairs in the transmit state weighted by the probability of being in that state.

$$\Gamma = \sum_m m S_{lmn}. \tag{19}$$

V.   RESULTS AND SIMULATIONS

The maximum throughput of DCA is prone to the operating characteristics of the detector for the Query. Peak throughput depends on both the probability of False-alarm as well as Missed-detection. Each receiver may pick up its operating point based on it's individual requirements. For simplicity, however, we assume that all receivers operate at the same point on the ROC. The throughput then relates to the ROC as shown earlier in figure 5. Once the system SNR is computed at the receiver, the threshold of the detector is set at the point on the ROC curve that maximizes the throughput.

A comparison of the three schemes discussed earlier: MACA-CT, CHMA, and DCA, is made for a fully connected 20 node network carrying data-packets geometrically distributed in length and with a mean length of 10 mini-packets. The number of data-channels depends on the protocol. For DCA, we randomly choose 5 data-channels. MACA-CT has 20 channels, determined by the size of the network. For CHMA, this number would have to be greater than the largest data-packet in the network, which is infinity. We compare against the normalized throughput of Modified-CT which illustrates the best case performance of CHMA for a channel hopping sequence that is twice as long as the length of the average data-packet length. (See Appendix)

The normalized throughput of the 3 protocols are plotted below. The Query detector for DCA is assumed to be operating at 2dB SNR.
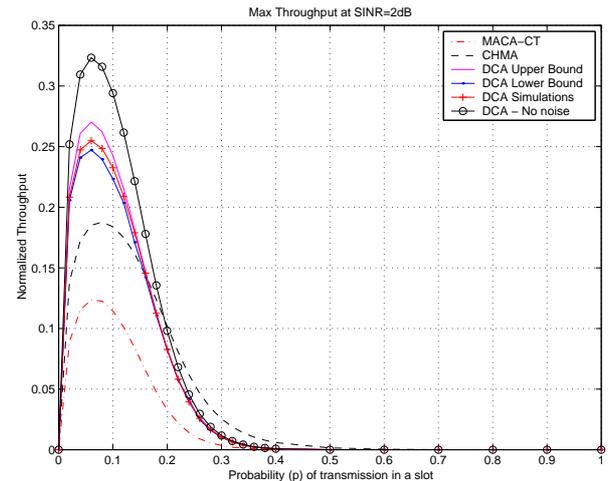

Fig. 8.   Normalized throughput for different schemes

Figure 9, shows the maximum normalized throughput of DCA, at various SNR levels, compared with that of MACA-CT

and CHMA. Significant performance gains are observed for the parameters indicated. Clearly, the scheme performs uniformly better for any probability of transmission and channel interference over the given bandwidth expansion available.
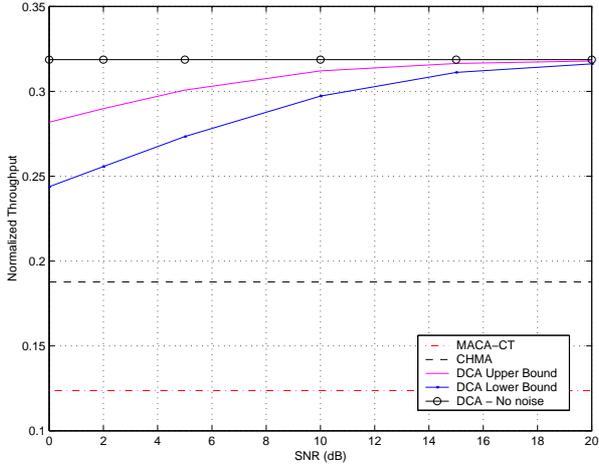

Fig. 9. Maximum throughput of DCA as a function of the SNR ($\hat{L} = 10$ and $M = 5$)

A. Parameter selection

The efficiency of the protocol depends upon the length of the data-packets and the number of data-channels. Increasing the number of channels increases the success rate of the Query and thus the overall throughput per slot. However, this would also require an increase in the spreading gain, thus wiping out the advantages of DCA. Increasing the length of the data-packet should increases the protocol efficiency by reducing the fraction of the number of control packets per packet of data. At the same time, larger data-packets are more prone to collisions which would result in the data channel being Locked for longer intervals. Clearly, there are trade-offs involved in selection of both parameters.

1) Performance as a function of the number of data-channels: From Fig. 10, the normalized throughput appears to be almost monotonically decreasing beyond the addition of the first few channels. The best case performance is for systems with 2 to 4 data-channels. The results are not totally surprising since one might expect the control channel to be the bottle neck as more channels are made available for Data. Increasing the available number of channels does not yield to a proportionate increase in data traffic. Interestingly, performance of DCA is superior till the number of channels equals 8. Fixed channel allocation schemes would yield better throughput than DCA if more channels might be made available.

2) Performance as a function of the length of the data-packet: As seen from Fig. 11, the throughput increases with an increase in the length of the data-packet and then drops off. Again, this is not unexpected since longer data streams are more likely to be involved in Missed-detections of the Query and result in Locked states.

This seems to suggest that the average data-packet should be kept approximately 15 slots. Networks with differing traffic requirements might be able achieve better performance by assigning some channels for longer data packets and maintaining a non uniform probability for selection of channels. This would entitle successful transmission of longer data streams without increasing the latency on the shorter transmissions.

Thus, the gains by Dynamic Channel Allocation are more significant for networks with short data-packets and fewer channels.

B. Transmission Delay

The system delay depends, along-with other factors, on the performance of the Query detect and re-transmission. We can however estimate the minimum delay in packet reception by assuming perfect detection of the Query. The re-transmission policy is defined with a buffer of one packet at each node. the packet arrivals are Bernoulli with a probability p for every idle node.

Similar to the argument given in [9], we use Little's theorem to calculate the average delay. The average delay D is the time taken for a new arriving packet to be transmitted and successfully received by the intended receiver. For a stabilized system, the arrival rate is equal to the throughput of the system (Γ). The total number of nodes (B) in the system are the nodes that are
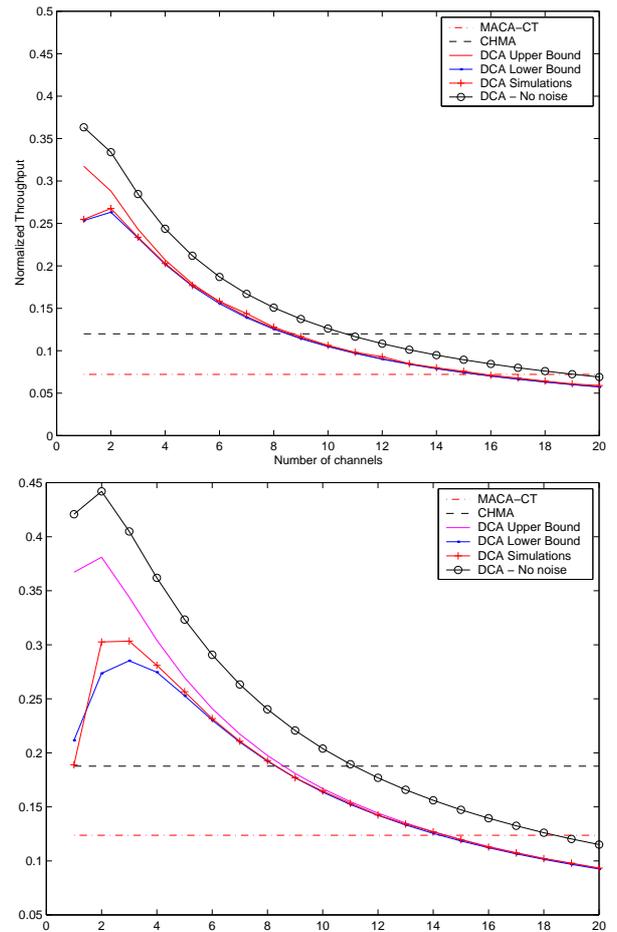

Fig. 10. Throughput as a function of the number of channels for a 20 Node network with average data-packet length $\hat{L} =$(a) 5, (b) 10
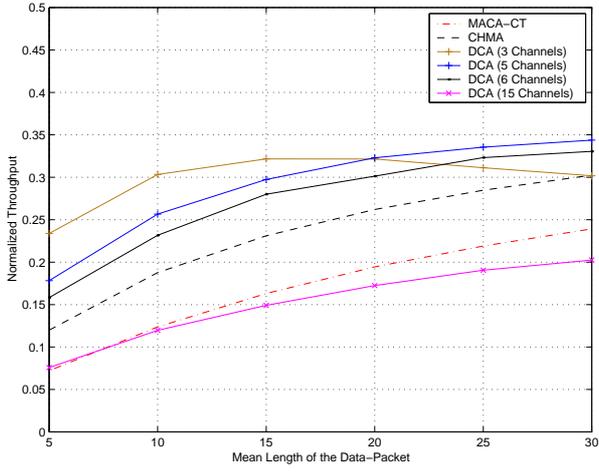
Fig. 11. Throughput as a function of packet length for a 20 Node network (at 2dB)

either receiving, transmitting or have a packet to transmit.

$$B = \sum_{l,m,n} \left[(N - 2m - 2l)p + m + l \right] S_{lmn} \qquad (20)$$

Thus the average delay per mini-packet is $\bar{D} = B/\Gamma$. Since the average packet length is $\hat{L} = 1/(1 - q)$, the average system delay is

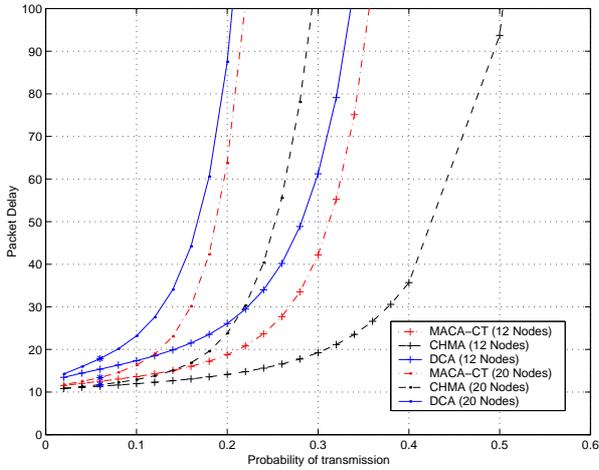$$D = \bar{D}\hat{L} = \bar{D}/(1 - q) \qquad (21)$$

Fig. 12. Mean packet delay for DCA, MACA-CT and CHMA with an average data-packet length of 10 mini-packets

For light loads ($p < 0.1$), the protocols appear to have bounded delays. Delay for DCA is increased due to the additional overhead required for the resolution of the Query. The best case performance of DCA would in-fact be the the curve for MACA-CT and would occur in the event that every Query was successful.

It may also be noted that the delay increases exponentially and is much steeper. Thus, proper selection of probabilities for transmission is very critical. Packet delay at the point of maximum throughput, denoted in the figure above by a "*", though is finite and comparable.

C. Multi-hop Networks

All the above analysis is for a fully connected single hop scenario. Modeling of a multi-hop network is difficult. However, a few reference cases were simulated to postulate the applicability of DCA to multi-hop networks and to exhibit its performance gain over existing protocols.
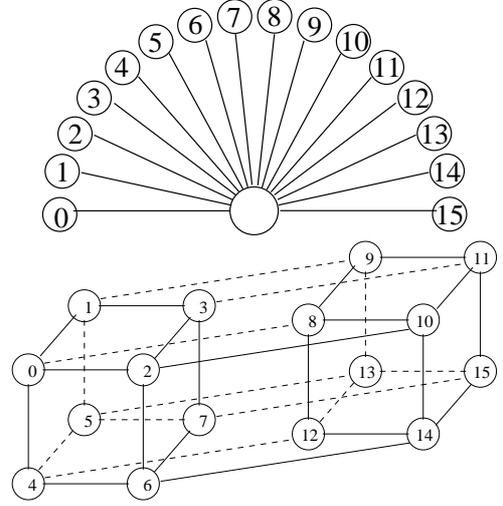

Fig. 13. 16 node ad hoc networks

Figure 13.a shows a fully-connected network in which all the traffic is directed to the base station. Figure 13.b is a multi-hop network of 16 nodes with each node having 4 neighbors [17]. The lines between the nodes show the connectivity between the nodes.

It is interesting to note the structural dependence on the requirement of the number of spreading codes for the other protocols. In case b, MACA-CT can be designed using a minimal of 11 data channels by taking advantage of spatial separation. For either situation, CHMA would still require as many channels as the maximal data-packet length. Both problems can be avoided by a dynamic allocation of channels.
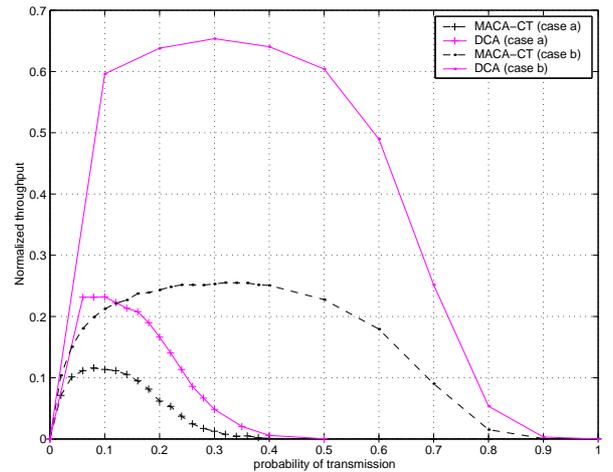

Fig. 14. Throughput comparisons for different scenarios

The parameters used in the simulations are identical to those

used previously. We consider 5 data-channels with one common control channel. Mean data-packet length is 10 slots, with a geometric distribution. Nodes have a single packet buffer. The network throughput is recorded with a constant probability of packet arrival (p).

As can be seen for case b, since the contention neighborhood is much smaller, the throughput of DCA is significantly greater that that for a fully connected network of the same size. Also the gains of DCA over MACA-CT are clearly visible.

Thus, in the context specified, DCA is superior to the other protocols and offers significant advantage. The penalty is the increased complexity of the receiver and the need for proper parameter selection. These could either be set apriori or kept variant, dependent on the network load.

VI. CONCLUSIONS

Medium access control is a critical issue in ad hoc networks. One of the biggest stumbling block remains the proper scheduling and reception of data-packets in the absence of a central controller. Contention of data-packets occurs at the receivers and hence, proper scheduling of data-packets requires the propogation of the contention information from the receivers to the transmitters. This is particularly interesting for multi-channel ad hoc networks since the contention information can also be used in channel allocation.

In multi-channel ad hoc networks, the channel assignment has conventionally been regarded as a separate issue and isolated from the MAC. The spreading gain and consequent loss in the data-rate are mostly overlooked.

Our objective here has been to propose a MAC protocol for multichannel ad hoc networks based on the feedback of channel contention at the receiver. A channel is selected for transmission only if it does not cause any contention at any of the receivers in the neighborhood. The protocol is proposed in section III.

The salient features of the protocol include the fact that channel allocation is included as a part of the MAC and the introduction of a feedback mechanism to propogate channel contention. This results, not only in a tighter reuse of channels over a multi-hop network but also makes the spreading gain independant of the size of the neighborhood.

We propose a novel method for the dynamic allocation of channels to nodes by means of querying the channel. Querying is a binary hypothesis detection and it is shown that the detection of the Query can be modeled in terms of a Neyman-Pearson detector. The success of the hypothesis is quantized in terms of two quantities based on the signal-to-noise ratio at the receiver: the probability of False-alarm, and Missed-detection of the Query.

The throughput of the protocol is analysed for a fully-connected network in section IV. Our analysis and simulations reveal that the network throughput is a convex function of the spreading codes, data-packet length and the probability of transmission. The operating threshold of the Query detection also has significant impact on the network throughput. Proper selection of network parameters is crucial in order to maximize the throughput.

Performance of the system for different parameters is analysed in section V. It is seen that for low noise conditions DCA is superior to other protocols. DCA also manages to reduce the dependence of the protocol on the network topology thus being more versatile.

Before we conclude, it is perhaps important to note that most of the losses in DCA are the result of improper Querying. We believe that the efficiency of the protocol can be further improved with the use of a 'smart', non-random channel selection policy as well as by optimizing the Data state detectors and the Query state detector to independent thresholds.

ACKNOWLEDGMENTS

VII. APPENDIX

A. Modifications to MACA-CT

The improvement in throughput in CHMA is due to the relocation of the CTS from the common channel to the transmitter assigned channel. Unfortunately, due to the relation between the maximum data-packet length and the hopping sequence length, it is not easy to calculate the normalized throughput of CHMA. The same technique can however be implemented without channel hopping. This may be considered as an extension of MACA-CT. We call this protocol Modified-CT (to acknowledge it as an extension of the Common-Transmitter protocols) and is introduced primarily to obtain an estimate on the maximum normalized throughput achievable by CHMA.

B. Modified-CT: The Protocol

Consider a time-slotted system with N nodes. Each node has a pre-assigned channel on which it transmits all the data-packets. Thus, there are N fixed data-channels. In addition, there is a common control channel. Any node that has a packet to transmit sends a RTS on the control channel. The RTS specifies the transmitter, the receiver and the transmitter-assigned channel. This part of the protocol is exactly identical to MACA-CT.
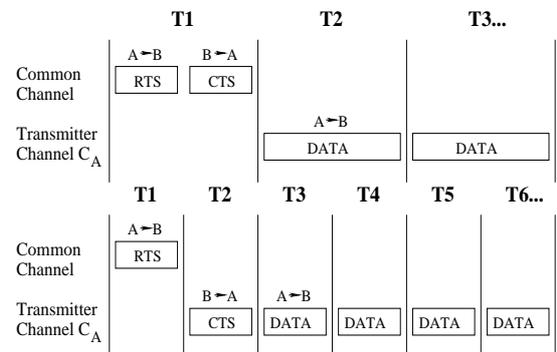

Fig. 15. Packet scheduling in (a) MACA-CT, and (b) Modified-CT

Since all the idle nodes are tuned to the common channel, if the RTS is received successfully by the intended receiver, it

sends a CTS to the source node over the transmitter assigned channel. This is the basic difference between MACA-CT and Modified-CT (Figure 15). At that time, the two given nodes will proceed to exchange data over the transmitter assigned channel. When the transmission of the data is completed, the sender and the receiver reset and tune back to the common channel.

If either multiple RTS are sent or the destination does not receive the RTS, no CTS is sent, and consequently the source node reverts back to idle. In the absence of detection errors, the CTS always succeeds. Since the channel chosen for transmission of the data-packet depends upon the transmitter and is not dependent on the slot number, the normalized throughput can be calculated for this case and is simply the network throughput divided by the total number of channels ($N + 1$).

C. Analysis of throughput for Modified-CT

The Modified-CT protocol is analysed for a single hop fully connected network under the same assumptions made in CHMA.
For any time-slot, the network can be described by,

- k: the number of nodes transmitting a RTS in the current mini-slot.
- l: the number of nodes that sent a RTS in the previous time-slot but failed the contention.
- m: the number of node-pairs communicating on the transmitter assigned channel. As seen from Fig 15, the packet will be either CTS or Data.

Given a network with N nodes, any combination of these parameters (k, l, m) completely describes the current *state* of the network. Also, let (w, x, y) represent identical parameters for the previous time-slot.

We assume that the length of the Data packet has a geometric distribution with a probability q of the Data transmission continuing to the next time-slot. Thus the length (D) of the packet is $P(D = d) = q^{(d-1)}(1 - q)$. Then the state in the next time-slot (w, x, y) would depend only on the current state (k, l, m) and the states form a Markov chain.

Let $\mathcal{T}$ represent the event that the transition from (k, l, m) to (w, x, y) occurs, $\mathcal{I}$ the event that exactly one RTS is sent (*i.e.* $k = 1$) and it is sent to an idle node, and $\mathcal{B}$ the event that exactly one RTS is sent (*i.e.* $k = 1$) but it is sent to a busy node. The transition probabilities for the state in the Markov chain can be computed as:

$$P_{klm,wxy} = P(w, x, y | k, l, m)$$
$$= \sum_{i=0}^{m} P\left(\begin{array}{c} \text{i Data pairs} \\ \text{become Idle} \end{array} \right) \left[P(\mathcal{T} \cup \mathcal{I}) + P(\mathcal{T} \cup \mathcal{B}) + P(\mathcal{T} \cup (k \neq 1)) \right]$$
$$= \sum_{i=0}^{m} B(m, 1 - q, i) \Big[\delta(m' - 1)\delta(x)\delta(k - 1).$$
$$B(N' - 1, p, w) \left(\frac{N''}{N - 1} \right) + \delta(m')\delta(x - 1).$$
$$\delta(k - 1)B(N', p, w) \left(\frac{N - N'' - 1}{N - 1} \right)$$
$$+ \delta(m')\delta(k - x)(1 - \delta(k - 1))B(N', p, w) \Big]$$
$$(22)$$

where, $B(m, 1-q, i)$ is the Binomial distribution and represents the probability that i out of the m data streams terminate, $N' = N - 2(m-i) - k$ is the number of nodes that are not transmitting or receiving at the end of the slot, $N'' = N - 2m - l - k$ is the number of idle nodes for the duration of the slot, and $m' = y - (m - i)$ is the number of new node pairs that start transmitting.

The chain is aperiodic and irreducible, thus a steady state distribution (S_{klm}) exists. Since the CTS is also transmitted on the data-channel, it needs to be subtracted from our computation of the average number of packets carried per slot. The network throughput is given by,

$$\Gamma = \sum_{k,l,m} m S_{klm} - \sum_{k=1,l,m} S_{klm} \left(\sum_{w,x=0,y} P_{klm,wxy} \right). \quad (23)$$

where the first term on the right hand side is the average number of packets carried over the data-channels, and the second term represents the average number of RTS successful in one time slot. Since for every successful RTS, the CTS is always successful, the difference denotes the raw data throughput. Also important to note is that the slot length for Modified-CT is one half that of MACA-CT.

TABLE II
THROUGHPUT OF MACA-CT, CHMA AND MODIFIED-CT FOR
NETWORKS OF DIFFERENT SIZE

	8	12	16	20
MACA-CT	1.7669	2.1521	2.4131	2.5981
CHMA	2.4148	3.2190	3.7832	4.3363
Modified-CT	2.4148	3.2190	3.7832	4.3363

Numerical values for the throughput of Modified-CT are compared against that of MACA-CT and CHMA for fully connected networks of different sizes and with a mean data-packet length that is 20 times the length of the RTS (Table II). It is seen that the network throughput of CHMA and Modified-CT are the same. This substantiates our claim that the normalized throughput of Modified-CT represents a limit on the performance acheivable by CHMA.

REFERENCES

[1] 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. Communications*, vol. COM-23, No. 12, pp. 1400-1416, Dec 1975.
[2] L. Kleinrock and F.A. Tobagi – "Packet switching in radio channels: Part II - The hidden terminal problem in carrier sense multiple access and the busy-tone solution", *IEEE Trans. Communications*, vol. COM-23, No. 12, pp. 1417-1433, Dec 1975.
[3] P. Karn, "MACA - a new channel access method for packet radio", in *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, pp. 134-40, ARRL, 1990.
[4] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A Media Access Protocol for Wireless LANs", in *Proc. ACM SIGCOMM 1994*, pp. 212-225, Sept. 1994.
[5] Z.J. Haas, Jing Deng and S. Tabrizi, " Collision-free medium access control scheme for ad hoc networks", in *Proc. IEEE MILCOM 1999*, Vol 1, pp. 276-280, 1999.
[6] C. L. Fullmer and J.J. Garcia-Luna-Aceves, "Floor Acquisition Multiple Access (FAMA) for Packet-Radio Networks", in *Proc. ACM SIGCOMM 1995*, Cambridge, MA, Sept. 1995.

[7] E.S. Sousa and J.A. Silvester, "Spreading code protocols for distributed spread-spectrum packet radio networks", *IEEE Trans. Communications*, vol 36, pp. 272-281, March 1988.

[8] M. Joa-Ng and I-Tai Lu, "Spread-spectrum medium access protocol with collision avoidance in mobile ad hoc wireless network", in *Proc. IEEE INFOCOM 1999*, vol. 2, pp. 776-783, 1999.

[9] A. Tzamaloukas and J.J. Garcia-Luna-Aceves, "Channel-hopping multiple access", in *Proc. IEEE ICC 2000*, vol. 1, pp. 415-419, 2000.

[10] Sanjay Lal and E.S. Sousa, "Distributed resource allocation for DS-CDMA-based multimedia ad hoc wireless LANs", *IEEE Journal on Selected Areas in Communications*, vol. 17, pp. 947-967, May 1999.

[11] Amit Butala, "Dynamic channel allocation for medium access control in spread-spectrum ad hoc networks", *Master's Dissertation, Cornell University*, August 2001.

[12] Amit Butala, Lang Tong, "Dynamic Channel Allocation and Optimal Detection for MAC in CDMA Ad hoc Networks", 36th Asilomar Conference on Signals, Systems and Computers, November 2002.

[13] Vincent Poor, "An introduction to Signal Detection and Estimation", Springer-Verlag, 1994.

[14] Sergio Verdu, "Multiuser Detection", Cambridge University Press, 1998.

[15] Imrich Chlamtac and Andras Farago, "An optimal Channel access protocol with multiple reception capability", IEEE Trans. Computers vol. 43, pp480-484, April 1994.

[16] J.J. Garcia-Luna-Aceves and J. Raju, "Distributed Assignment of Codes for Multihop Packet-Radio Networks", in *Proc. IEEE MILCOM 1997*, vol. 1, pp450-454, 1997.

[17] A. Tzamaloukas and J.J. Garcia-Luna-Aceves, "A receiver-initiated collision-avoidance protocol for multi-channel networks", in *Proc. IEEE INFOCOM 2001*, vol. 1, pp189-198, 2001.

[18] E.S. Sousa, "Interference modeling in a direct-sequence spread-spectrum packet radio network", *IEEE Trans. Communications*, Vol. 38, Issue 9, pp1475-1482, Sept. 1990.

[19] *IEEE P802.11 Draft Standard for Wireless LAN: Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE, July 1996.

[20] Kyungwhoon Cheun, "Optimum arrival-time distribution for delay capture in spread-spectrum packet radio networks", *IEEE Trans. Vehicular Technology*, vol. 46, Issue 4, pp981-991, Nov. 1997.