A Game Theoretic Approach to Network Coding

Jason R. Marden and Michelle Effros

***Abstract*—We introduce a game theoretic framework for studying a restricted form of network coding in a general wireless network. The network is fixed and known, and the system performance is measured as the number of wireless transmissions required to meet n unicast demands. Game theory is here employed as a tool for improving distributed network coding solutions. We propose a framework that allows each unicast session to independently adjust his routing decision in response to local information. Specifically, we model the unicast sessions as self-interested decision makers in a noncooperative game. This approach involves designing both local cost functions and decision rules for the unicast sessions so that the resulting collective behavior achieves a desirable system performance in a shared network environment. We compare the performance of the resulting distributed algorithms to the best performance that could be found and implemented using a centralized controller. We focus on the performance of stable solutions – where stability here refers to a form of Nash equilibrium defined below. Results include bounds on the best- and worst-case stable solutions as compared to the optimal centralized solution. We show that our bounds on the best- and worst-case stable performance cannot be improved using cost functions that are independent of the network structure. Results in learning in games prove that the best-case stable solution can be learned by self-interested players with probability approaching 1.**

I. Introduction

The network coding literature treats the design and performance of network codes aimed at goals such as maximizing capacity, minimizing power consumption, or improving the robustness of communication in a network environment. While early results primarily treat the multicast problem – where a single source transmits the same information to all sinks in the network – more recent work changes the focus to problems where multiple independent communication sessions share the network environment. A multiple unicast problem, characterized by a list of source-sink pairs with a distinct information flow to be established from each source to its corresponding sink, is one example of such a coding scenario. Multi-session network coding problems like the multiple unicast problem differ from the single-session network coding problems in that they establish competition between independent information flows for shared network resources.

A spectrum of approaches ranging from centralized control systems to totally distributed design and operation is possible for tackling multi-session coding problems. In centralized approaches like [2], the codes at all nodes of the network are designed by a code designer with access to complete information about the network and all sessions competing for network resources. At the other extreme end of the spectrum are distributed approaches like [3], which tackle large optimization problems by treating nodes or possibly sessions as independent decision makers in a shared network environment. One advantage of distributed algorithms is the savings in computation and coordination achieved by taking a large central optimization problem and dividing it into smaller problems. One disadvantage of this approach is that dividing the central optimization problem in this fashion may induce suboptimal solutions since each individual problem is typically solved with incomplete information about the full problem, and independently choosing the best solution for each subproblem may yield inferior performance for the problem as a whole.

Recently, a branch of noncooperative game theory that focuses on coordination games has been proposed as a tool for cooperative control of distributed systems [4], [5]. Cooperative control focuses on designing autonomous agents to optimize a given global objective. Utilizing game theory for cooperative control or distributed optimization requires the following:

(i) ***Game design:*** The system designer must specify the set of decision makers, which we refer to as agents or players, and their respective actions. Each agent is assigned a local objective function that he selfishly seeks to maximize. An agent's objective function may depend solely on his action, or more generally on the actions of all agents. The compilation of the agents, actions, and objective functions is referred to as a *game*.

(ii) ***Agent decision rules:*** The system designer must specify an iterative procedure for how each agent selects his respective action in response to local information.

The goal is to design *both* the game and the agent decision rules such that the emergent global behavior is desirable with respect to the global objective. In this paper, we primarily focus on game design and appeal to the theory of learning in game for the agent decision rules. The theory of learning in games includes several agent decision rules, also referred to as distributed learning algorithms, that provide guarantees on the emergent global behavior. We direct the readers to [6]–[11] for a comprehensive review.

In this paper, we explore the applicability of noncooperative game theory for designing distributed algorithms for network coding in a wireless network. We formulate the network coding problem as a noncooperative game where the individual unicast sessions are *designed* as self-interested decision makers. We design local objective functions for the unicast sessions with the goal of minimizing total network power consumption using a simple form of network coding. These objective functions are designed without knowledge of the spe-

This paper is an extension of [1]. This work was supported by the Social and Information Sciences Laboratory at California Institute of Technology, DARPA ITMANET grant # W911NF–07–1–0029, and the Lee Center for Advanced Networking at California Institute of Technology.

J. R. Marden is a junior fellow with the Social and Informational Sciences Laboratory, California Institute of Technology, M/C 136–93, Pasadena, CA 91125, marden@caltech.edu.

M. Effros is a professor with the Department of Electrical Engineering, California Institute of Technology, M/C 136–93, Pasadena, CA 91125, effros@caltech.edu.

978-1-4244-4536-3/09/$25.00 ©2009 IEEE

cific network or the demands traversing the network but with the aim of optimizing a centralized network objective when utilizing available distributed learning algorithms. We evaluate the desirability of the objective functions by examining the system performance at the stable solutions, i.e, the equilibria. We focus on the global performance of both the best- and worst-case equilibria.[1]

Because of the inherent structure of the network coding problem, there is an inefficiency that results from decomposing the centralized optimization problem in this fashion. The main result of this paper characterizes this inefficiency by establishing tight worst-case bounds on the global performance of the best- and worst-case equilibria for *any* objective function design that is independent of the network structure. We identify the worst-case networks that give rise to this inefficiency and derive a class of local objective functions that achieve these performance bounds.

Several papers have used game theoretic methods for analyzing network coding problems by viewing either the individual unicasts or individual nodes in the network as selfish decision makers [16]–[18]. Most of these results are only applicable in restricted settings. For example, the authors of [17] focus primarily on single-source multicast with network coding. In that setting, the authors derive a cost mechanism, i.e., a procedure for distributing the cost of a particular edge to the players using that edge, such that a Nash equilibrium exists and the flow allocation at a Nash equilibrium corresponds to the minimum cost. An alternative example is [16], where the authors focus on a simple generalization of the butterfly network with two users. The authors propose cost functions for the two users and show that the desirable capacity achieving solution emerges as a dominant strategy equilibrium point of the game. The goal of this paper is to highlight the applicability of game theory for attaining distributed solutions to a more general class of network coding problems.

II. Background: Noncooperative Games

A. Definitions

We consider a finite strategic-form game. The n players are represented by the set $N := \{1, ..., n\}$. Each player $i \in N$ has a finite action set $\mathcal{A}_i$ and a cost function $J_i : \mathcal{A} \to \mathbb{R}$ where $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_n$ denotes the joint action set. We refer to a finite strategic-form game as "a game," and we sometimes use a single symbol, e.g., G, to represent the entire game, i.e., the player set, N, action sets, $\mathcal{A}_i$, and cost functions J_i.

For an action profile $a = (a_1, a_2, ..., a_n) \in \mathcal{A}$, let a_{-i} denote the profile of player actions *other than* player i, i.e., $a_{-i} = (a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n)$. With this notation, we sometimes write a profile a of actions as (a_i, a_{-i}). Similarly, we may write $J_i(a)$ as $J_i(a_i, a_{-i})$. We also use $\mathcal{A}_{-i} = \prod_{j \neq i} \mathcal{A}_j$ to denote the set of possible collective actions of all players other than player i.

[1]In some settings there are distributed learning algorithms that provide convergence to the best-case equilibrium [12]–[14]; however, the provable convergence rates for these algorithms are not desirable [15]. With this issue in mind, we focus on the global performance of not only the optimal equilibria but also the worst-case equilibria, since trivially it is easier to converge to any equilibrium as opposed to a specific equilibrium. It is important to note that these convergence rates are defined for a general class of games and not specialized for the problem at hand.

The most well known form of an equilibrium is the Nash equilibrium.

Definition 1 (Pure Nash Equilibrium)**.** *An action profile $a^* \in \mathcal{A}$ is called a pure Nash equilibrium if for each player $i \in N$,*

$$J_i(a_i^*, a_{-i}^*) = \min_{a_i \in \mathcal{A}_i} J_i(a_i, a_{-i}^*). \tag{1}$$

A (pure) Nash equilibrium represents a scenario for which no player has an incentive to unilaterally deviate. In a distributed engineering system, a Nash equilibrium represents a stable operating point [5].

B. Cooperative control

Cooperative control problems entail several autonomous players seeking to collectively accomplish a global objective. The network coding problem is one example of a cooperative control problem, where the global objective is for all players to efficiently use a common network by taking advantage of possible coding opportunities. The central challenge in cooperative control problems is to derive local control mechanisms for the individual players such that the players operate in a manner that collectively aids the desired global objective.

In this paper, we focus on the problem of cooperative control using the framework of game theory as proposed in [4]. Let $C : \mathcal{A} \to \mathbb{R}$ represent the true global cost associated with a particular joint action. Applying game theory for distributed control requires designing local player cost functions $J_i : \mathcal{A} \to \mathbb{R}$ that are aligned with the true global cost. As in [19], our goals in designing the local cost functions are:

(i) A Nash equilibrium is guaranteed to exist in the game $(N, \{\mathcal{A}_i\}, \{J_i\})$.
(ii) The Nash equilibria of the game $(N, \{\mathcal{A}_i\}, \{J_i\})$ are efficient with respect to the global cost C.

We gauge the efficiency using the well known worst case measures called the *price of anarchy (PoA)* and *price of stability (PoS)* [20]. Let $\mathcal{G}$ denote a set of games. For any particular game $G \in \mathcal{G}$ let $\mathcal{E}(G)$ denote the set of pure Nash equilibria, $PoA(G)$ denote the price of anarchy, and $PoS(G)$ denote the price of stability for the game G, where

$$PoA(G) := \max_{a \in \ \mathcal{E}(G)} \ \frac{C(a)}{C(a^{\mathrm{opt}})} \tag{2}$$

$$PoS(G) := \min_{a \in \ \mathcal{E}(G)} \ \frac{C(a)}{C(a^{\mathrm{opt}})}, \tag{3}$$

and $a^{\mathrm{opt}} \in \arg\min_{a^* \in \mathcal{A}} C(a^*)$ is any optimal action profile for the game G. We define the price of anarchy and the price of stability for the set of games $\mathcal{G}$ as

$$PoA(\mathcal{G}) \ := \ \sup_{G \in \mathcal{G}} PoA(G), \tag{4}$$

$$PoS(\mathcal{G}) \ := \ \sup_{G \in \mathcal{G}} PoS(G). \tag{5}$$

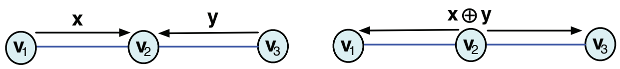

Fig. 1. Illustration of Reverse Carpooling

III. A Simple Wireless Network Coding Problem

We consider the distributed design of network codes for multiple unicasts in a shared wireless network. We restrict our attention to the simplest form of network codes, where any node relaying one message in each direction between a pair of neighboring nodes can reduce the power required for transmission by broadcasting the bit-wise binary sum of the received messages in a single transmission rather than transmitting the two messages sequentially. Each neighbor can then determine its intended message by adding the information that it sent to the received sum, as illustrated in Figure 1. This type of coding is sometimes called "reverse carpooling" since it allows two flows to share a single transmission provided that the two flows traverse the node in opposite directions. The goal of our network code design for this wireless network is to minimize the power required to simultaneously satisfy a given collection of unicast flow demands. For simplicity, we measure the cost of a network coding solution by evaluating the number of transmissions per packet required under steady state flow conditions. The following notation helps make these ideas concrete.

We describe a network by a set $V = \{v_1, ..., v_m\}$ of vertices, or nodes, and for each $v_i \in V$, the neighbors of node v_i are denoted as $\mathcal{N}(v_i) \subseteq V$; each transmission by node v_i is heard by all the nodes in $\mathcal{N}(v_i)$ and only those nodes.

Suppose the network needs to be shared by a finite set of players $N = \{1, ..., n\}$. Each player i represents a single unicast from sender s_i to receiver t_i, where $(s_i, t_i) \in V^2$. A path a_i from source s_i to terminal t_i equals a set of nodes capable of transmitting the information, i.e., $a_i = \{v_1, v_2, \ldots, v_{|a_i|}\}$ where $|a_i|$ denotes the number of nodes in a_i, $v_1 = s_i$, $v_{|a_i|} = t_i$, and $v_{k+1} \in \mathcal{N}(v_k)$ for all $k \in \{1, 2, ..., |a_i| - 1\}$. We use $\mathcal{A}_i$ to denote all paths from s_i to t_i available to player i and $\mathcal{A} = \prod_i \mathcal{A}_i$ to denote the paths available to all players.

For analysis, it is convenient to label each transmission from v by the node from which the information was obtained and the node for which it is next intended so that we can recognize coding opportunities. To that end, let the detailed path of $a_i = \{v_1, v_2, \ldots, v_{|a_i|}\}$ be defined as

$$I(a_i) := \left\{v_1[\emptyset, v_2], v_2[v_1, v_3], \ldots, v_{|a_i|-1}[v_{|a_i|-2}, t_i]\right\}$$

where $v_k[v_{k-1}, v_{k+1}]$ represents a transmission by node v_k to node v_{k+1} of information that was received from node v_{k-1}. The transmission $v_1[\emptyset, v_2]$ represents a transmission by node v_1 to node v_2 of information that originated at node v_1. Notice that for any path a_i there is a unique detailed path $I(a_i)$ consisting of $|a_i| - 1$ elements corresponding to the $|a_i| - 1$ transmissions required to send a packet along that path.

A node that participates in more than one path may have the opportunity to combine messages and save on transmission costs if the paths traverse the node in opposite directions. Suppose players i and j traverse node v in opposite directions, i.e., $v \in a_i \cap a_j$, where player i sends message z_i using the transmission $v[v_i', v_i'']$ and player j sends message z_j using the transmission $v[v_j', v_j'']$. If $v_i' = v_j''$ and $v_j' = v_i''$, then node v can transmit $z_{ij} = z_i \oplus z_j$ and both v_i'' and v_j'' will be able to decode their intended messages. This allows node v to serve players i and j with one transmission instead of two. This is true since node v_i' knows z_i and receives $z_i \oplus z_j$, allowing it to decode its intended message $z_j = z_i \oplus (z_i \oplus z_j)$.

We assume that each vertex $v \in V$ has a cost $C_v : \mathcal{A} \rightarrow \mathbb{R}$ that measures the number of transmissions by that vertex necessary for any routing profile. For our problem, this cost takes on a simplified form that depends only on each player's transmission through the vertex v. Before defining the structure of the cost function for the reverse carpooling setting, we first introduce some notation. For a given routing profile $a \in \mathcal{A}$, let $\sigma(a, v[v_x, v_y])$ be defined as the number of players sending information from v_x to v_y through v, i.e.,

$$\sigma(a, v[v_x, v_y]) := |\{i \in N : v[v_x, v_y] \in I(a_i)\}|. \quad (6)$$

Using only reverse carpooling codes, the transmission cost at node $v \in V$ for profile $a \in \mathcal{A}$ is defined as

$$C_v(a) := \sum_{(v_x, v_y) \in \mathcal{N}(v)^2 : x > y} \max\left\{\sigma\left(a, v\left[v_x, v_y\right]\right), \ \sigma\left(a, v\left[v_y, v_x\right]\right)\right\}, \quad (7)$$

where $v_0 := \emptyset$ and $v_0 \in \mathcal{N}(v)$ for all v. Hence, the power consumption at a vertex depends only on the number of players using each transmission through the vertex. The system cost for a routing profile $a \in \mathcal{A}$ is

$$C(a) := \sum_{v \in V} C_v(a) \quad (8)$$

A global planner would like to use a profile $a \in \mathcal{A}$ that minimizes the system cost.

IV. The Reverse Carpooling Game

In this section we model the reverse carpooling problem as a noncooperative game with player set N, action sets $\mathcal{A}_i$, and system cost C in (8). Modeling the reverse carpooling problem as a noncooperative game involves assigning each player a cost function that is appropriately aligned with the system cost. We henceforth refer to the reverse carpooling problem modeled as a noncooperative game as the reverse carpooling game.

A. Performance bounds for anonymous cost designs

Rather than focusing on the performance of a specific cost design, in this section we seek to characterize the efficiency for a general class of cost functions. Our first results proves that any *anonymous* cost functions, that is any cost function that is independent of the network structure, has at least a price of anarchy of 2. This is a negative result since the system cost associated with each player choosing his shortest path and applying no network coding is also at most twice the cost of the optimal solution.

Let $\bar{\sigma}(a, v[v_x, v_y]) := \{i \in N : v[v_x, v_y] \in I(a_i)\}$ be the set of players using the directed transmission $v[v_x, v_y]$ in action profile a. A player's cost function is anonymous if it takes on the form

$$J_i(a_i, a_{-i}) = \sum_{v[v_x, v_y] \in I(a_i)} f_i\left(\bar{\sigma}(a, v[v_x, v_y]), \bar{\sigma}(a, v[v_y, v_x])\right), \quad (9)$$

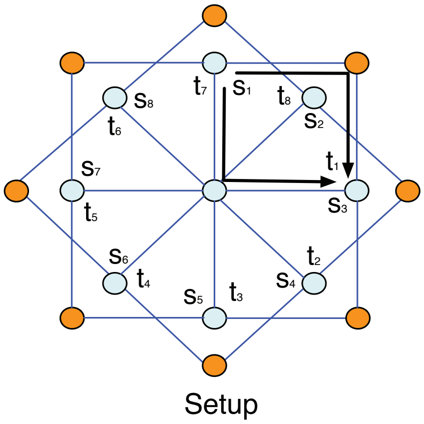

Fig. 2. Problem setup: Price of anarchy of 2 for anonymous cost functions

where $f_i : 2^N \times 2^N \to \mathbb{R}$ is the cost function of player i. Note that the cost that any player i pays for using a directed transmission is not dependent on the particular node; rather, the cost is only a function of the players using the node in each direction.

Theorem 1. *Let $\mathcal{G}$ be the set of all reverse carpooling games where each player has an anonymous cost function of the form (9). The price of anarchy is at least 2.*

Proof: Consider the example in Figure 2. There are eight players with the highlighted sources and terminals. Each player only has the option of taking an exterior path or an interior path, each consisting of 2 segments consisting of m internal nodes. The interior and exterior paths for player 1 are illustrated. If each player's cost function is anonymous and if all players take their exterior paths as shown on the left side of Figure 3, then we have an equilibrium. At this equilibrium, each player is alone on a path consisting of 2 segments. If a player unilaterally switched to his interior path from this equilibrium, the player would also be alone on a path consisting of 2 segments; hence it would result in the same cost since cost functions are anonymous. If each player travels on the exterior, the system cost is $16(m+1)$ as there are no carpooling opportunities for any player. If each player travels through the interior as shown on the right side of Figure 3, each player carpools on the m interior nodes but not the boundary nodes of each edge segment. The system cost of this scenario is $8(m+2)$. Therefore, this example exhibits a price of anarchy of $\frac{16(m+1)}{8(m+2)}$ which can be made arbitrarily close to 2. ∎

Theorem 1 proves that any cost design that is independent of the network structure cannot guarantee a price of anarchy strictly less than 2. It is an open research question to understand how to incorporate attainable information regarding network structure into players' cost functions to improve the price of anarchy. Our second result characterizes a hard constraint on the relationship between the price of anarchy and price of stability for all cost functions.

Theorem 2. *Let $\mathcal{G}$ be the set of all reverse carpooling games, and let each cost function J_i be arbitrary but fixed. If the price of anarchy is γ, then the price of stability $\geq \frac{\gamma+1}{\gamma}$.*

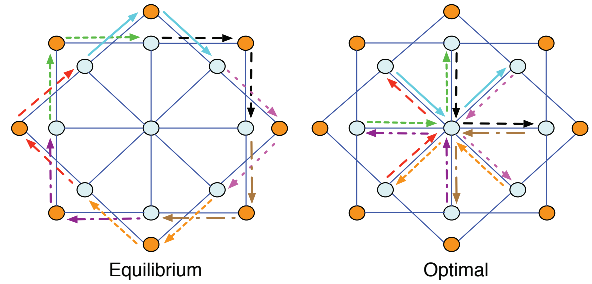

Fig. 3. Equilibrium behavior: Price of anarchy of 2 for anonymous cost functions

Proof: Consider the example in Figure 4. In this example, player 1 has source and terminal (s_1, t_1) and can select either the top path which consists of γm internal nodes or the bottom path which consists of $m-2$ internal nodes, which we denote as T and B respectively. Player 2 has source $s_2 = t_1$ and terminal $t_2 = s_1$ and can also select either the top or bottom path. The cost functions and system cost for the four joint action profiles are highlighted in the following payoff matrices.

P2

P1	T	B
T	$J_1(T,T), J_2(T,T)$	$J_1(T,B), J_2(T,B)$
B	$J_1(B,T), J_2(B,T)$	$J_1(B,B), J_2(B,B)$

Cost Function

P2

P1	T	B
T	$\gamma m + 2$	$\gamma m + m$
B	$\gamma m + m$	m

System Cost C

Cost functions J_1 and J_2 achieve $PoA = \gamma$ by assumption. Consider the action profile $a = (T, T)$. Since the price of anarchy is γ, we know that (T, T) is not a Nash equilibrium. If (T, T) were an equilibrium, then the price of anarchy for this game would be $(\gamma m+2)/m > \gamma$, which yields a contradiction. Therefore, we know that either player 1 or 2 must prefer the bottom route over the top route for this situation. Without loss of generalities, suppose $J_1(B, T) < J_1(T, T)$. Consider the situation where player 2 only has the option of selecting the top path, i.e., $\mathcal{A}_2 = \{T\}$ as in Figure 4. Note that this restriction does not change the cost functions for player 1. For this game, the cost functions and system cost are now

P2

P1	T
T	$J_1(T,T), J_2(T,T)$
B	$J_1(B,T), J_2(B,T)$

Cost Matrix

P2

	T
T	$\gamma m + 2$
B	$\gamma m + m$

System Cost C

This game has a unique Nash equilibrium, namely (B, T). The optimal allocation for this game is (T, T). Therefore, the price of stability for this game is $(\gamma m + m)/(\gamma m + 2)$, which can be made arbitrarily close to $(\gamma + 1)/\gamma$. ∎

While there are many approaches to distributed optimization, this section introduces two worst-case network examples that provide efficiency limitations for any approach that sets the paths of the unicast sessions as the decision variables. It

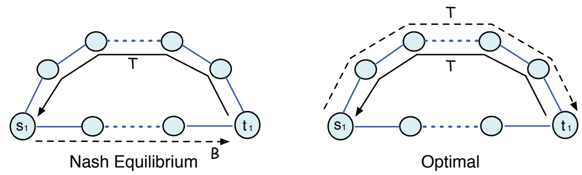

Fig. 4. Worst case example for relationship between the price of anarchy and price of stability.

remains an open question as to whether alternative decompositions, such as designing the nodes in the network (as opposed to the unicast sessions) as the decision makers, could give rise to more efficient designs. However, such designs could require each node to be aware of the entire network structure which is impractical.

B. Cost design that achieves performance bounds

In [1], we identify a particular cost design that achieves the optimal performance bounds set forth in Theorems 1 and 2.

Theorem 3. *Let $\mathcal{G}$ be the set of all reverse carpooling games where the cost function for player i is defined as*

$$J_i(a_i, a_{-i}) = |I(a_i)| + (\alpha - 1)(C(a) - C(a_i^0, a_{-i})) \quad (10)$$

where $\alpha \geq 1$ is a given constant and a_i^0 is the null action for player i and represents the situation where the player sends no information through the network. This cost design guarantees the existence of a Nash equilibrium irrespective of the network, the price of anarchy is

$$PoA(\mathcal{G}; \alpha) = \begin{cases} 2, & \alpha \in [1, 2], \\ \alpha, & \alpha \in (2, \infty), \end{cases}$$

and the price of stability is

$$PoS(\mathcal{G}; \alpha) = \frac{\alpha + 1}{\alpha}.$$

The cost functions in (10) are anonymous and can guarantee a price of anarchy and price of stability pair $(\alpha, \frac{\alpha+1}{\alpha})$ for any $\alpha \in [2, \infty)$. By Theorem 1, we know that one cannot guarantee a price of anarchy strictly less than 2 for any anonymous cost functions. By Theorem 2, we know that for any set of cost functions that yield a price of anarchy of γ, the price of stability must be at least $\frac{\gamma+1}{\gamma}$. Therefore, the cost functions in (10) achieve the optimal bounds on the relationship between the price of anarchy and price of stability.

When $\alpha = 2$, the cost functions in (10) guarantee that the global performance of any equilibrium is within a factor of 2 of the optimal and there exists at least one equilibrium that is within a factor of 1.5 of optimal. These performance guarantees are irrespective of the network structure or the demands traversing the network. Furthermore, since these are worst-case measures, most network structures will achieve significantly better results than indicated by these measures.

C. Implications for more general network coding problems

The general results on optimality of cost functions in this section also provide some impossibility results for more general network coding problems. Consider any network coding problem, unicast or multicast, with various coding options such that reverse carpooling is a special case of the allowable coding structures. Then, the worst case examples used to prove Theorems 1 and 2 are still applicable in this new domain. Therefore, for any network coding problem where reverse carpooling is a special case, it is not possible to construct anonymous player cost functions such that the price of anarchy is strictly less than 2. Notice, however, that in these scenarios the price of anarchy of 2 is not necessarily achievable by shortest paths. Furthermore, it is not possible to guarantee a better relationship between the price of anarchy and price of stability than the relationship set forth in Theorem 2, though this bound is not necessarily achievable for those generalizations.

REFERENCES

[1] J. R. Marden and M. Effros, "The price of selfishness in network coding," in *Workshop on Network Coding, Theory, and Applications*, June 2009, to appear.

[2] D. Traskov, N. Ratnakar, D. S. Lun, R. Koetter, and M. Medard, "Network coding for multiple unicasts: An approach based on linear optimization," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, July 2006.

[3] S. Katti, D. Katabi, W. Hu, H. Rahul, and M. Médard, "The importance of being opportunistic: practical network coding for wireless enviornments," in *43rd Allerton Annual Conference on Communications, Control, and Computing*. Monticello, IL: IEEE, Sept. 2005, invited paper.

[4] J. R. Marden, G. Arslan, and J. S. Shamma, "Connections between cooperative control and potential games," *IEEE Transactions on Systems, Man and Cybernetics. Part B: Cybernetics*, 2009, to appear.

[5] G. Arslan, J. R. Marden, and J. S. Shamma, "Autonomous vehicle-target assignment: a game theoretical formulation," *ASME Journal of Dynamic Systems, Measurement and Control*, vol. 129, pp. 584–596, September 2007.

[6] D. Fudenberg and D. Levine, *The Theory of Learning in Games*. Cambridge, MA: MIT Press, 1998.

[7] H. P. Young, *Individual Strategy and Social Structure*. Princeton, NJ: Princeton University Press, 1998.

[8] ——, *Strategic Learning and its Limits*. Oxford University Press, 2005.

[9] J. Hofbauer and K. Sigmund, *Evolutionary Games and Population Dynamics*. Cambridge, UK: Cambridge University Press, 1998.

[10] J. Weibull, *Evolutionary Game Theory*. Cambridge, MA: MIT Press, 1995.

[11] L. Samuelson, *Evolutionary Games and Equilibrium Selection*. Cambridge, MA: MIT Press, 1997.

[12] L. Blume, "The statistical mechanics of strategic interaction," *Games and Economic Behavior*, vol. 5, pp. 387–424, 1993.

[13] ——, "Population games," in *The Economy as an evolving complex system II*, B. Arthur, S. Durlauf, and D. Lane, Eds. Reading, MA: Addison-Wesley, 1997, pp. 425–460.

[14] J. R. Marden and J. S. Shamma, "Revisiting log-linear learning: Asynchrony, completeness and a payoff-based implementation,," *Games and Economic Behavior*, 2008, submitted.

[15] M. Benaim and W. H. Sandholm, "Logit evolution in potential games: Reversibility, rates of convergence, large deviations, and equilibrium selection," 2007, working paper.

[16] J. Price and T. Javidi, "Network coding games with unicast flows," *IEEE Journal on Selected Areas in Communications*, 2008, to appear.

[17] S. Bhadra, S. Shakkottai, and P. Gupta, "Min-cost selfish multicast with network coding," *IEEE Transactions on Information Theory*, vol. 52, no. 11, pp. 5077–5087, 2006.

[18] D. S. Lun, N. Ratnakar, R. Koetter, M. Medard, E. Ahmed, and H. Lee, "Achieving minimum-cost multicast: A decentralized approach based on network coding," in *Proceedings of IEEE INFOCOM*, 2005, pp. 1607–1617.

[19] J. R. Marden and A. Wierman, "Distributed welfare games," *Operations Research special issue on Computational Economics*, 2008, submitted to.

[20] T. Roughgarden, *Selfish Routing and the Price of Anarchy*. Cambridge, MA, USA: MIT Press, 2005.