Optimal MAC Design Based on Utility Maximization: Reverse and Forward Engineering

Jang-Won Lee, Mung Chiang, and A. Robert Calderbank
Electrical Engineering Department, Princeton University, Princeton, NJ 08544, USA

***Abstract*— This paper analyzes and designs medium access control (MAC) protocols for wireless ad-hoc networks through the network utility maximization (NUM) framework. We first reverse-engineer the current binary exponential backoff (BEB) protocol in the IEEE 802.11 standard through a non-cooperative game-theoretic model. This MAC protocol is shown to be implicitly maximizing, using a stochastic subgradient, a selfish local utility at each link in the form of expected net reward for successful transmission. While the existence of a Nash equilibrium can be established, neither convergence nor social welfare optimality is guaranteed due to the inadequate feedback mechanism in the BEB protocol. This motivates the forward-engineering part of the paper, where a network-wide utility maximization problem is formulated, using the collision and persistence probability model and aligning selfish utility with total social welfare. By adjusting the parameters in the utility objective functions of the NUM problem, we can also control the tradeoff between efficiency and fairness of radio resource allocation through a rigorous and systematic design. We develop two distributed algorithms to solve the MAC design NUM problem, which lead to random access protocols that have slightly more message passing overhead than the current BEB protocol, but significant potential for efficiency and fairness improvement. In addition to numerical experiments that illustrate the value of the NUM approach to the complexity-performance tradeoff in MAC design, we also provide readily-verifiable sufficient conditions under which convergence of the proposed algorithms to a global optimality of network utility can be guaranteed.**

Keywords: Wireless network, Ad-hoc network, Medium access control (MAC), Mathematical programming/optimization, Network utility maximization, Network control by pricing.

I. Introduction

A. Motivation

In wireless networks, if contentions among transmissions on different links are not appropriately controlled, a large number of collisions may occur, resulting in waste of resources such as bandwidth and energy, as well as loss of system efficiency and degradation of resource sharing fairness. The important issue of contention resolution in the wireless medium access control (MAC) protocol is studied in this paper in two complementary directions: reverse and forward-engineering.

To rigorously and systematically explore alternative designs of MAC protocols, we need a tool that incorporates both efficiency and fairness metrics in the most flexible way. In this paper we use the framework of Network Utility Maximization (NUM), which provides a rigorous method for analyzing MAC performance and designing MAC protocols. This approach complements the well-established queuing-theoretic stochastic analysis, with a focus on attaining optimality with respect to general utility objectives for long-lived flows.

In wireless networks especially the multi-hop ones, there may not be any central controller, thus we focus on random access MAC protocols that can be implemented in a distributed way. Currently, the Distributed Coordination Function (DCF) is the standardized MAC protocol in IEEE 802.11 [1]. However, it has been concluded by many researchers that DCF and its Binary Exponential Backoff (BEB) mechanism for contention resolution can be inefficient and unfair in face of location-dependent contentions (*e.g.*, [2]). Various new algorithms have been developed to tackle these issues (*e.g.*, [2], [3], [4], [5], [6], [7], [8], [9]).

To better understand the BEB protocol in wireless MAC, we pose the following question: are the distributed and selfish actions by each link in the BEB protocol in fact implicitly maximizing some local utility functions? We provide a non-cooperative game model for the BEB protocol, reverse-engineering the underlying utility function's form and establishing the existence of a Nash equilibrium. But due to the inadequate feedback mechanism in the BEB protocol, neither convergence nor social welfare optimality can be assured.

This motivates the need for forward-engineering: what kind of new distributed algorithms will be provably convergent to the global optimum of total network utility? After formulating a probabilistic-modeled NUM problem for wireless MAC, we develop optimal algorithms to solve the NUM problem, and these algorithms are then turned into random access MAC protocols. Through this design approach, optimality with respect to prescribed user utilities, which in turn determine protocol efficiency and fairness, is guaranteed.

B. Review: TCP Reverse Engineering and Basic NUM

Since the seminal paper by Kelly et. al. [10] in 1998, the basic NUM framework has been extensively studied for rate allocation and congestion control in wired networks (*e.g.*, [11], [12], [13]). Consider a communication network with L logical links, each with a fixed capacity of c_l bps, and S sources (*i.e.*, end users), each transmitting at a source rate of x_s bps. Each source s emits one flow, using a fixed set $L(s)$ of links in its path, and has a utility function $U_s(x_s)$. Each link l is shared by a set $S(l)$ of sources. NUM, in its basic version for wired networks, is the following problem of maximizing the network utility $\sum_s U_s(x_s)$, over the source rates $\mathbf{x}$, subject to linear flow constraints $\sum_{s \in S(l)} x_s \leq c_l$ for all links l:

$$\begin{array}{ll} \text{maximize} & \sum_s U_s(x_s) \\ \text{subject to} & \sum_{s \in S(l)} x_s \leq c_l, \ \ \forall l, \\ & \mathbf{x}^{min} \preceq \mathbf{x} \preceq \mathbf{x}^{max}. \end{array} \tag{1}$$

Making the standard assumption on the concavity of the utility functions, problem (1) is a simple concave maximization of decoupled terms under linear constraints, which has long been studied in optimization theory as a monotropic program [14]. Recent work in congestion control literature (*e.g.,* in [15]) has shown that TCP congestion control protocols can be thus reverse engineered: TCP variants are implicitly solving the basic NUM (1) for different utilities using dual decomposition.

In the NUM framework, there are two interpretations of each source's utility function. It can be interpreted as the level of satisfaction attained by a user as a function of resource allocation. Each user may have a different utility function depending on its type of service. By maximizing the network utility (*i.e.,* the sum of all user utilities), we maximize the social welfare of the system [16]. Efficiency of resource allocation algorithms can be measured by the achieved network utility. Utility functions can also be interpreted as the 'knobs' to control the tradeoff between efficiency and fairness. Different shapes of utility functions lead to different types of fairness. For example, a family of utility functions parameterized by $\alpha \geq 0$ is proposed in [12]:

$$U^{\alpha}(x) = \left\{ \begin{array}{ll} (1-\alpha)^{-1} x^{1-\alpha}, & \text{if } \alpha \neq 1 \\ \log x, & \text{otherwise} \end{array} \right. . \tag{2}$$

If we set $\alpha = 0$, NUM reduces to system throughput maximization. If $\alpha = 1$, proportional fairness among competing users is attained; if $\alpha = 2$, then harmonic mean fairness; and if $\alpha \to \infty$, then max-min fairness.

Recently, NUM has also been used to develop MAC protocols in wireless multi-hop networks in [2], [5], [6], [7], [8], [9]. The paper by Nandagopal et. al. [2] is the first one to extend Kelly's wired NUM model to wireless MAC design by proposing the contention graph construction. In [2], [5], [6], [7], [8], [9], MAC protocols for wireless multi-hop networks are developed by using NUM. In [7] and [8], [9], max-min fairness (*i.e.,* $\alpha \to \infty$ in (2)) and proportional fairness (*i.e.,* $\alpha = 1$ in (2)) are considered, respectively. The NUM framework for more general utility functions are studied in [2], [5], [6] through deterministic approximations. Extensive discussion of and comparison with all related work can be found in Sections IC, IV and V.

C. Summary of Results

We start with reverse-engineering the exiting BEB protocol, which has never been addressed in the literature before. Internet TCP/AQM protocols in the transport layer have recently been reverse-engineered (*e.g.,* [15]) as implicitly solving a basic NUM using different Lagrange multipliers or congestion prices. Even though TCP/AQM protocols were first designed without regard to global optimization, a reverse-engineering model provides a rigorous path towards understanding the equilibrium and dynamic properties of complicated interactions across sources and routers. In those models, the utility function of each source depends only on its data rate that can be directly controlled by the source itself and there are adequate feedback from the network. Hence, the TCP/AQM protocol can be modeled as an algorithm that converges to the globally optimal rate allocation by solving the basic NUM problem (1) and its Lagrange dual problem.

In contrast, in the BEB protocol, the utility of each link directly depends not only on its own transmission (*e.g.,* persistence probability) but also transmissions of other links through collisions that cannot be controlled by the link itself. Moreover, there is no explicit feedback from the network. Hence, a non-cooperative game model is more appropriate for the BEB protocol than a global optimization model. We show that the BEB protocol can be reverse-engineered through a non-cooperative game in which each link try to maximize, using a stochastic subgradient formed by local information, its own utility function in the form of expected net reward for successful transmission. However, even though the existence of a Nash equilibrium can be proved, neither convergence nor social welfare optimality is guaranteed. This necessitates the forward-engineering of the MAC protocol, which aims at developing a new MAC protocol explicitly considering desired efficiency and fairness properties.

For forward-engineering, we formulate a NUM problem in terms of the persistence probabilities of each node and link to develop a random access protocol. It turns out that, except for the special case of logarithmic utility, the resulting NUM problem is non-convex and non-separable. Convexity and separablility properties are essential to develop a distributed and optimal algorithm for a NUM problem. Despite these difficulties, we show how to develop distributed MAC protocols in which each link adjusts its own persistence probability based on only its local information and limited message passing. We also provide readily-verifiable sufficient conditions under which convergence of the proposed algorithms to a global optimality can be proved.

Compared with all the previous studies on MAC protocols by using NUM, our approach in this paper has following distinguishing features. In contrast to [2], [5], [6] where contention is modeled using 'deterministic approximation' and well-developed algorithms for NUM in wired networks, we directly model contentions among transmissions through a collision and persistence probability model. It will be shown that the deterministic approximation approach [2], [5], [6] cannot accurately model contention in the random access protocol. In [2], a distributed algorithm considering collision probabilities that approximately solves the NUM problem is proposed. However, it is not rigorously shown how closely it can approximate the optimal solution. In contrast to [7], [8], [9] where only maxmin fairness and proportional fairness (*i.e.,* only special types of utility functions such as log utility functions) are considered, we consider general types of utility functions, which facilitates the provisioning of different types of services and different efficiency-fairness tradeoffs.

After developing new contention-based MAC protocols based on NUM distributed solutions, we compare the performance of these algorithms with the existing MAC standards and recent proposals in the literature.

- Our random access MAC protocols (proposed in Section IV).
- Random access protocol based on known deterministic approximations.
- Random access BEB protocol in the current IEEE 802.11 standard.

We show that our protocol provides better efficiency-fairness tradeoffs than both the deterministic approximation and BEB protocols.

In summary, there are three contributions to the optimal design of wireless MAC in this paper:

- A non-cooperative game-theoretic model for the BEB protocol, which is currently used for MAC in ad-hoc networks, is established. Various properties of the BEB protocol are then characterized, such as the utility function that each link implicitly tries to maximize, a Nash equilibrium that is an operating point of the game, and convergence properties of the BEB protocol to the Nash equilibrium. The methodology introduced is also applicable to other types of contention resolution backoff protocols, and fills the hole in the recent research thrust of reverse-engineering layers 2-4 protocols (after the basic NUM model for TCP in layer 4 and Stable Path Routing model for BGP in layer 3).
- A wireless NUM problem is formulated for random access MAC through explicit modeling of collision and persistence probabilities. This problem is solved by two distributed algorithms, which are proved to converge to a global optimum if the utility functions' curvatures are sufficiently negative.
- The value of a small amount of message passing overhead (in the proposed random access protocols) is quantified by comparing efficiency and fairness against the existing BEB protocol. The advantages of probabilistic models and deficiency of deterministic models for wireless MAC NUM problems are illustrated.

The rest of the paper is organized as follows. In Section II, we provide the system model that we consider in this paper. In Sections III and IV, we present reverse and forward engineering for MAC protocols, respectively. We provide detailed numerical examples comparing our protocols with the state-of-the-art in Section V, and conclude in Section VI.

II. System Model and Notation

In this paper, we consider an ad-hoc network represented by a directed graph $G(V, E)$, *e.g.*, as in Figure 1, where V is the set of nodes and E is the set of logical links. We define $L_{out}(n)$ as the set of outgoing links from node n, $L_{in}(n)$ as the set of incoming links to node n, t_l as the transmitter node of link l, and r_l as the receiver node of link l. We also

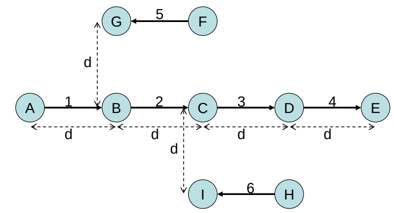

Fig. 1. A network topology graph.

define $N_{to}^I(l)$ as the set of nodes whose transmissions[1] cause interference to the receiver of link l, excluding the transmitter node of link l, (*i.e.*, t_l), and $L_{from}^I(n)$ as the set of links whose transmissions get interfered from the transmission of node n, excluding outgoing links from node n (*i.e.*, $l \in L_{out}(n)$). Hence, if the transmitter of link l and a node in set $N_{to}^I(l)$ transmit data simultaneously, the transmission of link l fails. If node n and the transmitter of a link l in set $L_{from}^I(n)$ transmit data simultaneously, the transmission of link l also fails.

Each node and each link has a contention resolution protocol based on the transmission persistence probability. Each node n transmits data with a probability P^n. When it determines to transmit data, it chooses one of its outgoing links with a probability q_l, $\forall l \in L_{out}(n)$, such that $\sum_{l \in L_{out}(n)} q_l = 1$, and transmits data only on the chosen link. Hence, there is no collision among links that have the same transmitter node. Consequently, link l, $l \in L_{out}(n)$, transmits data with a probability $p_l = P^n q_l$ such that

$$\sum_{l \in L_{out}(n)} p_l = P^n, \ \forall n. \tag{3}$$

We call q_l and p_l conditional persistence probability and persistence probability of link l, respectively. Based on its adopted algorithm, which will be studied in next two sections, each node and link adjusts its persistence probability.

III. Reverse Engineering: Non-cooperative Game Model of BEB Protocol

In this section, we reverse-engineer the current BEB protocol in wireless ad-hoc networks, which is one of the popular contention resolution algorithms used in various random access systems such as IEEE 802.11 DCF. We characterize the problem that is implicitly solved by the BEB protocol and analyze its performance. As mentioned earlier, in contrast to the TCP/AQM protocol that can be modeled as a basic NUM (1), we model the BEB protocol as a non-cooperative game due to the coupled utility of each link through collisions and the lack of sufficient feedback from the network.

In the IEEE 802.11 implementation, the BEB protocol is window-based. In this protocol, each link l maintains its contention window size W_l, current window size CW_l, and minimum and maximum window sizes W_l^{min} and W_l^{max}. After each transmission, contention window size and current window

[1]We say that node n transmits data if one of its outgoing links transmits data.

size are updated. If transmission is successful, the contention window size is reduced to the minimum window size (*i.e.*, $W_l = W_l^{min}$), otherwise it is doubled until reaching the maximum window size W_l^{max} (*i.e.*, $W_l = \min\{2W_l, W_l^{max}\}$). Then, current window size CW_l is updated to be a number between $(0, W_l)$ following a uniform distribution. It decreases in every time-slot, and when it becomes zero, the link transmits data.

Here we study the window-based BEB MAC protocol based on a persistence probabilistic model, an approach analogous to the source rate model in the literature for the window-based TCP congestion control protocol. In our model, each link maintains its persistence probability with which it decides to transmit or not in each time-slot. After each transmission attempt, if the transmission is successful without collisions, then link l sets its persistence probability to be its maximum value, p_l^{max}. Otherwise, it reduces its persistence probability by a factor β_l $(0 < \beta_l < 1)$ up to its minimum value p_l^{min}. This update algorithm can be written as:

$$p_l(t+1) = \max\{p_l^{min}, p_l^{max}\mathbf{1}_{\{T_l(t)=1\}}\mathbf{1}_{\{C_l(t)=0\}} + \beta_l p_l(t)\mathbf{1}_{\{T_l(t)=1\}}\mathbf{1}_{\{C_l(t)=1\}} + p_l(t)\mathbf{1}_{\{T_l(t)=0\}}\}, \quad (4)$$

where $p_l(t)$ is a persistence probability of link l at time-slot t, $\mathbf{1}_a$ is an indicator function of event a, and $T_l(t)$ and $C_l(t)$ are the events that link l transmits data at time-slot t and that there is a collision to link l's transmission given that link l transmits data at time-slot t, respectively. Then, given $\mathbf{p}(t)$ (*i.e.*, $\mathbf{P}(t)$ from (3)), we have

$$\text{Prob}\{T_l(t) = 1|\mathbf{p}(t)\} = p_l(t),$$

and

$$\text{Prob}\{C_l(t) = 1|\mathbf{p}(t)\} = 1 - \prod_{n\in N_{to}^I(l)}(1 - P^n(t)) = 1 - \prod_{n\in N_{to}^I(l)}(1 - \sum_{m\in L_{out}(n)} p_m(t)).$$

Note that $p_l(t)$ is a random process whose transitions depend on random events $T_l(t)$ and $C_l(t)$. We first study its average behavior and return to (4) later in this section. Slightly abusing the notation, we still use $p_l(t)$ to denote the average behavior of the persistence probability. From (4), we have

$$\begin{aligned} & p_l(t+1) \\ &= \max\{p_l^{min}, p_l^{max}\mathrm{E}\{\mathbf{1}_{\{T_l(t)=1\}}\mathbf{1}_{\{C_l(t)=0\}}|\mathbf{p}(t)\} \\ &\quad +\beta_l \mathrm{E}\{p_l(t)\mathbf{1}_{\{T_l(t)=1\}}\mathbf{1}_{\{C_l(t)=1\}}|\mathbf{p}(t)\} \\ &\quad +\mathrm{E}\{p_l(t)\mathbf{1}_{\{T_l(t)=0\}}|\mathbf{p}(t)\}\} \\ &= \max\{p_l^{min}, p_l^{max}p_l(t)\prod_{n\in N_{to}^I(l)}\left(1 - \sum_{m\in L_{out}(n)} p_m(t)\right) \\ &\quad +\beta_l p_l(t)p_l(t)\left(1 - \prod_{n\in N_{to}^I(l)}(1 - \sum_{m\in L_{out}(n)} p_m(t))\right) \\ &\quad +p_l(t)(1-p_l(t))\}, \end{aligned} \quad (5)$$

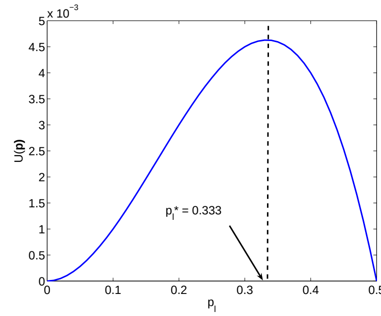

Fig. 2. Utility function of a link as a function of its own persistence probability with $\beta_l = 0.5$, $p_l^{max} = 0.5$, and $\prod_{n\in N_{to}^I(l)}(1-\sum_{m\in L_{out}(n)} p_m) = 0.5$.

where $\mathrm{E}\{a|b\}$ is the expected value of a given b.

We now model the update algorithm in (5) as a non-cooperative game, in which each link l updates its strategy, *i.e.*, its persistence probability p_l, to maximize its utility U_l based on strategies of the other links, *i.e.*, $\mathbf{p}_{-l} = (p_1, \cdots, p_{l-1}, p_{l+1}, \cdots, p_{|L|})$. We can then formulate the BEB protocol as a non-cooperative game, $G_{BEB-MAC} = [L, \times_{l\in L}A_l, \{U_l\}_{l\in L}]$, where L is a set of players, *i.e.*, links, $A_l = \{p_l \mid p_l^{min} \le p_l \le p_l^{max}\}$ is an action set of player l, and U_l is a utility function of player l. We refer to this as the BEB-MAC Game and now study its properties and solutions.

In the non-cooperative game, one of the most important questions is whether a Nash equilibrium [17] exists or not. In the case of BEB-MAC Game, we have the following

Definition 1: A persistence probability vector $\mathbf{p}^*$ is said to be a Nash equilibrium if

$$U_l(p_l^*, \mathbf{p}_{-l}^*) \ge U_l(p_l, \mathbf{p}_{-l}^*), \ p_l^{min} \le p_l \le p_l^{max}, \ \ \forall l.$$

Hence, no link can improve its utility by unilaterally deviating its persistence probability from the Nash equilibrium. The following reverse-engineering theorem obtains the underlying utility functions in the BEB-MAC Game and establishes the existence of a Nash equilibrium for the game.

Theorem 1: The utility function is the following expected net reward (expected reward minus expected cost) that the link can obtain from its transmission:

$$U_l(\mathbf{p}) = R(p_l)S(\mathbf{p}) - C(p_l)F(\mathbf{p}), \ \ \forall l \quad (6)$$

where $S(\mathbf{p}) = p_l\prod_{n\in N_{to}^I(l)}(1-\sum_{m\in L_{out}(n)} p_m)$ is the probability of transmission success, $F(\mathbf{p}) = p_l(1-\prod_{n\in N_{to}^I(l)}(1-\sum_{m\in L_{out}(n)} p_m)$ is the probability of transmission failure, and $R(p_l) \stackrel{\text{def}}{=} p_l(\frac{1}{2}p_l^{max} - \frac{1}{3}p_l)$ can be interpreted as the reward for transmission success, $C(p_l) \stackrel{\text{def}}{=} \frac{1}{3}(1-\beta_l)p_l^3$ can be interpreted as the cost for transmission failure.

Furthermore, there exists a Nash equilibrium in the BEB-MAC Game $G_{BEB-MAC} = [L, \times_{l\in L}A_l, \{U_l\}_{l\in L}]$, which is obtained by

$$p_l^* = \frac{p_l^{max}\prod_{n\in N_{to}^I(l)}(1-\sum_{m\in L_{out}(n)} p_m^*)}{1-\beta_l(1-\prod_{n\in N_{to}^I(l)}(1-\sum_{m\in L_{out}(n)} p_m^*))}, \ \forall l. \quad (7)$$

Proof: We first obtain the utility function of each link based on the update algorithm in (5). Assuming that there exists an equilibrium persistence probabilities $\mathbf{p}^*$, $\mathbf{p}^{min} < \mathbf{p}^* < \mathbf{p}^{max}$, then we see from (5) that $\mathbf{p}^*$ satisfies the following:

$$p_l^* = p_l^{max} p_l^* \prod_{n \in N_{to}^I(l)} (1 - \sum_{m \in L_{out}(n)} p_m^*) + \beta_l p_l^* p_l^* (1 - \prod_{n \in N_{to}^I(l)} (1 - \sum_{m \in L_{out}(n)} p_m^*)) + p_l^*(1 - p_l^*). \quad (8)$$

Since each link adjusts its own persistence probability to maximize its utility given persistence probabilities of the other link, from (8) and the first order necessary condition, each link l has its utility function, $U_l(\mathbf{p})$, such that

$$\frac{\partial U_l(\mathbf{p})}{\partial p_l} = p_l^{max} p_l \prod_{n \in N_{to}^I(l)} (1 - \sum_{m \in L_{out}(n)} p_m) + \beta_l p_l p_l (1 - \prod_{n \in N_{to}^I(l)} (1 - \sum_{m \in L_{out}(n)} p_m)) + p_l(1 - p_l) - p_l. \quad (9)$$

Hence, the utility function of link l, $U_l(\mathbf{p})$, which is unique up to a constant offset, is obtained as

$$\begin{aligned} U_l(\mathbf{p}) &= \frac{1}{2} p_l^{max} \prod_{n \in N_{to}^I(l)} (1 - \sum_{m \in L_{out}(n)} p_m) p_l^2 + \frac{1}{3}\beta_l (1 - \prod_{n \in N_{to}^I(l)} (1 - \sum_{m \in L_{out}(n)} p_m)) p_l^3 - \frac{1}{3} p_l^3 \\ &= p_l^2 \prod_{n \in N_{to}^I(l)} (1 - \sum_{m \in L_{out}(n)} p_m)(\frac{1}{2} p_l^{max} - \frac{1}{3} p_l) - \frac{1}{3}(1 - \beta_l) p_l^3 (1 - \prod_{n \in N_{to}^I(l)} (1 - \sum_{m \in L_{out}(n)} p_m)) \\ &= R(p_l) S(\mathbf{p}) - C(p_l) F(\mathbf{p}), \end{aligned} \quad (10)$$

where $R(p_l) = p_l(\frac{1}{2} p_l^{max} - \frac{1}{3} p_l)$, $C(p_l) = \frac{1}{3}(1 - \beta_l) p_l^3$, $S(\mathbf{p}) = p_l \prod_{n \in N_{to}^I(l)} (1 - \sum_{m \in L_{out}(n)} p_m)$, and $F(\mathbf{p}) = p_l(1 - \prod_{n \in N_{to}^I(l)} (1 - \sum_{m \in L_{out}(n)} p_m)$.

It can be verified that utility function U_l is quasi-concave in p_l. The action set $A_l = \{p_l \mid p_l^{min} \le p_l \le p_l^{max}\}$ of each link l is a nonempty compact convex subset of a Euclidian space and the utility function U_l of each link l is continuous and quasi-concave on A_l. Hence, by Proposition 20.3 in [17], there exists a Nash equilibrium.

Moreover, from (9), we can easily show that

$$\frac{\partial U_l(\mathbf{p})}{\partial p_l} \begin{cases} > 0, \text{if } p_l < \frac{p_l^{max} \prod_{n \in N_{to}^I(l)} (1 - \sum_{m \in L_{out}(n)} p_m)}{1 - \beta_l (1 - \prod_{n \in N_{to}^I(l)} (1 - \sum_{m \in L_{out}(n)} p_m))} \\ < 0, \text{otherwise} \end{cases}.$$

Hence, we can characterize the Nash equilibrium for persistence probabilities of links as

$$p_l^* = \frac{p_l^{max} \prod_{n \in N_{to}^I(l)} (1 - \sum_{m \in L_{out}(n)} p_m^*)}{1 - \beta_l (1 - \prod_{n \in N_{to}^I(l)} (1 - \sum_{m \in L_{out}(n)} p_m^*))}, \ \forall l.$$

■

From (7), we can conclude that, other conditions being the same, at the Nash Equilibrium a link l will have a higher persistence probability if it has a higher value of p_l^{max}, a higher value of β_l, or a higher value of $\prod_{n \in N_{to}^I(l)} (1 - \sum_{m \in L_{out}(n)} p_m^*)$, *i.e.*, a higher transmission success probability.

We also have the next corollary that immediately follows from (5) and (7).

Corollary 1: If $\mathbf{p}(t)$ updated by (5) converges to $\mathbf{p}^*$, $\mathbf{p}^{min} < \mathbf{p}^* < \mathbf{p}^{max}$, then $\mathbf{p}^*$ is a Nash equilibrium.

We next introduce a natural strategy of each link, in which each link updates its persistence probability for the next time-slot such that it maximizes its utility based on the persistence probabilities of the other links in the current time-slot:

$$p_l^*(t+1) = \underset{p_l^{min} \le p_l \le p_l^{max}}{\operatorname{argmax}} U_l(p_l, \mathbf{p}_{-l}^*(t)). \quad (11)$$

Hence, $p_l^*(t+1)$ is the 'best response' of link l given $\mathbf{p}_{-l}^*(t)$. Then, the following theorem provides the convergence properties of the best response strategy to the Nash equilibrium in the BEB-MAC Game.

Theorem 2: Suppose that the persistence probability of each link is updated by the best response function in (11) in each time-slot with $\mathbf{p}^*(0) = \mathbf{p}^{min}$.

(a) When $|L| = 2$, the persistence probabilities converge to the Nash equilibrium.

(b) For a general system with $|L| > 2$ links,

$$\mathbf{p}^*(2t+1) \to \hat{\mathbf{p}} \text{ and } \mathbf{p}^*(2t) \to \tilde{\mathbf{p}} \text{ as } t \to \infty.$$

If $\hat{\mathbf{p}} = \tilde{\mathbf{p}}$ *i.e.*, if $\mathbf{p}^*(t)$ converges to $\hat{\mathbf{p}}$, then $\hat{\mathbf{p}}$ is a Nash equilibrium.

Proof: We have

$$\frac{\partial^2 U_l(\mathbf{p})}{\partial p_l \partial p_k} = \begin{cases} \prod_{n \in N_{to}^I(l), k \notin L_{out}(n)} (1 - \sum_{m \in L_{out}(n)} p_m)(\beta_l p_l^2 - p_l^{max} p_l) \le 0, \\ \quad \text{if } k \in L_{out}(r),\ r \in N_{to}^I(l) \\ 0, \quad \text{otherwise} \end{cases}.$$

Since $\beta_l < 1$ and $p_l \le p_l^{max}$, the utility function is submodular[2]. Moreover, the action set of a link does not depend on the strategies of the other links. Hence, by applying Theorems 2.3 and 5.1 in [19], the proof is completed. ■

In practice, the persistence probability in the BEB protocol is not updated by the best response strategy (11), but by (4) (or by (5) on average). Now using (9), we can rewrite (5) as

$$p_l(t+1) = \max\left\{ p_l^{min}, p_l(t) + \frac{\partial U_l(\mathbf{p})}{\partial p_l}|_{\mathbf{p}=\mathbf{p}(t)} \right\}.$$

Hence, instead of instantaneously setting $p_l(t+1)$ to the best response $p_l^*(t+1)$, as in (11), in (5), each link updates its

[2]If U_l is twice differentiable and $\frac{\partial^2 U_l(\mathbf{p})}{\partial p_l \partial p_k} \le 0, \forall \mathbf{p} \in \times_{l \in L} A_l\ \forall k \ne l$, then U_l is submodular. We refer readers to [18], [19] for more details on submodularity.

persistence probability to the direction of the maximizer by using the gradient. To update its persistence probability by using (5), each link l must know the persistence probabilities of its adjacent links, *i.e.*, link m, $m \in L_{out}(n)$, $n \in N^I_{to}(l)$. However, in the BEB protocol, there is no explicit message passing among links, and the link cannot obtain the exact information to evaluate the gradient of its utility function. Instead of using the exact gradient of its utility function, as in (5), each link attempts to approximate it using (4). In fact, we can rewrite (4) as

$$p_l(t+1) = \max\{p_l^{min}, p_l(t) + v_l(t)\},$$

where

$$\begin{aligned} v_l(t) = & p_l^{max}\mathbf{1}_{\{T_l(t)=1\}}\mathbf{1}_{\{C_l(t)=0\}} + \beta_l p_l(t)\mathbf{1}_{\{T_l(t)=1\}}\mathbf{1}_{\{C_l(t)=1\}} \\ & + p_l(t)\mathbf{1}_{\{T_l(t)=0\}} - p_l. \end{aligned}$$

Since

$$\begin{aligned} & \mathrm{E}\{v_l(t)|\mathbf{p}(t)\} \\ = & p_l^{max} p_l(t) \prod_{n\in N^I_{to}(l)} (1 - \sum_{m\in L_{out}(n)} p_m(t)) \\ & + \beta_l p_l(t) p_l(t)(1 - \prod_{n\in N^I_{to}(l)} (1 - \sum_{m\in L_{out}(n)} p_m(t))) \\ & + p_l(t)(1 - p_l(t)) - p_l(t) \\ = & \frac{\partial U_l(\mathbf{p})}{\partial p_l}|_{\mathbf{p}=\mathbf{p}(t)}, \end{aligned}$$

$v_l(t)$ is a stochastic subgradient [20] of U_l at $\mathbf{p}(t)$.

In summary, we have the following reverse-engineering result in addition to Theorem 1:

Theorem 3: The BEB protocol described by (4) is a stochastic subgradient algorithm to maximize utility (6), where each stochastic subgradient v_l can be measured by the link itself through collision and success of its transmission (without explicit message passing among links).

IV. Forward Engineering: Design of Optimal Random Access MAC Protocol

As the last section on reverse engineering shows, the current IEEE 802.11 MAC protocol does not provide adequate feedback information to the contending nodes. In this section on forward engineering, we propose a cooperative NUM formulation and develop enhanced MAC protocols as distributed solutions to such a network-wide maximization problem. We will show that limited message passing of 'contention prices' can provide the optimal coordination to align the selfish non-cooperative nature of the BEB protocol into a new MAC protocol that always converges, and to the global optimum of network utility.

Each link has a utility function $U_l(x_l)$, an increasing non-linear function of its average data rate x_l, which is in turn a function of persistence probabilities. We would like to discover how each link should adjust its own persistence probability so as to globally maximize the total network utility. The average data rate x_l on link l is obtained as:

$$x_l = c_l p_l \prod_{k\in N^I_{to}(l)} (1 - P^k), \ \forall l, \tag{12}$$

where c_l is a fixed data rate of link l. The NUM problem can be formulated as follows, over the optimization variables of $\mathbf{p}, \mathbf{P}$:[3]

$$\begin{array}{ll} \text{maximize} & \sum_l U_l(c_l p_l \prod_{k\in N^I_{to}(l)}(1-P^k)) \\ \text{subject to} & x_l^{min} \le c_l p_l \prod_{k\in N^I_{to}(l)}(1-P^k) \le x_l^{max}, \ \forall l \\ & \sum_{l\in L_{out}(n)} p_l = P^n, \ \forall n \\ & 0 \le P^n \le 1, \ \forall n \\ & 0 \le p_l \le 1, \ \forall l \end{array} \tag{13}$$

where x_l^{min} and x_l^{max} are constraints on minimum and maximum average data rates of link l, respectively.

The objective of this problem is to obtain the optimal persistence probabilities $\mathbf{p}$ for links and $\mathbf{P}$ for nodes so as to maximize the network utility. For log utilities, problem (13) is readily shown to be a decomposable convex optimization [8]. However, in general, it is non-convex and non-separable, thus extremely difficult to be distributively solved for global optimality. We show that under a readily-verifiable sufficient condition on curvatures of the utility functions, it can be turned into a convex and separable optimization problem.

We first introduce a new variable x_l for the average data rate of link l, using (12):

$$\begin{array}{ll} \text{maximize} & \sum_{l\in L} U_l(x_l) \\ \text{subject to} & x_l = c_l p_l \prod_{k\in N^I_{to}(l)}(1-P^k), \ \forall l \\ & \sum_{l\in L_{out}(n)} p_l = P^n, \ \forall n \\ & x_l^{min} \le x_l \le x_l^{max}, \ \forall l \\ & 0 \le P^n \le 1, \ \forall n \\ & 0 \le p_l \le 1, \ \forall l. \end{array} \tag{14}$$

Without loss of generality, we can replace the equality in the first constraint with an inequality:[4]

$$\begin{array}{ll} \text{maximize} & \sum_{l\in L} U_l(x_l) \\ \text{subject to} & x_l \le c_l p_l \prod_{k\in N^I_{to}(l)}(1-P^k), \ \forall l \\ & \sum_{l\in L_{out}(n)} p_l = P^n, \ \forall n \\ & x_l^{min} \le x_l \le x_l^{max}, \ \forall l \\ & 0 \le P^n \le 1, \ \forall n \\ & 0 \le p_l \le 1, \ \forall l. \end{array} \tag{15}$$

The next step of problem transformation is to take the log of both sides of the first constraint in problem (15) and a log change of variables and constants: $x'_l = \log x_l$, $x'^{max}_l = \log x_l^{max}$, $x'^{min}_l = \log x_l^{min}$, $U'_l(x'_l) = U_l(e^{x'_l})$, and $c'_l =$

[3] Similar models have recently been considered in [7], [8], [9] for a restricted class of utility functions, which indeed can be recovered as special cases of the general results in this section.

[4] This is because the first inequality in (15) will always be satisfied with equality at optimality.

$\log c_l$. This reformulation turns the problem into:

$$\begin{aligned}
\text{maximize } & \textstyle\sum_{l\in L} U'_l(x'_l) \\
\text{subject to } & c'_l + \log p_l + \textstyle\sum_{k\in N^I_{to}(l)} \log(1-P^k) - x'_l \geq 0, \ \ \forall l \\
& \textstyle\sum_{l\in L_{out}(n)} p_l = P^n, \ \ \forall n \\
& x'^{min}_l \leq x'_l \leq x'^{max}_l, \ \ \forall l \\
& 0 \leq P^n \leq 1, \ \ \forall n \\
& 0 \leq p_l \leq 1, \ \ \forall l.
\end{aligned} \tag{16}$$

Note that problem (16) is now separable but still may not be a convex optimization problem, since the objective $U'_l(x'_l)$ may not be a (strictly) concave function, even though $U_l(x_l)$ is a (strictly) concave function. However, the following lemma provides a sufficient condition for its concavity. Define

$$g_l(x_l) = \frac{d^2U_l(x_l)}{dx_l^2}x_l + \frac{dU_l(x_l)}{dx_l}.$$

Lemma 1: If $g_l(x_l) < 0$, $U'_l(x'_l)$ is a strictly concave function of x'_l.

Proof: Since $x_l = e^{x'_l}$,

$$\begin{aligned}
\frac{d^2U'_l(x_l)}{dx'^2_l} &= \frac{d^2U^x_l(x_l)}{dx_l^2}\left(\frac{dx_l}{dx'_l}\right)^2 + \frac{dU_l(x_l)}{dx_l}\frac{d^2x_l}{dx'^2_l} \\
&= e^{x'_l}\left(\frac{d^2U_l(x_l)}{dx_l^2}x_l + \frac{dU_l(x_l)}{dx_l}\right) \\
&= e^{x'_l}g_l(x_l).
\end{aligned}$$

Hence, if $g_l(x_l) < 0$, $U'_l(x'_l)$ is a strictly concave function of x'_l. ■

Remark 1: The condition of $g_l(x_l) < 0$ is equivalent to:

$$\frac{d^2U_l(x_l)}{dx_l^2} < -\frac{dU_l(x_l)}{x_l dx_l}.$$

Since utility functions are increasing, $\frac{dU_l(x_l)}{dx_l}$ has a positive value. The above inequality states that the utility function needs to be not just strictly concave (*i.e.*, $\frac{d^2U_s(x_l)}{dx_l^2} < 0$), but with a curvature that is bounded away from 0 by as much as $\frac{dU_l(x_l)}{x_l dx_l}$, *i.e.*, the application needs represented by the utility functions must be elastic enough.

For example, consider the following utility function (2) parameterized by $\alpha \geq 0$ [12]:

$$U_l(x_l) = \begin{cases} (1-\alpha)^{-1}x_l^{1-\alpha}, & \text{if } \alpha \neq 1 \\ \log x_l, & \text{otherwise} \end{cases}.$$

Then,

$$\begin{cases} g_l(x_l) > 0, & \text{if } \alpha < 1 \\ g_l(x_l) = 0, & \text{if } \alpha = 0 \\ g_l(x_l) < 0, & \text{if } \alpha > 1 \end{cases}.$$

Hence, in this type of utility functions, if $\alpha > 1$, $U'_l(x'_l)$ becomes a strictly concave function. Throughout this section, we will assume that the condition in Lemma 1 is satisfied.

We now solve problem (16) by using a dual decomposition approach. We write down the Lagrangian function associated with problem (16) as follows, where λ_l is the Lagrange multiplier on link l with an interpretation of 'contention price':

$$\begin{aligned}
& L(\boldsymbol{\lambda}, \mathbf{x}', \mathbf{p}, \mathbf{P}) \\
&= \sum_{l\in L} U'_l(x'_l) \\
&\quad + \sum_{l\in L}\{\lambda_l(c'_l + \log p_l + \sum_{k\in N^I_{to}(l)} \log(1-P^k) - x'_l)\} \\
&= \sum_{l\in L}\{U'_l(x'_l) - \lambda_l x'_l\} + \sum_{l\in L}\lambda_l \log p_l \\
&\quad + \sum_{l\in L}\lambda_l \sum_{k\in N^I_{to}(l)} \log(1-P^k) + \sum_{l\in L}\lambda_l c'_l \\
&= \sum_{n\in N}\sum_{l\in L_{out}(n)}\{U'_l(x'_l) - \lambda_l x'_l\} + \sum_{n\in N}\sum_{l\in L_{out}(n)}\lambda_l \log p_l \\
&\quad + \sum_{n\in N}\sum_{m\in L^I_{from}(n)}\lambda_m \log(1-P^n) + \sum_{l\in L}\lambda_l c'_l.
\end{aligned} \tag{17}$$

Note that in this Lagrangian, we do not need to relax the second constraint in problem (16). The Lagrange dual function is

$$Q(\boldsymbol{\lambda}) = \max_{\substack{\sum_{l\in L_{out}(n)} p_l = P^n, \ \forall n \\ \mathbf{x}'^{min} \preceq \mathbf{x}' \preceq \mathbf{x}'^{max} \\ \mathbf{0} \preceq \mathbf{p} \preceq \mathbf{1} \\ \mathbf{0} \preceq \mathbf{P} \preceq \mathbf{1}}} L(\boldsymbol{\lambda}, \mathbf{x}', \mathbf{p}, \mathbf{P}). \tag{18}$$

The dual problem is formulated as

$$\min_{\boldsymbol{\lambda}\succeq \mathbf{0}} Q(\boldsymbol{\lambda}). \tag{19}$$

To solve the dual problem, we first consider problem (18). This maximization of the Lagrangian over $(\mathbf{x}', \mathbf{p}, \mathbf{P})$ can be conducted in parallel at each node n:

$$\max_{x'^{min}_l \leq x' \leq x'^{max}_l} \{U'_l(x'_l) - \lambda_l x'_l\}, \ \forall l \in L_{out}(n) \tag{20}$$

and

$$\begin{aligned}
\text{maximize } & \sum_{l\in L_{out}(n)} \lambda_l \log p_l + \sum_{k\in L^I_{from}(n)} \lambda_k \log(1-P^n) \\
\text{subject to } & \textstyle\sum_{l\in L_{out}(n)} p_l = P^n \\
& 0 \leq p_l \leq 1, \ \ \forall l \in L_{out}(n) \\
& 0 \leq P^n \leq 1.
\end{aligned} \tag{21}$$

To obtain the solutions of problem (21), we first define k_n as

$$k_n = \sum_{l\in L_{out}(n)} \lambda_l + \sum_{k\in L^I_{from}(n)} \lambda_k, \forall n.$$

Then, for a given $\boldsymbol{\lambda}$, the $\mathbf{P}(\boldsymbol{\lambda})$ and $\mathbf{p}(\boldsymbol{\lambda})$ that maximize problem (21) are obtained in closed-form as[5]

$$P^n(\boldsymbol{\lambda}) = \begin{cases} \frac{\sum_{l\in L_{out}(n)} \lambda_l}{\sum_{l\in L_{out}(n)} \lambda_l + \sum_{k\in L^I_{from}(n)} \lambda_k}, & \text{if } k_n \neq 0 \\ \frac{|L_{out}(n)|}{|L_{out}(n)| + |L^I_{from}(n)|}, & \text{if } k_n = 0 \end{cases}, \ \forall n$$

[5] If $k_n = 0$ then $P^n(\boldsymbol{\lambda})$ and $p_l(\boldsymbol{\lambda})$, $\forall l \in L_{out}(n)$, can be any feasible solutions that satisfy the constraints of problem (21).

and

$$p_l(\boldsymbol{\lambda}) = \begin{cases} \frac{\lambda_l}{\sum_{l\in L_{out}(n)}\lambda_l+\sum_{k\in L^I_{from}(n)}\lambda_k}, & \text{if } k_n \neq 0 \\ \frac{1}{|L_{out}(n)|+|L^I_{from}(n)|}, & \text{if } k_n = 0 \end{cases}, \quad \forall l \in L_{out}(n), \forall n.$$

We can now solve the dual problem (19) by using a subgradient projection algorithm[6] at each link l, *i.e.*, at each node n such that $l \in L_{out}(n)$, as

$$\lambda_l(t+1) = \Big[\lambda_l(t) - \alpha(t)\Big(c'_l + \log p_l(t) + \sum_{k\in N^I_{to}(l)} \log\left(1-P^k(t)\right) - x'_l(t)\Big)\Big]^+, \quad \forall l \in L_{out}(n),\ \forall n, \tag{22}$$

where $x'_l(t)$, and $p_l(t)$ and $P^k(t)$ are the solutions to problems (20) and (21), respectively, with $\boldsymbol{\lambda}(t)$. Hence, problems (20), (21), and (22) can be solved in each node n in a distributed way with only local information, *i.e.*, $\lambda_l(t)$, $\forall l \in L^I_{from}(n) \cup L_{out}(n)$ and $P^m(t)$, $\forall m \in N^I_{to}(l)$, $\forall l \in L_{out}(n)$. The proposed MAC protocol can be implemented through the following algorithm with very simple local computational steps, and a limited amount of explicit message passing in Step 3.2, which we will later remove through addtional improvements and heuristics.

Distributed Algorithm 1 for MAC Protocol

1: Each node n constructs its *local* interference graph to obtain sets $L_{out}(n)$, $L_{in}(n)$, $L^I_{from}(n)$, and $N^I_{to}(l)$, $\forall l \in L_{out}(n)$.
2: Each node n sets $t = 0$, $\lambda_l(1) = 1$, $\forall l \in L_{out}(n)$, $P^n(1) = \frac{|L_{out}(n)|}{|L_{out}(n)|+|L^I_{from}(n)|}$, and $p_l(1) = \frac{1}{|L_{out}(n)|+|L^I_{from}(n)|}$, $\forall l \in L_{out}(n)$.
3: *Locally* at each node n, do
3.1: Set $t \leftarrow t+1$.
3.2: Inform $\lambda_l(t)$ to all nodes in $N^I_{to}(l)$, $\forall l \in L_{out}(n)$, and $P^n(t)$ to t_l, $\forall l \in L^I_{from}(n)$.
3.3: Set $k_n(t) = \sum_{l\in L_{out}(n)}\lambda_l(t) + \sum_{k\in L^I_{from}(n)}\lambda_k(t)$ and $\alpha(t) = \frac{1}{t}$.
3.4: Solve the following problems to obtain $P^n(t+1)$, and $x'_l(t+1)$, $p_l(t+1)$, and $\lambda_l(t+1)$, $\forall l \in L_{out}(n)$:

$$P^n(t+1) = \begin{cases} \frac{\sum_{l\in L_{out}(n)}\lambda_l(t)}{\sum_{l\in L_{out}(n)}\lambda_l(t)+\sum_{k\in L^I_{from}(n)}\lambda_k(t)}, & \text{if } k_n(t) \neq 0 \\ \frac{|L_{out}(n)|}{|L_{out}(n)|+|L^I_{from}(n)|}, & \text{if } k_n(t) = 0 \end{cases}$$

$$p_l(t+1) = \begin{cases} \frac{\lambda_l(t)}{\sum_{l\in L_{out}(n)}\lambda_l(t)+\sum_{k\in L^I_{from}(n)}\lambda_k(t)}, & \text{if } k_n(t) \neq 0 \\ \frac{1}{|L_{out}(n)|+|L^I_{from}(n)|}, & \text{if } k_n(t) = 0 \end{cases}$$

$$x'_l(t+1) = \underset{x'^{min}_l \le x' \le x'^{max}_l}{\operatorname{argmax}} \left\{U'_l(x'_l) - \lambda_l(t)x'_l\right\},$$

and

$$\lambda_l(t+1) = \Big[\lambda_l(t) - \alpha(t)\Big(c'_l + \log p_l(t) + \sum_{k\in N^I_{to}(l)} \log\left(1-P^k(t)\right) - x'_l(t)\Big)\Big].$$

3.5: Each node n sets the conditional persistence probability of each of its outgoing links $q_l(t) = p_l(t)/P^n(t)$, $\forall l \in L_{out}(n)$.
3.6: Each node n decides if it will transmit data with a probability $P^n(t)$. If it decides to transmit data, it chooses to transmit on one of its outgoing links with a probability $q_l(t)$, $\forall l \in L_{out}(n)$.
while (1).

Remark 2: Note that the above algorithm is conducted at each node n to calculate P^n, p_l, λ_l, and x'_l for its *outgoing* link l (*i.e.*, $\forall l \in L_{out}(n)$). Hence, the above algorithm is conducted at the *transmitter node* of each link. If we assume that two nodes within interference range can communicate with each other (*e.g.*, if nodes within distance $2d$ in Figure 1 can establish a communication link), in the above algorithm each node requires information from nodes within two-hop distance from it. To calculate P^n and p_l for its outgoing link l (*i.e.*, $\forall l \in L_{out}(n)$), node n needs λ_m from the transmitter node t_m of link m that is interfered from the transmission of node n (*i.e.*, from t_m, $\forall m \in L^I_{from}(n)$). Note that t_m is within two-hop from node n. For example, in Figure 1, node D needs λ_2 from the transmitter of link 2 (node B), since link 2 is in set $L^I_{from}(D)$, and node B is at two-hop distance from node D.

Also, to calculate λ_l for its outgoing link l (*i.e.*, $\forall l \in L_{out}(n)$), node n needs P^k from node k from whose transmission its outgoing link l is interfered (*i.e.*, from $k \in N^I_{to}(l)$, $\forall l \in L_{out}(n)$). For example, in Figure 1, node B needs P^D from node D, since its outgoing link, link 2 is interfered from the transmission of node D, *i.e.*, $D \in N^I_{to}(2)$ and $2 \in L_{out}(B)$, and node D is at two-hop distance from node B.

Alternatively, if λ_l and x'_l for each link l are calculated at its *receiver* node r_l instead of its transmitter node t_l, it is possible to implement an algorithm to solve (16) in which each node requires information from nodes only within *one-hop* distance.

Distributed Algorithm 2 for MAC Protocol

1: Each node n constructs its *local* interference graph to obtain sets $L_{out}(n)$, $L_{in}(n)$, $L^I_{from}(n)$, and $N^I_{to}(l)$, $\forall l \in L_{out}(n)$.
2: Each node n sets $t = 0$, $\lambda_l(1) = 1$, $\forall l \in L_{in}(n)$, $P^n(1) = \frac{|L_{out}(n)|}{|L_{out}(n)|+|L^I_{from}(n)|}$, and $p_l(1) = \frac{1}{|L_{out}(n)|+|L^I_{from}(n)|}$, $\forall l \in L_{out}(n)$.
3: *Locally* at each node n, do
3.1: Set $t \leftarrow t+1$.
3.2: Inform $\lambda_l(t)$ to node k, $\forall k \in N^I_{to}(l)$ and node t_l, $\forall l \in L_{in}(n)$, $P^n(t)$ to t_l, $\forall l \in L^I_{from}(n)$, and $p_l(t)$, $\forall l \in L_{out}(n)$ to each node r_l.
3.3: Set $k_n(t) = \sum_{l\in L_{out}(n)}\lambda_l(t) + \sum_{k\in L^I_{from}(n)}\lambda_k(t)$ and $\alpha(t) = \frac{1}{t}$.
3.4: Solve the following problems to obtain $P^n(t+1)$ and

[6]Since the solution to problem (21) may not be unique, $Q(\boldsymbol{\lambda})$ is not everywhere differentiable. Hence, we use a subgradient projection algorithm instead of using a gradient projection algorithm.

$p_l(t+1)$, $\forall l \in L_{out}(n)$, and $x'_l(t+1)$ and $\lambda_l(t+1)$, $\forall l \in L_{in}(n)$:

$$P^n(t+1) = \begin{cases} \frac{\sum_{l\in L_{out}(n)} \lambda_l(t)}{\sum_{l\in L_{out}(n)} \lambda_l(t) + \sum_{k\in L^I_{from}(n)} \lambda_k(t)}, & \text{if } k_n(t) \neq 0 \\ \frac{|L_{out}(n)|}{|L_{out}(n)|+|L^I_{from}(n)|}, & \text{if } k_n(t) = 0 \end{cases},$$

$$p_l(t+1) = \begin{cases} \frac{\lambda_l(t)}{\sum_{l\in L_{out}(n)} \lambda_l(t) + \sum_{k\in L^I_{from}(n)} \lambda_k(t)}, & \text{if } k_n(t) \neq 0 \\ \frac{1}{|L_{out}(n)|+|L^I_{from}(n)|}, & \text{if } k_n(t) = 0 \end{cases},$$

$$x'_l(t+1) = \underset{x'^{min}_l \le x' \le x'^{max}_l}{\operatorname{argmax}} \left\{ U'_l(x'_l) - \lambda_l(t) x'_l \right\},$$

and

$$\lambda_l(t+1) = \Bigg[\lambda_l(t) - \alpha(t) \Bigg(c'_l + \log p_l(t) + \sum_{k\in N^I_{to}(l)} \log\left(1 - P^k(t)\right) - x'_l(t) \Bigg)\Bigg].$$

3.5: Each node n sets the conditional persistence probability of each of its outgoing links $q_l(t) = p_l(t)/P^n(t)$, $\forall l \in L_{out}(n)$.
3.6: Each node n decides if it will transmit data with a probability $P^n(t)$. If it decides to transmit data, it chooses to transmit on one of its outgoing links with a probability $q_l(t)$, $\forall l \in L_{out}(n)$.
while (1).

Remark 3: In Algorithm 2, all message passing can be done within one-hop distance, at the expense of additional message passing of λ_l and p_l between transmitter and receiver nodes of link l. Steps in Algorithm 2 are conducted at each node n to calculate P^n and p_l for its outgoing link l (*i.e.*, $\forall l \in L_{out}(n)$), and λ_m and x'_m for its *incoming* link m (*i.e.*, $\forall m \in L_{in}(n)$). Hence, P^n of the transmitter node of link l and p_l for link l are calculated at the transmitter node of link l (*i.e.*, t_l), and λ_l and x'_l for link l are calculated at the *receiver node* of link l (*i.e.*, r_l). To calculate P^n and p_l for its outgoing link l (*i.e.*, $\forall l \in L_{out}(n)$), each node n now needs λ_m from the receiver node r_m of link m that is interfered from the transmission of node n (*i.e.*, from r_m, $\forall m \in L^I_{from}(n)$) and also λ_l from the receiver node r_l of its outgoing link l (*i.e.*, from r_l, $\forall l \in L_{out}(n)$). These r_m and r_l are at one-hop distance from node n. For example, in Figure 1, node D needs λ_2 from the receiver node of link 2 (node C), since link 2 is in set $L^I_{from}(D)$, and node C is at one-hop distance from node D.

Similarly, to calculate λ_l for its incoming link l (*i.e.*, $\forall l \in L_{in}(n)$), node n needs P^k from node k from whose transmission its incoming link l is interfered (*i.e.*, from $\forall k \in N^I_{to}(l), \forall l \in L_{in}(n)$), and also p_l from the transmitter node t_l of its incoming link l (*i.e.*, from t_l, $\forall l \in L_{in}(n)$), and they are at one-hop distance from node n. For example, in Figure 1, node C, which is the receiver node of link 2, needs P^D from node D, since its incoming link, *i.e.*, link 2, is interfered from the transmission of node D, *i.e.*, $D \in N^I_{to}(2)$, $2 \in L_{in}(C)$, and node D is at one-hop distance from node C.

Remark 4: The number of message passing required in each of the above two algorithms depends on the network topology. The average numbers of message passing in each iteration for Algorithm 1 and Algorithm 2, M_1 and M_2, are obtained as

$$M_1 = H \sum_n \left\{ |L^I_{from}(n)| + \sum_{l\in L_{out}(n)} |N^I_{to}(l)| \right\},$$

where H is the average number of hops that each message traverses, $1 \le H \le 2$, and

$$M_2 = \sum_n \Bigg\{ |L^I_{from}(n)| + |L_{out}(n)| + \sum_{l\in L_{in}(n)} |N^I_{to}(l)| + |L_{in}(n)| \Bigg\}.$$

Since $\sum_n \sum_{l\in L_{out}(n)} |N^I_{to}(l)| = \sum_n \sum_{l\in L_{in}(n)} |N^I_{to}(l)|$ and $\sum_n |L_{out}(n)| = \sum_n |L_{in}(n)| = |L|$, the difference between M_1 and M_2 is obtained by

$$M_1 - M_2 = (H-1) \sum_n \left\{ |L^I_{from}(n)| + \sum_{l\in L_{out}(n)} |N^I_{to}(l)| \right\} - 2|L|.$$

For the network in Figure 1, $M_1 = 49$ and $M_2 = 40$. Note that both M_1 and M_2 only grow linearly with the number of interfering nodes and links.

Remark 5: There are several heuristics that can substantially reduce the amount of message passing:

- **Heuristic 1**: In (22), $\lambda_l(t+1)$ is determined by the difference between its desired data rate $x'_l(t)$ and the experienced data rate $c'_l + \log p_l(t) + \sum_{k\in N^I_{to}(l)} \log\left(1 - P^k(t)\right)$ of link l. Hence, if a node for each link (the transmitter in Algorithm 1 and the receiver in Algorithm 2) can measure its experienced data rate, messages $P^k(t)$ and $p_l(t)$ do *not* have to be exchanged among the nodes. Message passing overhead is substantially reduced, *e.g.*, we now have $M_1 = 23$ and $M_2 = 20$ in the earlier example.
- **Heuristic 2**: Each node needs to transmit messages only if the difference between the current and previous values for each variable exceeds some threshold. For example, node n transmits $\lambda_l(t)$ only if $|\lambda_l(t) - \lambda_l(t-1)| > \epsilon$. By doing this, as the algorithms converges to the optimum, the amount of message passing will be reduced and eventually there will be no message passing.
- **Heuristic 3**: In Algorithm 2, since message passing is conducted within one-hop distance, each node may broadcast its information through data packets by piggybacking to its neighboring nodes. If a node receives information successfully, it can update its information accordingly and perform the algorithm with the updated information. However, some nodes may not receive information successfully due to collision. In this case, those nodes may perform the algorithm with outdated information. Performance of this heuristic is found to be excellent, as shown by simulation in Section V.

We now prove the optimality and convergence of Algorithms 1 and 2. For a rigorous proof, we first need the

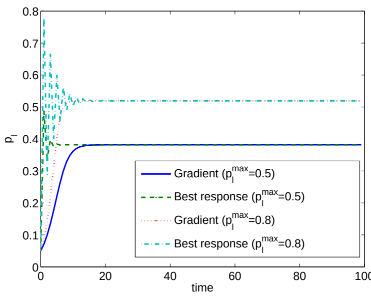

Fig. 3. Comparison of trajectories of $p_l(t)$ in a system with two links.

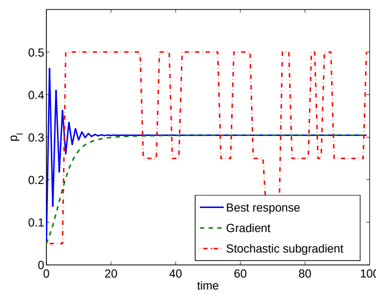

Fig. 4. Comparison of trajectories of $p_l(t)$ in the network in Figure 1, with $p_l^{max} = 0.5$.

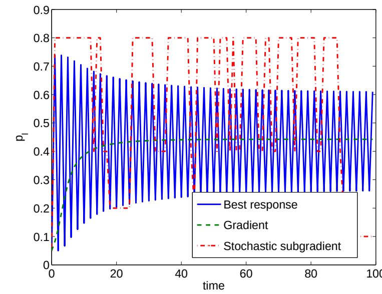

Fig. 5. Comparison of trajectories of $p_l(t)$ in the network in Figure 1, with $p_l^{max} = 0.8$.

following technical condition to have a unique solution to problem (21) at the optimal dual solution:

Assumption 1: At the optimal dual solution $\boldsymbol{\lambda}^*$,

$$\sum_{l \in L_{out}(n)} \lambda_l^* + \sum_{l \in L^I_{from}(n)} \lambda_l^* \neq 0, \quad \forall n.$$

One sufficient condition that satisfies the above assumption is that for each node n, there exists at least one link l, $l \in L_{out}(n) \cup L^I_{from}(n)$ such that $x_l^{'*} < x_l^{'max}$. Hence, it is easily satisfied in most cases.[7]

Theorem 4: Algorithms 1 and 2 converge to the optimal dual solution $\boldsymbol{\lambda}^*$ that solves problem (19). Furthermore, at $\boldsymbol{\lambda}^*$, solutions to problems in (20) and (21): $\mathbf{x}^{'*}$, $\mathbf{p}^*$, and $\mathbf{P}^*$ are the optimal solutions to problem (16) under Assumption 1.

Proof: By Danskin's theorem [21],

$$\frac{\partial Q(\boldsymbol{\lambda})}{\partial \lambda_l} = c_l' + \log p_l + \sum_{k \in N^I_{to}(l)} \log(1 - P^k) - x_l'.$$

Hence, (22) is a subgradient algorithm for the dual problem. There thus exists a step size $\alpha(t)$ (*e.g.*, $\alpha(t) = 1/t$) that guarantees $\boldsymbol{\lambda}(t)$ to converge to the optimal dual solution $\boldsymbol{\lambda}^*$ [22].[8]

Problem (16) is a convex optimization problem, and by Assumption 1, each of problems (20) and (21) has a unique solution at the dual optimal solution $\boldsymbol{\lambda}^*$. Hence, from Property 6.5 in [23], $\mathbf{x}^{'*}$, $\mathbf{p}^*$, and $\mathbf{P}^*$ are the optimal solutions to problem (16). ■

V. Numerical Examples

A. Reverse Engineering Simulation

We first present numerical results for our non-cooperative game model for the BEB protocol. In Figure 3, we consider a network with two links. We provide the results with $p_l^{max} = 0.5$ and $p_l^{max} = 0.8$, respectively. We set $\beta_1 = 0.5$ and $p_l^{min} = 0.05$ for both cases. We compare trajectories of the persistence probability of link 1, $p_1(t)$, which are obtained by (5), *i.e.*, by gradients, and by (11), *i.e.*, by best responses, respectively. Confirming Theorem 2, in the two-link case, the trajectory of the persistence probability obtained by (11) converges to the Nash equilibrium. The trajectory obtained by (5) converges to the same Nash equilibrium, but more smoothly than that obtained by (11).

In Figures 4 and 5, we consider the network in Figure 1, which has six links, with $\beta_l = 0.5$ and $p_l^{min} = 0.05$. In these figures, we also provide trajectories obtained by (4), *i.e.*, by stochastic subgradient.[9] In Figure 4, we set $p_l^{max} = 0.5$. The figure shows that trajectories obtained by (5) and (11) converge to the same equilibrium, which must be a Nash equilibrium from Theorem 2.[10] In Figure 5, we set $p_l^{max} = 0.8$. The figure shows that the trajectory obtained by (11) oscillates between two values. Indeed, as shown in Theorem 2, in general the BEB-MAC Game with the best response strategy may not converge to the Nash equilibrium. However, the trajectory obtained by (5) converges and, by Corollary 1, it converges to the Nash equilibrium.[11]

B. Forward Engineering Simulation I: Probabilistic Model vs. Deterministic Model

In this and the next subsection, we provide simulation results for the proposed algorithms in Section IV, considering the network in Figure 1. In this subsection, we first study the accuracy of our probabilistic NUM formulation based on collision and persistence probabilities, and compare with that of the deterministic approximation approach recently studied in [2], [5], [6].

We first briefly summarize the deterministic approximation approach. We refer readers to [5], [6] for more details. The key idea of this approach is to introduce a contention graph [2], where each vertex represents a link in the network and two vertices are connected with an edge if transmissions from the links in the network corresponding to those two vertices interfere with each other. Hence, if two vertices in the contention graph is connected, then the links in the network corresponding

[7] Note that we do not need this assumption, if we add a penalty term, such as $-\beta \sum_n (P^n)^2$ for a small $\beta > 0$. This makes the objective function of problem (16) strictly concave and, thus, the solution to problem (21) unique.

[8] In practice, a constant step size (*i.e.*, $\alpha(t) = \alpha$, $\forall t$) might be more desirable than a diminishing step size. In this case, $\boldsymbol{\lambda}(t)$ converges to a neighbor [22] of $\boldsymbol{\lambda}^*$, providing an approximately optimal solution.

[9] Since $p_l(t)$ is a stochastic process in this case, we plot its sample path.

[10] Although not shown in the graph, trajectories of the persistence probabilities of the other links also converge.

[11] From this result, it can be seen that the gradient based update strategy as in (5) may have more desirable convergence property than the best response strategy as in (11).

TABLE I

PERFORMANCE OF THE PROPOSED RANDOM ACCESS PROTOCOL (ALGORITHM 1).

Link	1	2	3	4	5	6	Total
p_l	0.5	0.25	0.20	0.25	0.25	0.25	
x_l (analysis)	2.25	0.84	0.84	1.88	0.75	1.13	7.69
x_l (simulation)	2.25	0.84	0.85	1.88	0.75	1.13	7.70
$U_l(x_l)$ (analysis)	0.81	-0.17	-0.17	0.63	-0.29	0.12	0.93
$U_l(x_l)$ (simulation)	0.81	-0.17	-0.17	0.63	-0.29	0.12	0.93

TABLE II

DETERMINISTIC APPROXIMATION WITH $C_{CL_i} = 1$.

Link	1	2	3	4	5	6	Total
x_l (analysis)	3.35	1.67	1.67	3.35	3.35	3.35	16.74
x_l (simulation)	1.55	0.41	0.49	2.23	1.55	1.55	7.78
$U_l(x_l)$ (analysis)	1.21	0.51	0.51	1.21	1.21	1.21	5.86
$U_l(x_l)$ (simulation)	0.44	-0.89	-0.71	0.80	0.44	0.44	0.52

to those vertices cannot transmit data simultaneously without collision. In other words, only one link in the same maximal clique in the contention graph can transmit data successfully at a time. Hence, in the deterministic approximation, each maximum clique is defined as a resource with a finite capacity that is shared by the links belonging to the clique. Capacity of a clique is defined as the maximum value of the sum of time fractions that each link in the clique can transmit data without collision. Consequently, a NUM problem has been formulated as follows, with capacity constraint C_{CL_i} at each maximal clique CL_i:

$$\begin{aligned} &\text{maximize} \sum_l U_l(x_l) \\ &\text{subject to} \sum_{l \in L(CL_i)} \frac{x_l}{c_l} \leq C_{CL_i} \quad \forall i, \\ &\qquad\quad \mathbf{x}^{min} \preceq \mathbf{x} \preceq \mathbf{x}^{max}, \end{aligned} \tag{23}$$

where x_l is the average data rate of link l, c_l is the data rate of link l, $L(CL_i)$ is a set of links that are in maximal clique CL_i, and C_{CL_i} is the capacity of CL_i, ($0 \leq C_{CL_i} \leq 1$). The above problem is a separable and convex optimization problem and similar to the basic NUM problem in (1). Hence, one can easily solve it by using the same algorithm used to solve the basic NUM problem (1) for TCP congestion control.

However, there are several drawbacks in this deterministic approximation. First, in this approach to wireless MAC, each maximal clique generates feedback information to solve the problem. The maximal clique is an artificially constructed entity that does not have a physical controller. Second, in general we do not know the capacity of the maximal clique *a priori*. Third, it is assumed that the average data rate of each link depends only on the fraction of the time that it transmits data. This is implicitly assuming that no collision occurs, if the constraint in problem (23) is satisfied. However, this can happen only when transmissions of links are properly scheduled (probably by a central controller). For a random access protocol, where the time fraction that each link transmits is mapped into persistence probability or backoff window size, there exists a non-negligible probability of collision even when the constraint in problem (23) is satisfied.[12] Hence, the actual performance of the resulting random access protocol can be quite different from the analytical solution obtained by the deterministic approximation.

In the first example, we consider proportional fairness, *i.e.*, the utility function of each link l is defined as

$$U_l(x_l) = \log(x_l).$$

We set $c_l = 10$ (Mbps), $x_l^{min} = 0$, and $x_l^{max} = c_l$ for each link l.

In Table I, we summarize the performance of our random access protocol. We show the average data rate x_l, the achieve utility $U_l(x_l)$, and the optimal persistence probability p_l of each link. We compare analysis results, which are solutions to problem (13), with simulation results, which are obtained by the proposed protocol. The results from simulation are very close to those from analysis.

We now present the performance of the deterministic approximation approach. First of all, to solve problem (23), we need to know the capacity of each clique to ensure the feasibility of the problem, which is in general difficult to know. However, in this simple example, we can easily know that the capacity of each clique is one. In Table II, we first provide the solutions of problem (23) with $C_{CL_i} = 1$, $\forall i$, which are denoted as 'analysis'. We also provide simulation results of the random access MAC protocol with the deterministic approximation. Since x_l/c_l is the time fraction that link l must transmit data to achieve the average data rate x_l, x_l/c_l is interpreted in [6] as the persistence probability of link l. We provide simulation results for the resulting random access MAC protocol in Table II, which are denoted as 'simulation'. The results show that there is a large difference between the solution to problem (23) and the performance of the random access MAC protocol. This discrepancy arises due to the ignorance of collisions in the analytic model of the

[12] In [6], a scheduling-based algorithm that does not require the capacity of the clique and guarantees no collision is also proposed. However, to be implemented in a distributed way, it requires a major approximation, in which each link rounds the obtained solution to either zero or its maximum capacity.

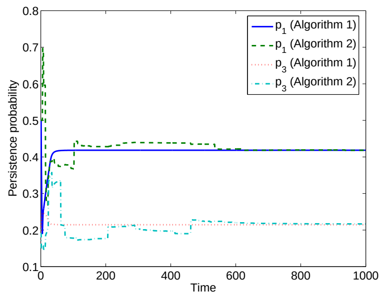

Fig. 6. Comparison of the trajectories of persistence probabilities in two versions of the proposed random access protocols.

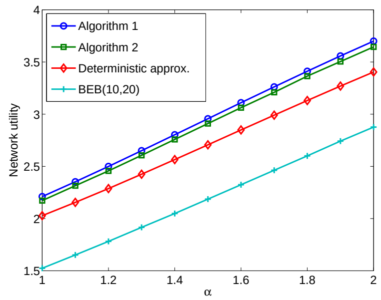

Fig. 7. Comparison of network utilities across four contention based protocols.

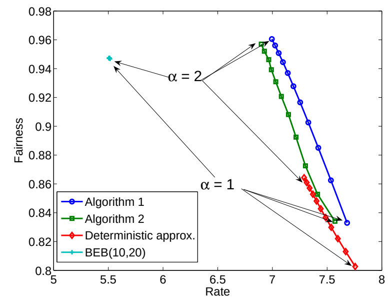

Fig. 8. Comparison of rate-fairness tradeoff across four contention based protocols.

deterministic approximation approach. This implies that the deterministic approximation model is appropriate only for the collision-free scheduling-based protocol that requires a central controller.

C. Forward Engineering Simulation II: Efficiency-Fairness Comparison

This subsection is devoted to comparing the efficiency-fairness tradeoffs of various protocols: our random access protocols (Algorithm 1 and Algorithm 2), the deterministic approximation protocol, and the window-based BEB protocol. We also show how the desired efficiency-fairness tradeoff can be achieved by appropriately adjusting the parameters of utility functions. The results show that our protocols provide much better efficiency-fairness tradeoffs than both the deterministic approximation and BEB protocols.

- In Algorithm 1 and the deterministic approximation, we assume that each node always has up-to-date information to perform the algorithm in each time-slot. For the deterministic approximation, we set the capacity of each clique to be one, which is the true capacity of the clique. In general, the capacity of each clique is not known.
- In Algorithm 2, each node n broadcasts its information (*i.e.*, $P^n(t)$, $p_l(t)$, $\forall l \in L_{out}(n)$, and $\lambda_l(t)$, $\forall l \in L_{in}(n)$) through data packets by piggy-backing to its neighboring nodes (Heuristic 3 in Remark 5). Due to collision, some nodes may not receive it successfully. Hence, in Algorithm 2 simulation, some nodes actually carry out the algorithm with outdated information.
- Window-based BEB protocol's performance highly depends on the choice of maximum and minimum window sizes, W_l^{max} and W_l^{min}. Hence, we first simulate the BEB protocol with various values for maximum and minimum window sizes for each value of α. We present the performance of the BEB protocol when average-performance parameters are chosen: $W_l^{max} = 20$ and $W_l^{min} = 10$.

In this experiment, the utility function for each link l, $U_l(x_l)$ is in the following standard form of concave utility parameterized by α in (2), shifted such that $U_l(x_l^{min}) = 0$ and $U_l(x_l^{max}) = 1$:

$$U_l(x_l) = \frac{x_l^{(1-\alpha)} - x_l^{min(1-\alpha)}}{x_l^{max(1-\alpha)} - x_l^{min(1-\alpha)}}.$$

We set $x_l^{min} = 0.5$ and $x_l^{max} = 5$, $\forall l$, and compare the network utility, and the tradeoff curve between rate and fairness for each protocol, varying the value of α from 1 to 2 with a step size 0.1.[13]

In Figure 6, we compare trajectories of persistence probabilities of links 1 and 3 in Algorithm 1 and Algorithm 2 with $\alpha = 1.5$. The result shows that even though the persistence probability of the link in Algorithm 2 converges slower than that in Algorithm 1, they converge to almost the same value. Hence, we expect that Algorithm 2, with substantial savings in message passing overhead, provides almost the same performance as that of Algorithm 1, as confirmed in the next set of results.

In Figure 7, we compare the network utility achieved by each protocol. This figure shows that our two protocols (Algorithm 1 and Algorithm 2 with reduced message passing overhead) provide almost the same network utility. It also shows that our protocols outperform the other protocols, *i.e.*, BEB and deterministic approximation protocols.

In Figure 8, we show the tradeoff curve of rate and fairness for each protocol. To compare fairness, we use the following fairness index $f(\mathbf{x})$ defined in [24]:

$$f(\mathbf{x}) = \frac{(\sum_l x_l)^2}{|L| \sum_l x_l^2}, \tag{24}$$

where x_l is the average data rate achieved by link l and $|L|$ is the number of links in the network. A higher value of $f(\mathbf{x})$ implies a higher degree of fairness.

For each protocol shown in the graph, the area to the left and below of the tradeoff curve is the achievable region (*i.e.*, every (rate, fairness) point in this region can be obtained), and the area to the right and above of the tradeoff curve is the infeasible region (*i.e.*, it is impossible to have any combination of (rate, fairness) represented by points in this region). Operating on the boundary of the achievable region,

[13]From Lemma 1, we need $\alpha > 1$. For $\alpha = 1$, we used $\alpha = 1.001$ instead.

i.e., the Pareto optimal tradeoff curve, is the best. Since the BEB protocol is a static protocol, it always provides the same efficiency (rate) and fairness regardless of the choice of utility functions. Hence, we cannot control the efficiency-fairness tradeoff in the BEB protocol. The figure shows that our protocol not only provides a higher fairness index but also has a much wider dynamic range of tradeoff than deterministic approximation and BEB protocols.

VI. Conclusion and Future Work

We have reverse-engineered the BEB protocol in IEEE 802.11 as a non-cooperative game where each link is implicitly using stochastic subgradient to maximize a quasi-concave utility function in the form of net reward for successful transmission. Due to the lack of proper feedback mechanisms in the current BEB protocol, such selfish, local actions are not aligned to maximize the network-wide total utility, nor are they guaranteed to converge even though a Nash equilibrium for the MAC game always exists.

Along the forward-engineering direction, we have developed a NUM framework to achieve desired efficiency and fairness and their tradeoff by appropriately adjusting the utility function and solving the resulting NUM problem. Unlike recent publications on MAC NUM, we explicitly model collision and persistence probabilities and allow general utility functions, and the resulting NUM problem is coupled and non-convex optimization. We show how to distributively solve it for global optimality despite such difficulties. The solution then leads to two distributed random access MAC protocols.

We also have compared the performances of our protocols with those of the deterministic approximation protocol and the standard BEB protocol, showing that both of our protocols can provide not only a higher network utility and a larger fairness index, but also a wider dynamic range of the tradeoff curve between efficiency and fairness. Performance guarantee of convergence to the global optimum of the NUM formulation is rigorously proved for the proposed algorithms.

There are several directions to extend our work. For example, other carrier-sensing based MAC protocols in current standards may also be reverse engineered to better understand their properties on efficiency and fairness. Stochastic versions of MAC NUM formulations and solutions need to be developed to incorporate the arrival statistics of packets and sessions. Then MAC protocols can be analyzed and designed using both stochastic stability results in traditional queuing models and utility optimality results in the recent NUM models. Furthermore, using the 'layering as NUM decomposition' approach ([6], [9], [25]) and formulating utilities as functions of end-to-end rates, the methodologies to tackle coupling and non-convexity in this paper can be readily extended to investigate the interactions among transport (*e.g.*, end-to-end rate allocation), network (*e.g.*, routing), link (*e.g.*, medium access), and physical (*e.g.*, power and coding level control) layers in wireless multi-hop networks.

Acknowledgement

We thank Steven Low and Lijun Chen at Caltech and Koushik Kar at RPI for helpful discussions on the topic.

References

[1] IEEE, *Wireless LAN medium access control (MAC) and physical layer (PHY) specifications*, June 1999, iEEE Standard 802.11.

[2] T. Nandagopal, T.-E. Kim, X. Gao, and V. Bharghavan, "Achieving MAC layer fairness in wireless packet networks," in *ACM MobiCom*, 2000, pp. 87–98.

[3] X. L. Huang and B. Bensaou, "On max-min fairness and scheduling in wireless ad-hoc networks: analytical framework and implementation," in *ACM MobiHOC*, 2001, pp. 221–231.

[4] L. Tassiulas and S. Sarkar, "Maxmin fair scheduling in wireless networks," in *IEEE INFOCOM*, vol. 2, 2002, pp. 763–772.

[5] Z. Fang and B. Bensaou, "Fair bandwidth sharing algorithms based on game theory frameworks for wireless ad-hoc networks," in *IEEE INFOCOM*, vol. 2, 2004, pp. 1284–1295.

[6] L. Chen, S. Low, and J. Doyle, "Joint congestion control and media access control design for ad hoc wireless networks," in *IEEE INFOCOM*, 2005.

[7] X. Wang and K. Kar, "Distributed algorithms for max-min fair rate allocation in ALOHA networks," in *41th Annual Allerton Conference*, 2003.

[8] K. Kar, S. Sarkar, and L. Tassiulas, "Achieving proportional fairness using local information in Aloha networks," *IEEE Trans. on Automatic Control*, vol. 49, no. 10, pp. 1858–1862, October 2004.

[9] X. Wang and K. Kar, "Cross-layer rate control for end-to-end proportional fairness in wireless networks with random access," in *ACM Mobihoc*, 2005, pp. 157–168.

[10] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, vol. 49, no. 3, pp. 237–252, March 1998.

[11] S. H. Low and D. E. Lapsley, "Optimization flow control-I: basic algorithm and convergence," *IEEE/ACM Trans. on Networking*, vol. 7, no. 6, pp. 861–874, December 1999.

[12] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Trans. on Networking*, vol. 8, no. 5, pp. 556–567, October 2000.

[13] R. J. La and V. Anantharam, "Utility-based rate control in the Internet for elastic traffic," *IEEE/ACM Trans. on Networking*, vol. 10, no. 2, pp. 272–286, April 2002.

[14] R. T. Rockafellar, *Network Flows and Monotropic Programming*. Athena Scientific, 1998.

[15] S. H. Low, "A duality model of TCP and queue management algorithms," *IEEE/ACM Trans. on Networking*, vol. 11, no. 4, pp. 525–536, August 2003.

[16] H. R. Varian, *Microeconomic Analysis*. W. W. Norton & Company, 1992.

[17] M. J. Osborne and A. Rubinstein, *A course in game theory*. MIT Press, 1994.

[18] D. M. Topkis, "Equilibrium points in nonzero-sum n-person submodular games," *SIAM Journal of Control and Optimization*, vol. 17, no. 6, pp. 773–787, November 1979.

[19] D. D. Yao, "S-modular games, with queueing applications," *Queueing Systems*, vol. 21, pp. 449–475, 1995.

[20] P. Kall and S. W. Wallace, *Stochastic programming*. Wiley, 1994.

[21] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1999.

[22] N. Z. Shor, *Minimization methods for non-differentiable functions*. Springer-Verlag, 1985.

[23] M. Minoux, *Mathematical programming:theory and algorithms*. Wiley, 1986.

[24] R. Jain, D. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," Tech. Rep., September 1984.

[25] M. Chiang, "Balancing transport and physical layers in wireless multihop networks: Jointly optimal congestion control and power control," *IEEE J. Sel. Area Comm.*, vol. 23, no. 1, pp. 104–116, January 2005.