Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008

Controlled Random Access MAC for Network Utility Maximization in Wireless Networks

Robert J. McCabe, Nikolaos M. Freris, and P. R. Kumar

Abstract— **There has been much recent interest in protocol design for wireless networks based on maximizing a network utility function. A significant advance in recent years is the observation that a decomposition of the Lagrangian suggests an approach where transmissions are scheduled to minimize backpressure. However, a satisfactory Medium Access Control (MAC) protocol that can realize such a scheduling algorithm is notably missing, and that is the goal of this paper.**

We present a candidate random access MAC protocol that extends an existing algorithm to calculate the access probabilities. We also consider the online adaptation of access probabilities using local information about queue lengths and active links. In addition, we also modify the backpressure algorithm itself, by incorporating a minimum hop bias to alleviate the inherent problem of routing loops.

We have implemented a general purpose simulation framework to study the comparative performance of network management protocols for congestion control, routing, MAC, and their cross-layer interaction. Using this, we compare the performance of our scheme with the leading schemes.

I. Introduction

In a wireless ad hoc network, nodes forward packets in a multi-hop fashion to deliver the data to the desired destinations. The decentralized nature of ad hoc networks makes them suitable for applications where there exists no centralized coordinating node and where minimal configuration and quick deployment are desired. The stage for the movement towards wireless ad hoc networks was set by technologies such as IEEE 802.11. Currently, wireless ad hoc protocol design follows the layered architecture of traditional wireline networks which may be suboptimal in the realm of wireless ad hoc networks. The complex and unpredictable nature of the wireless medium make the management of wireless ad hoc networks difficult, and substantially different from that of wireline networks.

The employment of network utility maximization (NUM) [1] provides mathematical guidance in the design of network algorithms. In this approach, users determine a utility function, and the network is responsible to choose their rates so that the sum of the user utilities is maximized subject to maintaining stability of the network. In recent years NUM has also been studied for wireless networks. A significant insight emerging from a formulation of the NUM problem [2], is that the dual decomposition suggests a *backpressure* scheduling and routing policy in which packets are routed based on the queue length information of a node's neighbors.

One major challenge for the utility of the NUM approach is that many of the algorithms suggested by the NUM framework require global information, and thus may not be implementable in an actual protocol. The backpressure policy assumes that each node has instantaneous access to its neighbors' queue lengths, which practically requires an additional message passing protocol. The design of a distributed Medium Access Control (MAC) protocol is another concern in the context of the NUM framework. The NUM solution does suggest a scheduling policy [2]; however the policy requires global network coordinating functions and constitutes a difficult combinatorial problem in its own right. The ultimate goal is, therefore, the development of a low overhead scheduling algorithm that achieves the maximum throughput [5]. Clearly though, a distributed scheduling algorithm would yield inferior performance compared to the globally optimal solution, and it is of interest to characterize the tradeoff between the required message passing and the reduction in network throughput.

Inspired by and following [6], we design a MAC protocol, using random access embedded in an RTS-CTS-DATA-ACK handshake, to maximize a utility that is a weighted sum of logarithms of success probabilities for individual flows. This protocol is designed to improve performance over a pure random access mechanism by employing a four phase handshake. Our protocol is intended to be a component of a protocol design using the NUM framework. We conduct a comparative performance evaluation using a general purpose network simulation package called OPNET [7]. The aim is to investigate the possible gains from NUM in actual networks and to motivate the need for further research in NUM from a practical standpoint. We also discuss issues related to backpressure based routing [2] and propose a hybrid backpressure policy that achieves the same theoretical performance.

II. Related Literature

A. Network Utility Maximization

The wireless network is assumed to serve a set S of users. For each user $s \in S$, let f_s and d_s be the source and destination nodes of flow s, respectively. Each user $s \in S$ has a utility function $U_s(x_s)$ which represents the reward of user s if the network allows it to send data at a rate x_s. It is assumed that x_s is bounded in $[0, M_s]$ and that $U_s(\cdot)$ is strictly concave and increasing for each $s \in S$. We say that

This material is based upon work partially supported by Rockwell-Collins under Contract RPS#2 4502282787. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the above agency.

Robert McCabe, Nikolaos M. Freris, and P. R. Kumar are with the Dept. of ECE and CSL, University of Illinois at Urbana-Champaign, 1308 W. Main Street, Urbana, IL 61801, USA. Email: rmccabe2@uiuc.edu, nfreris2@uiuc.edu, prkumar@uiuc.edu.

978-1-4244-3124-3/08/$25.00 ©2008 IEEE

a particular rate allocation $\vec{x} = [x_s, s \in S]$ is feasible, if there exists a scheduling policy that realizes that throughput allocation. We define the *capacity region* Λ as the set of such feasible rate vectors $\vec{x}$.

The network utility maximization problem is

$$\begin{array}{ll} \max_{x_s \le M_s} & \sum_{s \in S} U_s(x_s). \\ \text{s.t.} & x \in \Lambda \end{array} \quad (1)$$

A *node-centric* characterization of Λ was studied in [3]. The node-centric formulation imposes the constraint that, at every node, the outgoing rate of a given flow must be at least as large as the incoming flow rate, plus the rate of the flow originating from that node. We consider the abstract network model where $\mathcal{N}$ is the set of nodes and $\mathcal{L}$ is the set of links such that if $(i,j) \in \mathcal{L}$ then a transmission from node i to node j is permitted. Let $\vec{r} = [r_{ij}, (i,j) \in \mathcal{L}]$ denote a feasible vector of link rates at a given time instant, and let $\mathcal{R}$ denote the set of all possible link rate vectors. Any rate vector in $\mathcal{C}o(\mathcal{R})$ can be achieved by time-sharing between the transmission rates $\vec{r} \in \mathcal{R}$. A rate vector $\vec{x}$ is in the *node-centric* characterization of Λ if there exists a rate vector $\vec{r}^d$ associated with each destination node d, which satisfies

$$\begin{array}{l} r_{ij}^d \ \ge 0 \text{ for all } (i,j) \in \mathcal{L} \text{ and for all } d \\ \sum_{j:(i,j)\in\mathcal{L}} r_{ij}^d - \sum_{j:(j,i)\in\mathcal{L}} r_{ji}^d - \sum_{s: f_s = i, d_s = d} x_s \ge 0, \\ \quad \text{for all } d \text{ and all } i \ne d \\ \left[\sum_d r_{ij}^d : (i,j) \in \mathcal{L} \right] \in \mathcal{C}o(\mathcal{R}). \end{array} \quad (2)$$

This constraint set makes (1) a convex optimization problem, which can be shown to have no duality gap [2]. By assigning a dual variable q_i^d for each $i \in \mathcal{N}$ and each destination d, to the linear flow balance constraint in (2), the following dual algorithm is obtained in [2]:

$$x_s(t) = \arg \max_{0 \le x_s \le M_s} \left[U_s(x_s) - x_s q_{f_s}^{d_s} \right], \quad (3)$$

$$\vec{r}(t) = \arg \max_{\vec{r} \in \mathcal{R}} \sum_{(i,j)\in\mathcal{L}} r_{ij} \max_d \left(q_i^d - q_j^d \right), \quad (4)$$

$$q_i^d(t+1) = \left\{ q_i^d(t) - h_t \left[\sum_{j:(i,j)\in\mathcal{L}} r_{ij}^d(t) - \sum_{j:(j,i)\in\mathcal{L}} r_{ji}^d(t) - \sum_{s: f_s = i, d_s = d} x_s(t) \right] \right\}^+, \quad (5)$$

where $\{h_t\}$ is a sequence of positive step sizes and $\{x\}^+ := \max\{x, 0\}$.

The real power of the node-centric formulation is in the subgradient update [8] of the dual objective function (5). Suppose that each node $i \in \mathcal{N}$ has a queue Q_i^d for each destination d, then its queue lengths would evolve according to the equation:

$$Q_i^d(t+1) = \left\{ Q_i^d(t) - \left[\sum_{j:(i,j)\in\mathcal{L}} r_{ij}^d(t) - \sum_{j:(j,i)\in\mathcal{L}} r_{ji}^d(t) - \sum_{s: f_s = i, d_s = d} x_s(t) \right] \right\}^+. \quad (6)$$

The critical observation in [2] is that this is identical to (5) apart from the step-size parameter. Thus, the dual variables in the node-centric formulation have a direct physical interpretation in terms of queue lengths, and can therefore be measured locally at each node.

Equation (3) can be thought of as a congestion controller where each source adapts its injection rate based on its current utility as well as its own queue backlog. Note that (3) is a *local* algorithm because it requires only local queue length information at each source. Moreover, the scheduling component of the algorithm is taken care of by (4) where the link rates are allocated based on maximizing the sum of the rate-weighted backlogs. This algorithm is reminiscent of the backpressure scheduling policy [4], which is proved in [3] to have the largest stability region:

Backpressure Policy Property: If $\vec{x}$ is a vector of rates in the interior of the capacity region Λ, i.e., there exists a policy that stabilizes the queue lengths in the network subject to $\vec{x}$, then the backpressure routing policy will also stabilize the network queues. In addition, the backpressure policy achieves the maximum throughput among all stabilizing policies, i.e., is *throughput-optimal*.

III. An optimal design for a Random-Access MAC

We now discuss the complexity of the policy suggested by the dual decomposition, present our network model, and develop a theoretical framework for an optimal calculation of the access probabilities in random-access Media Access Control (MAC) protocols. Our results generalize and expand the analysis in [6], to account for an exchange of RTS and CTS packets among different nodes.

A. Protocol Design Considerations

The scheduling component (4) of the dual algorithm described in the previous section has several drawbacks from a practical standpoint. First, the optimization assumes *global* knowledge of the system's state, i.e., queue lengths and the set of rate allocation vectors $\mathcal{R}$. Even with this knowledge, solving (4) at every iteration is a difficult optimization problem. For example, consider the special case where $\mathcal{R} \subset \{0, R_{max}\}^{|\mathcal{L}|}$ and is characterized by the condition that any feasible rate vector $\vec{r} \in \mathcal{R}$ should satisfy $r_{ij} = 0$ if either $r_{jl} = R_{max}$ for some node l such that $(j,l) \in \mathcal{L}$, or if $r_{km} = R_{max}$ for some node m such that $(k,m) \in \mathcal{L}$ and $(k,j) \in \mathcal{L}$. In this case, (4) is a maximum weighted independent set problem and its distributed solution is an active area of research [9], [10]. Thus, the challenge is to design a distributed, low complexity scheduler that approximates (4).

To make it more tractable, our design is based on the following choices:

1) The protocol should be compatible with the IEEE 802.11 physical layer technology. IEEE 802.11 network cards employ the Carrier Sense Multiple Access (CSMA) technique for packet transmission. This serves to significantly reduce the complexity of the set $\mathcal{R}$ because a link rate vector is feasible only if the transmitter and receiver of any active link are the only active nodes among their respective one hop neighbors.
2) Each node in the network uses a common transmit power. This reduces the scheduling complexity by facilitating reservation techniques such as the RTS/CTS mechanism in IEEE 802.11 whose operation assumes link symmetry.
3) The transmission rates of all nodes are assumed to be fixed at R^* since the decomposition solution does not

provide much insight into making link rate adaptation tractable.

With these simplifications, the scheduling policy (4) can be written as

$$\vec{r}(t) = \arg \max_{\vec{r} \in \{0, R^*\} \cap \Gamma} \sum_{(i,j) \in \mathcal{L}} r_{ij} \max_d \left(Q_i^d(t) - Q_j^d(t) \right), \quad (7)$$

where Γ represents the set of feasible transmission modes, as shown in Figure 1. The quantity $\max_d \left(Q_i^d(t) - Q_j^d(t) \right)$ is called the *backpressure* on link (i, j).

While the backpressure algorithm (7) is an optimal policy in terms of the stabilization of the largest set of flow rates, it is not a desirable algorithm in actual networks. To illustrate this, consider a network in which a node wishes to send *one* bit to a node located multiple hops away. If the network employs the backpressure policy, there is no guarantee that the bit will ever reach its destination because, at any time, the backlog of every node in the network will be either 0 or 1. Thus, backpressure routing has an undesirable behavior when the network is not operating *near capacity*.

We therefore consider the following hybrid policy which is similar to the Enhanced DRPC policy in [3].

Hybrid Routing Policy:

$$\vec{r}(t) = \arg \max_{\vec{r} \in \{0, R^*\} \cap \Gamma} \sum_{(i,j) \in \mathcal{L}} w_{ij}(t), \quad (8)$$

$$w_{ij}(t) := r_{ij} \max_d \left\{ H(i,j;d) + \alpha \left(Q_i^d(t) - Q_j^d(t) \right) \right\}, \quad (9)$$

where $0 < \alpha < 1$ and $H(i,j;d) = 1$, if node j lies upon a shortest path from node i to destination d, and equals 0 otherwise. The policy (9) provides a bias towards routing packets to their destination. Moreover, the behavior of this hybrid policy converges to the standard backpressure policy when the network is highly loaded because then the backpressure term dominates the minimum hop bias.

Thus, the goal of our distributed scheduling algorithm to find the transmission mode (see Fig. 1) that maximizes (8) at every time instant. Even with our simplifications, however, solving (8) in a distributed manner is a daunting task. Thus, following [6], we design protocols that *approximate* the policy in (8) using a random access scheme. Next, we provide a distributed algorithm for the selection of access probabilities given the link weights defined in (9).

B. Network Model

Let $\mathcal{N}$ denote the set of nodes. For each node n, let $\mathcal{D}_n \subseteq \mathcal{N} \setminus n$ denote the *decoding set* of n, i.e., the set of nodes that can reliably decode transmissions from n. Let $\mathcal{I}_n$ denote the *interference set* of n, i.e., the set of nodes that are interfered by transmissions from n. We assume that $\mathcal{D}_n \subseteq \mathcal{I}_n$, and $n \in \mathcal{I}_n$. A transmission from n to m is successful if and only if (i) $m \in \mathcal{D}_n$ and (ii) $m \notin \mathcal{I}_k$ for any other node k that is transmitting in the same slot. This formulation does not consider a physical model based on signal-to-interference plus noise ratio.

We define the link set $\mathcal{L} := \{(i,j) : j \in \mathcal{D}_i\}$. We further assume that all links are bidirectional, i.e., $(i,j) \in \mathcal{L} \iff (j,i) \in \mathcal{L}$, or equivalently $m \in \mathcal{D}_n \iff n \in \mathcal{D}_m$. We make this assumption since we will study protocols based on the exchange of RTS and CTS packets.

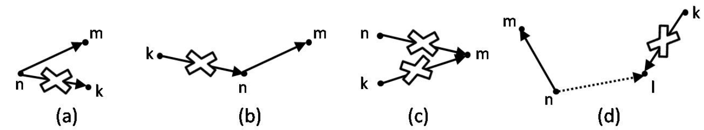

Fig. 1. Transmission modes (solid lines indicate active transmissions, dashed lines indicate interference, and crosses indicate collisions): (a) No parallel transmissions, (b) No simultaneous transmission & reception, (c) Collision of transmissions to the same node, (d) Interference caused by a transmission to another node (hidden node problem)

C. A Random Access Protocol

Let us consider the following slotted random access mechanism. At any time t, a node may attempt to transmit one unit of data to a node $m \in \mathcal{D}_n$ with some probability p_{nm}. Let

$$p_n := \sum_{m \in \mathcal{D}_n} p_{nm}, \quad (10)$$

denote the probability that n transmits in the time slot.

We consider the following mechanism for packet transmission, which is in the IEEE 802.11 protocol. When node n wants to transmit a packet to a node m, it first contends for the channel by sending a Request-To-Send (RTS) packet to m. Upon successful receipt of the RTS, node m responds to n by sending a Clear-To-Send (CTS) packet. Then, if the CTS is successfully received by n, it sends the data packet (DATA) containing the message to be delivered to m. Finally, once m gets the DATA packet, it sends an Acknowledgment (ACK) back to the sender n. The transmission of information from node n to node m is successful if and only if all four packets RTS, CTS, DATA, and ACK are delivered reliably. We use the term *four phase handshake* to refer to the exchange of these four packets (RTS, CTS, DATA, ACK).

Both the RTS and the CTS packets contain information about the identities of the transmitter-receiver pair (n, m) and the duration of the ensuing transmission. All nodes $i \in \mathcal{D}_n \backslash n$ that can decode the RTS will refrain from transmitting for the duration of the entire four phase handshake, so as to avoid interfering with the ongoing data transmission over the link (n, m). Similarly, all nodes $j \in \mathcal{D}_m \setminus m$ that can decode the CTS are required to silence themselves for the remainder of the four phase handshake. If, on the other hand, a node in $(\mathcal{I}_n \setminus \mathcal{D}_n) \cup (\mathcal{I}_m \setminus \mathcal{D}_m)$ wishes to send a packet, then it is required to refrain from initiating a transmission for as long as it senses interference, plus an additional amount of time, called the Extended Inter-Frame Space (EIFS).

We will denote the duration of an RTS packet, measured in slots, by c_r, the duration of a CTS packet by c_c, and the duration of the DATA packet, which will be assumed to be of fixed size, by c_d. The duration of the ACK is also equal to c_c, while the duration of EIFS is equal to c_r. In practice, we have $c_c < c_r < c_d$. For example, for a transmission at 1 Mbps, the duration of a slot is 9 μs, $c_r = 40$ slots, $c_c = 34$ slots and c_d varies, but is typically greater than 200 slots.

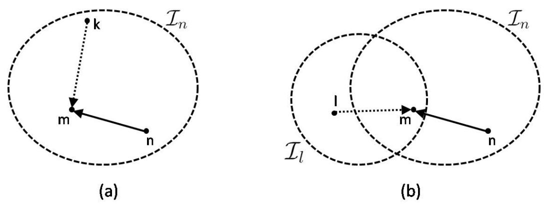

Fig. 2. Interference scenarios in a transmission

D. Optimization problem

We seek to extend the work in [6] to take advantage of the four phase handshake in achieving coordination. As in [6], we wish to determine the vector of access probabilities $p := \{p_{nm}\}_{(n,m)\in\mathcal{L}}$, as the solution to the following optimization problem:

$$\begin{array}{lll} \max_p & F(p) := \sum_{(n,m)\in\mathcal{L}} w_{nm} \log \mu_{nm} & \\ \text{s.t.} & p_{nm} \geq 0 & \text{for all } (n,m) \in \mathcal{L} \\ & \sum_{m\in\mathcal{D}_n} p_{nm} \leq 1 & \text{for all } n \in \mathcal{N}. \end{array} \tag{11}$$

Here μ_{nm} is intended to represent the probability of a successful transmission over the link (n,m), and $w_{nm} \geq 0$, are arbitrary fixed weights. In fact, we only have $w_{nm} = 0$ if data transmission on the link $(n,m) \in \mathcal{L}$ is not allowed, i.e., $p_{nm} := 0$. The quantity $w_{nm} \log \mu_{nm}$ models the utility provided to link (n,m). Note that this weighted log objective function corresponds to weighted proportional fairness for the success probabilities μ_{nm}. The objective function of problem (11) could incorporate the objective (8) if the weights w_{nm} are adapted according to (9).

In [6], RTS, CTS and ACK packets are not used, and all nodes just contend randomly, with access probabilities p_{nm} for link (n,m), using their DATA packets that can then be simply considered as one slot long. In that special case,

$$\mu_{nm} = p_{nm} \prod_{k:m\in\mathcal{I}_k, k\neq n} (1 - p_k), \tag{12}$$

since a transmission is successful only if there are no other simultaneous interfering transmissions. The optimal solution to this special case of the optimization problem is calculated in [6] to be

$$p_{nm} = \frac{w_{nm}}{\sum_{(l,k)\in\mathcal{S}_n} w_{lk}}, \text{ for all } (n,m) \in \mathcal{L}, \tag{13}$$

where

$$\mathcal{S}_n := \{(i,j) \in \mathcal{L} \mid j \in \mathcal{I}_n\}. \tag{14}$$

One issue with the approach taken in [6] is that every data packet is always in contention with other transmissions. Hence there is no attempt to reserve the channel using a smaller control packet, as is done in IEEE 802.11, which is more efficient in terms of bandwidth as opposed to gambling with large DATA packets. In this paper, motivated by the goal of providing better performance we employ RTS, CTS, DATA, and ACK packets and consider the more general problem described earlier. According to the standard IEEE 802.11, if a node wishes to send, it will randomly choose a *back-off* value in the range $\{0,\ldots,31\}$ (more generally $\{0,\ldots,2^k - 1\}$) and send its RTS when the appropriate number of slots have elapsed. If the node does not receive a CTS or ACK, then it will contend for the channel again. Every time this occurs, it will randomly choose a back-off time in a range that is twice as large as the one in its previous attempt. This mechanism is called binary exponential back-off and is an attempt to mitigate congestion if no explicit information about the network congestion is available.

Our goal is to develop a MAC protocol which uses link weights to calculate the access probabilities. Let p_{nm} denote the probability that node n attempts to send an RTS to node m in a given slot. In a collocated network, i.e., in a network where all nodes can communicate with one another, the probability of a successful transmission is given by equation (12), due to the carrier sensing mechanism of IEEE 802.11. In general networks, μ_{nm}, the probability of a successful transmission of an RTS packet from node n to node m, is therefore (see also Figure 2):

$$\mu_{nm} = p_{nm} \prod_{k\in\mathcal{I}_n\setminus n} (1 - p_k) \prod_{l:m\in\mathcal{I}_l, l\notin\mathcal{I}_n} (1 - p_l)^{c_r}. \tag{15}$$

The reasoning is as follows. Due to the carrier sensing mechanism, a node will defer transmission when it senses the channel busy. Thus, a successful transmission from node n to node m requires that all nodes in the interference range of node n be silent for one time slot so that n has the opportunity to acquire the channel. This gives rise to the first product term in (15). However, the nodes that do not belong to $\mathcal{I}_n$, but cause interference to the receiver m (see also Figure 2.(b)), need to remain silent for the entire duration of the RTS transmission, i.e., for c_r time slots. Hence the second product in (15).

The solution to (11) is given by the following theorem. We call the random access scheme implementing the rule (16) below as *Utility Maximizing MAC* (UMAC).

Theorem 1: For an arbitrary set of positive[1] weights $\{w_{nm}\}_{(n,m)\in\mathcal{L}}$, the unique vector $p = (p_{nm})_{(n,m)\in\mathcal{L}}$ attaining the maximum in (11) is given by

$$p_{nm} = \frac{w_{nm}}{\sum_{k\in\mathcal{D}_n} w_{nk} + \sum_{(l,k)\in\mathcal{S}_n^{(1)}} w_{lk} + c_r \sum_{(l,k)\in\mathcal{S}_n^{(2)}} w_{lk}}, \tag{16}$$

where

$$\mathcal{S}_n^{(1)} := \{(i,j) \in \mathcal{L} \mid i \in \mathcal{I}_n \setminus n\}, \tag{17}$$

$$\mathcal{S}_n^{(2)} := \{(i,j) \in \mathcal{L} \mid j \in \mathcal{I}_n, n \notin \mathcal{I}_i\}. \tag{18}$$

Proof: Substituting (15) into the objective function $F(p)$ of (11), and using (17), (18), we get

$$\begin{array}{rcl} F(p) & = & \sum_{(n,m)\in\mathcal{L}} w_{nm} \log p_{nm} \\ & + & \sum_{n\in\mathcal{N}} \log(1 - p_n) \sum_{(l,k)\in\mathcal{S}_n^{(1)}} w_{lk} \\ & + & \sum_{n\in\mathcal{N}} \log(1 - p_n) \sum_{(l,k)\in\mathcal{S}_n^{(2)}} c_r w_{lk}. \end{array} \tag{19}$$

Note that $F(p)$ is a strictly concave function in p. Differentiating with respect to the decision variables $\{p_{nm}\}$ and setting the partial derivatives equal to 0 yields:

[1]We may assume, without loss of generality, that all weights are positive, since we set $p_{nm} = 0$ whenever $w_{nm} = 0$.

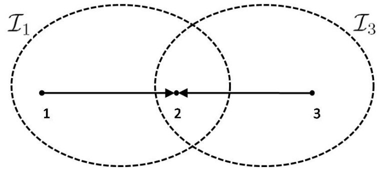

Fig. 3. Hidden node problem

$$\frac{\partial F}{\partial p_{nm}} = \frac{w_{nm}}{p_{nm}} - \frac{\sum_{(l,k)\in\mathcal{S}_n^{(1)}} w_{lk}}{1-p_n} - \frac{c_r \sum_{(l,k)\in\mathcal{S}_n^{(2)}} w_{lk}}{1-p_n}. \tag{20}$$

For each $n \in \mathcal{N}$ there are now two cases. If $\mathcal{S}_n^{(1)} \cup \mathcal{S}_n^{(2)} \neq \emptyset$, (20) yields

$$p_{nm} = (1-p_n)\frac{w_{nm}}{\sum_{(l,k)\in\mathcal{S}_n^{(1)}} w_{lk} + c_r \sum_{(l,k)\in\mathcal{S}_n^{(2)}} w_{lk}}. \tag{21}$$

Using (10), we obtain (16). In the case that $\mathcal{S}_n^{(1)} \cup \mathcal{S}_n^{(2)} = \emptyset$, optimizing $F(p)$ in (19) with respect to p_{nm}, for $m \in \mathcal{D}_n$, results in the following optimization problem

$$\begin{array}{lll} \max_{\{p_{nm}\}} & \sum_{m\in\mathcal{D}_n} w_{nm}\log p_{nm} & \\ \text{s.t.} & p_{nm} \geq 0 & \text{for all } m \in \mathcal{D}_n \\ & \sum_{m\in\mathcal{D}_n} p_{nm} \leq 1. & \end{array} \tag{22}$$

The solution to this constrained optimization problem can be solved using the Karush-Kuhn-Tucker (KKT) conditions:

$$p_{nm} = \frac{w_{nm}}{\sum_{k\in\mathcal{D}_n} w_{nk}}, \tag{23}$$

which is still consistent with (16), because in this case $\mathcal{S}_n^{(1)} = \mathcal{S}_n^{(2)} = \emptyset$. Since $F(p)$ is strictly concave, the global maximizer p is unique. ■

Example: Consider the case of the simple *hidden node problem* shown in Figure 3. The optimal access probabilities are

$$p_{12} = \frac{w_{12}}{w_{12} + c_r w_{32}}, \qquad p_{32} = \frac{w_{32}}{w_{32} + c_r w_{12}}. \tag{24}$$

As noted earlier, [6] considers the case $c_r = 1$. In our case, having $c_r > 1$ implies smaller access probabilities, i.e., a more conservative design for the hidden node problem. More generally, we could consider c_r as a design parameter that can be used to tune the performance of the resulting protocol. We will see in Section IV that our generalization is beneficial since it increases both performance and stability.

IV. Simulation Results

In this section, we present simulation results of our MAC and backpressure protocols.

A. Hybrid backpressure compared to pure backpressure

We first compare the performance of the hybrid backpressure policy (8)-(9) with $\alpha = 0.01$ against the *pure* backpressure policy (7). The simulated system consists of a random network topology with twenty nodes and two flows, each with an average traffic injection rate of 200 Kbps. The scheduling is implemented using a global deterministic optimization algorithm to solve both (7) and (8). The throughput performance of the hybrid policy is superior to that of the pure backpressure policy, as depicted in Figure 4.

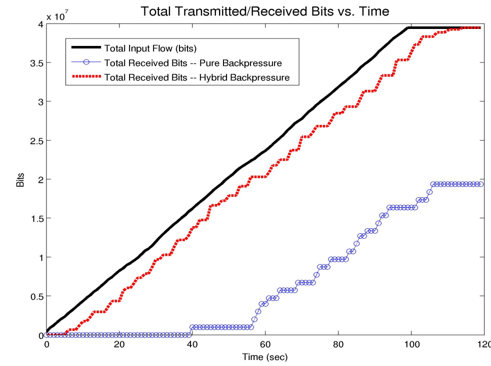

Fig. 4. Cumulative input flow compared against the received flow for both backpressure schemes.

B. Utility Maximizing MAC versus IEEE 802.11

Next, we simulated the performance of UMAC, whose access probabilities are given by equation (16), against IEEE 802.11, in the hidden node network shown in Figure 3. In the simulation, both nodes 1 and 3 attempt to send at 300 Kbps to node 2. Interestingly, the standard 802.11 significantly outperformed UMAC, in terms of received throughput. It turns out that this is an artifact of the lack of the binary exponential back-off mechanism in our protocol.

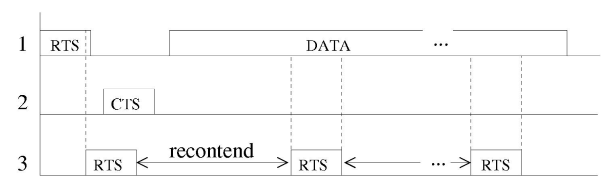

Fig. 5. Hidden node scenario causing poor performance of UMAC.

Examining the OPNET simulation log files, the poor performance of UMAC was due to scenarios similar to the one shown in Figure 5, where node 3 starts sending its RTS just before node 2 begins to send its CTS to node 1. If the SNR at node 2 is high enough, i.e., if node 3's transmission does not corrupt node 1's transmission, node 2 will be able to receive node 1's RTS, and will send the CTS back. However, since node 3 is busy sending its RTS it cannot hear the CTS; so it is not aware of the data session between nodes 1 and 2. In the case of IEEE 802.11, this situation is mitigated somewhat by the exponential back-off mechanism. Every time node 3 does not receive a CTS in response to its RTS, it doubles its congestion window, so there is less of a chance that node 3 will corrupt the session between nodes 1 and 2. With UMAC, however, node 3 recalculates its access probabilities whenever there is data in its queue, independent of the number of RTS attempts. This results in a larger chance that the DATA packet will be corrupted, as shown in Figure 5. A simple fix to UMAC is to forbid the receiver of a RTS to send back a CTS if the channel is sensed busy during a predefined interval after the RTS reception, called SIFS. This allows the receiver of the RTS to avoid a potential collision of the DATA packet. The rest

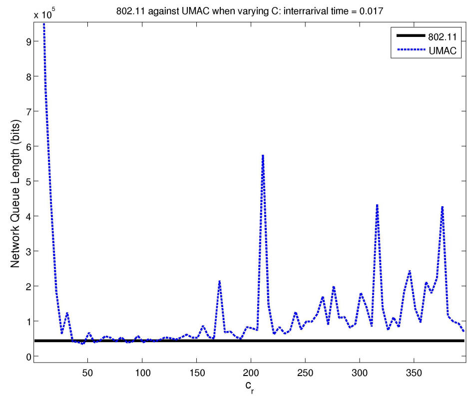

Fig. 6. Effects of varying c_r on the total network queue lengths.

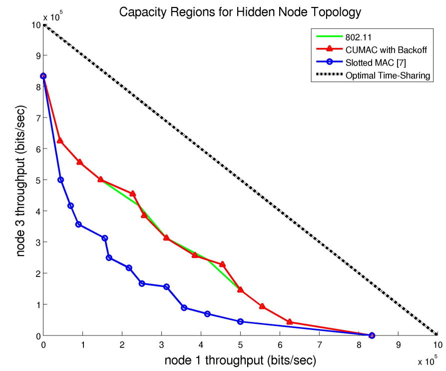

Fig. 7. Capacity region of the different protocols for the network topology in Figure 3.

of the simulations in this section use UMAC with this fix, which we will refer to as *Controlled UMAC* (CUMAC).

C. Performance of CUMAC with c_r versus IEEE 802.11

Next, we simulate the performance of CUMAC as the parameter c_r in (16) varies. We use the topology shown in Fig. 3 using exogenous flows to nodes 1 and 3 with exponentially distributed interarrival times and packet sizes. The mean packet size is 5000 bits, while the mean interarrival times are varied to realize several average flow rates. Fig. 6 illustrates the effect of varying c_r on the network queue lengths for an average packet interarrival time of 17 ms. This effect was common for various other interarrival times that were tested, too. The queue blowup for small values of c_r demonstrates the importance of Theorem 1 in choosing the access probabilities. The performance of IEEE 802.11 under the same network load is shown with a solid line in Figure 6. From the simulations, we observe that selecting c_r in the range $[40, 80]$ yields similar performance to IEEE 802.11 for several tested topologies and various flow rates.

D. Capacity regions for the hidden node topology

To illustrate the performance of the various MAC protocols we plot the capacity region in Figure 7. To generate this plot we varied the rates of nodes 1 and 3 until network instability was detected. Note that the algorithm [6] is not an actual protocol since each transmitter is assumed to somehow know if its transmission was successful without receiving an explicit acknowledgment from its receiver. So, in a sense, comparing the performance of IEEE 802.11 and CUMAC with the slotted MAC algorithm in [6] is unfair because both IEEE 802.11 and CUMAC involve extra overhead for ACKs along with the RTS/CTS signaling. Even with this idealistic setting, however, the algorithm in [6] is significantly outperformed by IEEE 802.11 and CUMAC. This can be attributed to the bandwidth savings of the RTS/CTS mechanism.

V. Conclusions and Future Work

We have considered the design of a MAC protocol in compliance with the *node-centric* formulation of the NUM problem. We have considered a hybrid backpressure policy (8)-(9) designed to deliver nearly the same capacity region as the throughput optimal policy (7), but with superior performance when the network is not operating near capacity. Also, we have designed the CUMAC protocol which is compatible with the backpressure scheduling scheme. CUMAC effectively utilizes the backpressure weights to calculate access probabilities while simultaneously incorporating an extension to the formulation in [6] to include the effects of hidden nodes in a CSMA wireless network.

It is of interest to develop a message passing protocol to provide the link weights needed for the calculation of the access probabilities at each node, and analyze its performance. Also of interest is the interaction between backpressure based routing schemes and CUMAC.

References

[1] F. P. Kelly, A. Maulloo, and D. Tan. “Rate control in communication networks: shadow prices, proportional fairness and stability.” Journal of the Operational Research Society, 49:237–252, 1998.

[2] X. Lin and N. B. Shroff, “Joint Rate Control and Scheduling in Multihop Wireless Networks,” Proceedings of the IEEE Conference on Decision and Control, Paradise Island, Bahamas, December 2004.

[3] M. J. Neely, E. Modiano, and C. E Rohrs, “Dynamic power allocation and routing for time varying wireless networks,” IEEE Journal on Selected Areas in Communications, vol. 23, no. 1, pp. 89-103, January 2005.

[4] L. Tassiulas and A. Ephremides, “Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks,” IEEE Trans. Autom. Control, pp. 1936-1948, December 1992.

[5] X. Lin, N. Shroff and R. Srikant, “A Tutorial on Cross-Layer Optimization in Wireless Networks,” IEEE Journal on Selected Areas in Communications, August 2006, pp. 1452-1463.

[6] P. Gupta, A.L. Stolyar, “Optimal Throughput Allocation in General Random Access Networks,” CISS 2006, Princeton, March 22-24.

[7] http://www.opnet.com/

[8] D. P. Bertsekas, “Nonlinear Programming,” Second edition. Athena Scientific, 1999.

[9] X. Wu, R. Srikant, “Regulated Maximal Matching: A Distributed Scheduling Algorithm for Multi-Hop Wireless Networks With Node-Exclusive Spectrum Sharing,” IEEE Conference on Decision and Control 2005, pp. 5342-5347.

[10] S. Sanghavi, L. Bui, R. Srikant. “Distributed link scheduling with constant overhead,” SIGMETRIC 2007, pp. 313-324.