Multipath Wireless Network Coding: An Augmented Potential Game Perspective

Vinod Ramaswamy*, Vinith Reddy* Srinivas Shakkottai*, Alex Sprintson* and Natarajan Gautam†
*Dept. of ECE, Texas A&M University
†Dept. of ISE, Texas A&M University
Email: {vinod83, vinith_reddy, sshakkot, spalex, gautam}@tamu.edu

***Abstract*—We consider wireless networks in which multiple paths are available between each source and destination. We allow each source to split traffic among all of its available paths, and ask the question: how do we attain the lowest possible number of transmissions per unit time to support a given traffic matrix? Traffic bound in opposite directions over two wireless hops can utilize the "reverse carpooling" advantage of network coding in order to decrease the number of transmissions used. We call such coded hops as "hyper-links". With the reverse carpooling technique longer paths might be cheaper than shorter ones. However, there is a peculiar situation among sources – the network coding advantage is realized only if there is traffic in both directions of a shared path. We consider the problem of routing with network coding by selfish agents (the sources) as a *potential game,* and develop a method of *state-space augmentation* in which additional agents (the hyper-links) decouple sources' choices from each other by declaring a hyper-link capacity, allowing sources to split their traffic selfishly in a distributed fashion, and then changing the hyper-link capacity based on user actions. Furthermore, each hyper-link has a scheduling constraint in terms of the maximum number of transmissions allowed per unit time. We show that our two-level control scheme is stable, and verify our analytical insights by simulation.**

I. Introduction

There has recently been significant interest in multihop wireless networks, both as a means for basic Internet access, as well as for building specialized sensor networks. However, limited wireless spectrum together with interference and fading pose significant challenges for network designers. The technique of network coding has the potential to improve the throughput and reliability of multihop wireless networks by taking advantage of the broadcast nature of wireless medium.

For example, consider a wireless network coding scheme depicted in Figure 1(a). In this example, two wireless nodes need to exchange packets x_1 and x_2 through a relay node. A simple *store-and-forward* approach needs four transmissions. However, the network coding approach uses a *store-code-and-forward* technique in which the two packets from the clients are combined by means of an XOR operation at the relay and broadcast to both clients simultaneously. The clients can then decode this coded packet (using information stored at clients) to obtain the packets they need.

Katti *et al.* [1] presented a practical network coding architecture, referred to as *COPE*, that implements the above idea while also making use of overheard packets to aid in decoding. Experimental results shown in [1] indicate that the network coding technique may result in a significant improvement in the network throughput.

Research was funded in part by NSF grants CNS-0904520, CNS-0954153, CMMI-0946935, DTRA grant HDTRA1-09-1-0051, and Qatar Telecom, Doha, Qatar.

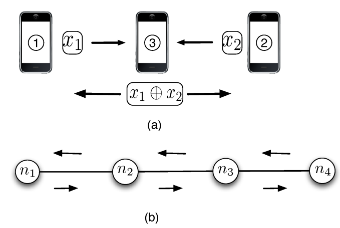


Fig. 1. (a) Wireless Network Coding (b) Reverse carpooling.

Effros *et al.* [2] introduced the strategy of *reverse carpooling* that allows two information flows traveling in opposite directions to share a path. Figure 1(b) shows an example of two connections, from n_1 to n_4 and from n_4 to n_1 that share a common path (n_1, n_2, n_3, n_4). The wireless network coding approach results in a significant (up to 50%) reduction in the number of transmissions for two connections that use reverse carpooling. In particular, once the first connection is established, the second connection (of the same rate) can be established in the opposite direction with little additional cost.

The key challenge in the design of network coding schemes is to maximize the number of *coding opportunities*, where a coding opportunity refers to an event in which at least one transmission can be saved by transmitting a combination of the packets. Insufficient number of coding opportunities may affect the performance of a network coding scheme and is one of the major barriers in realizing the coding advantage. Accordingly, the goal of this paper is to design, analyze, and validate network mechanisms and protocols that improve the performance of the network coding schemes through increasing the number of coding opportunities.

Consider the scenario depicted in Figure 2. We have two sources with equal traffic, each of which is aware of two paths leading to its destination. Each has one path that costs 6 units, while the other path costs 7 units. If both flows use their individually cheaper paths, the total cost is 12 units. However, if both use the more expensive path, since network coding is possible at the node n_2, the total cost is reduced to 11 units. Thus, we see that there is a dilemma here—savings can only be obtained if there is sufficient bi-directional traffic on

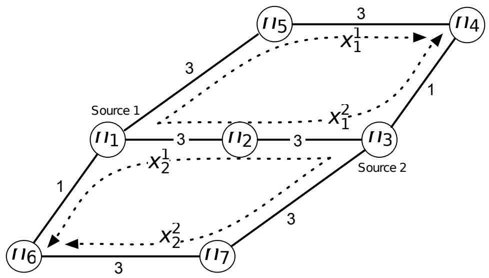


Fig. 2. Each flow has two routes available, one of which permits network coding. The challenge is to ensure that both sources are able to discover the low cost solution.

(n_1, n_2, n_3).

A commonly used framework in the study of routing problems is that of *potential games*. Here, there exits a so-called *potential function*—a scalar value that can be thought of as representing the global utility or cost of the system. The potential function is such that the marginal difference in the payoff received by an agent following from a unilateral change in action is equal to the marginal change in the potential function. Intuitively, it seems that the coupling between an individual agent's payoff and that of the whole system ought to ensure that the system state should converge under myopic learning dynamics. Indeed Sandholm et al. present results under which potential games converge to the optimal solution when it is unique [3], or when the number of players is sufficiently large and a probabilistic approach can be taken [4]. Extensions in the context of systems with inertia [5], as well as finding near-potential games with boundable error [6] have been studied more recently.

However, the problem that we consider presents the issue of a game with a finite number of players that has multiple equilibria, some which have lower cost than others. We can think of the system in Figure 2 as a potential game, with the potential function being the total cost given the traffic splits. However, if each source attempts to learn its optimal traffic split based on the marginal cost that it observes, it could easily choose the inefficient solution. The first mover here is clearly at a disadvantage as it essentially creates the route that the other can piggyback upon (in a reverse direction). Our challenge in this paper is to extend the potential game framework to eliminate the first-mover disadvantage. As we will discuss in Section I-B, a main contribution of this paper is the development of the idea of *state space augmentation* in potential games as a way of promoting optimal coordination in such situations.

A. Related Work

Network coding was initiated by a seminal paper by Ahlswede *et al.* [7] and since then has attracted a significant interest from the research community. The network coding technique was utilized in a wireless system developed by Katabi *et al.* [1]. The proposed architecture, referred to as COPE, contains a special network coding layer between the IP and MAC layers. Sagduyu and Ephremides [8] focused on the applications of network coding in simple path topologies (referred to in [8] as *tandem* networks) and formulated several related cross-layer optimization problems. Similarly, [9] considered the problem of utility maximization when network coding is possible. However, their focus is on opportunistic coding as opposed to *creating* coding opportunities that we focus on. The practicality of utilizing network coding over multiple paths for low latency applications was demonstrated by Feng *et al.* [10].

Sengupta *et al.* [11] consider a very similar problem to ours, and present a general linear programming formulation to solve it. However, their objective was to find a centralized solution, as opposed to the distributed learning dynamics that we seek. Das *et al.* [12] proposed a new framework called "context based routing" in multihop wireless networks that enables sources to choose routes that increase coding opportunities. They proposed a heuristic algorithm that measures the imbalance between flows in opposite directions, and if this imbalance is greater than 25%, provides a discount of 25% to the smaller flow. This has the effect of incentivizing equal bidirectional flows, resulting in multiple coding opportunities. Our objective is similar, but we develop iterated distributed decision making methods that trade off a potential increase in cost of longer paths, with the potential cost reduction due to enhanced coding opportunities.

Marden *et al.* [13] considered a similar problem to ours, but unlike our focus on how to align user incentives, their attention was largely on the efficiency loss of the Nash equilibrium attained. Thus, they considered the system as a potential game, and considered the worst case and best case equilibria that the system might converge to. They showed that under the potential game framework, the best case Nash equilibrium can be optimal, while the cost of the worst case Nash equilibrium can be unboundedly large. To the best of our knowledge, the initial version of our work that was presented at a conference [14] was the first to propose a distributed algorithm that attains the optimal solution. The underlying idea of state-space augmentation was presented in that work. In parallel with our work, Marden *et al.* [15] described a "state-based game," which also augments the potential game framework with additional state, and later used the framework in the context of consensus formation in networks [16]. Also in parallel work, ParandehGheibi et al. [17] presented an optimal solution specific to the network coding problem using classical Lagrange multiplier ideas. In contrast to their work, we present a new technique whereby we modify the potential function seen by players in order to ensure that they take system-wide optimal decisions. From a methodological standpoint, we believe that our approach can find application in equilibrium selection in a wide range of coordination problems *(eg.* in understanding how altruistic behavior can alter the set of achievable equilibria).

B. Main Results

The key contribution of this research is a distributed two-level control scheme that would iteratively lead the sources to discover the appropriate splits for their traffic among multiple paths. In a traditional potential game approach, the matrix of traffic splits of the different flows would be the state of the system. In our work, we introduce the idea of *augmenting* the state space with additional variables that are controlled separately by *augmented agents*. Unlike Lagrange multipliers, the additional state variables need not correspond to a constraint set. Instead, these augmented variables are used to modify the potential function seen by the original agents in such a way that they are directed towards the optimal equilibrium. In this sense, the idea can be thought of as a generalized Lagrange multiplier. We also illustrate that our approach can coexist with the usual Lagrange multiplier approach to handle constraints.

We explore the idea of state space augmentation using the network coding problem. Here, at one timescale we have sources that selfishly choose to split their traffic across available multiple paths using marginal costs on each path to direct their actions. The learning dynamics that they use are consistent with a potential game approach. However, the costs that they see are set by augmented agents as well as Lagrange multipliers, both of which operate at a different timescale from the source dynamics. The augmented agents in our problem are so-called *hyper-links* that consist of a node and two links over which the node can broadcast using network coding, as exemplified by the node n_2 in Figure 2. These hyperlinks provide a *rebate* for usage of the coded path in order to incentivize flows to explore their usage. The rebate takes the form of a *hyper-link capacity,* which simply means that the the hyper-link does not charge the flows for usage up to its chosen capacity. Besides the need to encourage flows to explore codable paths, we also impose a constraint that each link has a maximum rate that it can support due to scheduling or spectrum limitations. This constraint is realized via a Lagrange multiplier approach.

Hence, our approach consists of two control loops, with the inner employing well-studied learning algorithms such as BNN dynamics [18] assuming a fixed rebate by hyperlinks, as well as a price that corresponds to the Lagrange multiplier. The outer loop consists of gradient-type controllers that modify the rebate and price, respectively. All controllers only use local information for their decisions. The process of iteration continues until the entire network has reached local minimum which, since our formulation is convex, is also the socially optimal solution. We prove that this process is globally asymptotically stable. Note, however, that our optimality result involves two nested asymptotic results, so we cannot implement the idea directly. In practice, we have can only run each loop for a finite number of steps before switching to the other.

We illustrate this approach using numerical experiments. For comparison, we numerically solve the problem as a linear program to find the optimal solution. The experiments indicate that: the convergence of the augmented potential game is fast; the costs are reduced significantly upon using network coding; more expensive paths before network coding became cheaper and shortest paths were not necessarily optimal. Thus, the iterative algorithm that we develop performs well in practice.

This paper is organized as follows: Section-II develops a system model and problem formulation assuming no scheduling constraints on the maximum number of transmissions at each node. In Section-III, we introduce the concept of *hyper-links*. In Section-IV we reformulate the problem with constraints on peak transmissions from each node and present a bi-level distributed controller - a combination of rate controller and hyper-link controller- to solve the problem. The rate controller is presented Section-V and the hyper-link controller is presented in Section-VI. Section-VII contains simulation results and Section-VIII concludes the paper.

II. System Overview

Our objective is to design a distributed multi-path network coding system for multiple unicast flows traversing a shared wireless network. We model the communication network as a graph $G(\mathcal{N}, E)$, where $\mathcal{N}$ is a set of network nodes and E is a set of wireless links. For each link $(n_i, n_j) \in E$, where n_i and n_j are any two nodes, there exists a wireless channel that allows the node n_i to transmit information to the node n_j. Each link (n_i, n_j) is associated with a cost α_{ij}. The value of α_{ij} captures the cost (in expected number of required transmissions) of sending a packet successfully from n_i to n_j. Due to the broadcast nature of the wireless channels, the node n_i can transmit to two neighbors n_j and n_k simultaneously at a cost $\max\{\alpha_{ij}, \alpha_{ik}\}$.

In wireless networks, even though broadcasting enables simultaneous transmission to neighboring nodes, it also acts as interference at those nodes which are listening to some node other than the broadcasting node. This type of interference in wireless networks, called Co-Channel Interference, is handled by upper MAC protocols (for example CSMA) which schedules transmission periods of links in the network such that interference is minimized. We assume that a perfect schedule of wireless links is given to us and, therefore, there is no interference at the receivers. However, this imposes a constraint on the maximum number of transmissions per unit time on the nodes. In this section, we develop a basic framework, while ignoring these scheduling constraints. We will include these constraints in Section IV.

We assume that the network supports flows $\{1, 2, \ldots, \}$, where each flow is associated with a source and destination node. Each flow i is also associated with several paths $\{P_i^1, P_i^2, \ldots\}$ that connect its source and destination nodes. Our goal is to build a distributed traffic management scheme in which the source node of each flow i can split its traffic, x_i (packets per unit time), among multiple different paths, so as to reduce the *total number of transmissions per unit time* required to support given traffic demands. Note that on some of these paths there might be a possibility of network coding.

We will first examine a simple network with coding opportunities and derive system cost associated with the network, in terms of the total number of transmissions required. Then we will study how the coding helps in reducing the system cost.

Example 1: Consider the network depicted on Figure 2. The network supports three flows: (i) flow 1 from n_1 to n_4, (ii) flow 2 from n_4 to n_6, and (iii) flow 3 from n_5 to n_1. We denote by

x_i the traffic associated with flow i, $1 \le i \le 3$. Suppose that the packets that belong to flow 1 can be sent over two paths (n_1, n_2, n_3, n_4) and (n_1, n_2, n_5, n_4). We denote these paths by P_1^1 and P_1^2. The traffic split on paths P_1^1 and P_1^2 is given by x_1^1 and x_1^2, respectively, such that $x_1^1 + x_1^2 = x_1$. Similarly, flow 2 can be sent over two paths $P_2^1 = (n_4, n_3, n_2, n_6)$ and $P_2^2 = (n_4, n_8, n_6)$ at rates x_2^1 and x_2^2, such that $x_2^1 + x_2^2 = x_2$. Finally, flow 3 can be sent over two paths $P_3^1 = (n_5, n_7, n_1)$ and $P_3^2 = (n_5, n_2, n_1)$, at rates x_3^1 and x_3^2, with sum x_3.

Note that path $P_1^2 = (n_1, n_2, n_5, n_4)$ of flow 1 and path $P_3^2 = (n_5, n_2, n_1)$ of flow 3 share two links (n_1, n_2) and (n_2, n_5) in the opposite directions. Thus, the packets sent along these two paths can benefit from reverse carpooling. Specifically, the node n_2 can combine packets of flow 1 received from the node n_1 and packets of flow 3 received from the node n_5. Similarly, the node n_3 can combine packets of flow 1 received from the node n_2 and packets of flow 2 received from the node n_4. Note that the cost saving at the node n_2 is proportional to $\min\{x_1^2, x_3^2\}$, while the saving at the node n_3 is proportional to $\min\{x_1^1, x_2^1\}$. Recall that we are ignoring scheduling constraints in this section.

The cost (transmissions per unit time) at the node n_2 when coding is enabled is

$$
\begin{aligned}
C_{n_2}(x_1^2, x_3^2) = \quad & \max\{\alpha_{21}, \alpha_{25}\} \min\{x_1^2, x_3^2\} \\
& + \alpha_{25}(x_1^2 - \min\{x_1^2, x_3^2\}) \\
& + \alpha_{21}(x_3^2 - \min\{x_1^2, x_3^2\}).
\end{aligned} \quad (1)
$$

Here, the first term on the right is the cost incurred due to coding at the node n_2. This is because a coded packet from n_2 is broadcast to both destination nodes, n_1 and n_5, and so the cost per packet is $\max\{\alpha_{21}, \alpha_{25}\}$. The second and third term are "overflow" terms. Since it is possible that $x_1^2 \neq x_3^2$, the remaining flow of the larger (that cannot be encoded because of the lack of flow in the opposite direction) is sent without coding at the regular link cost.

The cost at the node n_2, given by (1), can be re-written as shown below:

$$
\begin{aligned}
C_{n_2}(x_1^2, x_3^2) &= \alpha_{25} x_1^2 \quad + \alpha_{21} x_3^2 + \Big\{ \max\{\alpha_{21}, \alpha_{25}\} \\
& \qquad - (\alpha_{21} + \alpha_{25}) \Big\} \min\{x_1^2, x_3^2\}.
\end{aligned}
$$

Using the fact that $\max\{x_1, x_2\} + \min\{x_1, x_2\} = x_1 + x_2$, we obtain

$$
\begin{aligned}
C_{n_2}(x_1^2, x_3^2) &= \alpha_{25} x_1^2 + \alpha_{21} x_3^2 \\
&- \min\{\alpha_{21}, \alpha_{25}\} \min\{x_1^2, x_3^2\}.
\end{aligned} \quad (2)
$$

The above equation can be interpreted as the cost at the node n_2 without coding minus the savings obtained when coding is used. Thus, the cost saved at the node n_2 due to network coding is $\min\{\alpha_{21}, \alpha_{25}\} \min\{x_1^2, x_3^2\}$. Similarly, for the node n_3 the cost saved is $\min\{\alpha_{32}, \alpha_{34}\} \min\{x_1^1, x_2^1\}$.

The total system cost can be expressed as:

$$
\begin{aligned}
C(X) = \sum_{i=1}^{3} \sum_{j=1}^{2} \beta_i^j x_i^j - \quad & \min\{\alpha_{21}, \alpha_{25}\} \min\{x_1^2, x_3^2\} \\
- \quad & \min\{\alpha_{32}, \alpha_{34}\} \min\{x_1^1, x_2^1\},
\end{aligned} \quad (3)
$$

where $X = \{x_1^1, x_1^2, x_2^1, x_2^2, x_3^1, x_3^2\}$ is the state of the system and β_i^j is the uncoded path cost (equal to the sum of the link costs on the path) j used by flow i. For example, $\beta_1^1 = \alpha_{12} + \alpha_{23} + \alpha_{34}$, for path $P_1^1 = (n_1, n_2, n_3, n_4)$. Thus, the first term on the right in (3) is the total cost of the system without any coding, while the second and third terms are the savings obtained by coding at nodes n_2 and n_3.

In the next subsection, we present a system model and derive a general expression for system cost. Then we formulate an optimization problem which minimizes system cost by finding an optimal traffic split of each flow, over the multiple paths available to them.

A. System Model

Our system model consists of a set of nodes $\mathcal{N} = \{n_1, \ldots, n_N\}$ and a set of flows $\mathcal{F} = \{1, \ldots, F\}$. Each flow, $f \in \mathcal{F}$ is defined as a tuple (n_f^s, n_f^d, x_f), where $n_f^s \in \mathcal{N}$ is the source node, $n_f^d \in \mathcal{N}$ is the destination node, and x_f *packets/sec* is its traffic demand. A flow may be associated with multiple paths connecting its source and destination nodes. Let P_f be the number of such paths available to flow f and x_f^s be the traffic sent by the flow over path s associated with it. Then, $\sum_{s=1}^{P_f} x_f^s = x_f$. Let $\mathbf{x}_f = \{x_f^1, \cdots, x_f^{T_f}\}$ represent a traffic split of flow f. Then, the state of the system X is defined as a set of traffic splits of all flows in the system. i.e $X = \{\mathbf{x}_1, \cdots, \mathbf{x}_F\}$.

A node participating in more than one path may have the opportunity to combine traffic and save on transmission if the paths traverse the node in reverse directions. Suppose paths q and r, associated with flow i and j respectively, traverse the node n_k in reverse directions. Assume the node n_k receives packets belonging to flow i which are sent over path q and transmits those packets to the node n_i. Similarly, it collects packets belonging to flow j traversing over path r and forwards them to the node n_j. Thus, the packets sent along these paths can benefit from reverse carpooling and there exists a coding opportunity for flows i and j at the node n_k. We represent this coding opportunity at the node n_k, which is associated with two neighboring nodes and two flows, as $h = n_k[(i, q, n_i), (j, r, n_j)]$[1]. For example, consider the network shown on Figure 2. In this network, the coding opportunity available at the node n_2 can be represented as $n_2[(1, P_1^2, n_3), (2, P_2^1, n_1)]$. Finally, we assume that H such coding opportunities are present in the system.

[1] In all the futute references of h, we may assume that it is associated with $n_k(h)[(i(h), q(h), n_i(h)), (j(h), r(h), n_j(h))]$. For notational convenience, we may drop the reference to h in the previous representation and simply use $n_k[(i, q, n_i), (j, r, n_j)]$

From (2), the cost (transmissions per unit time) at the node n_k after coding enabled is given by

$$C_{n_k}(x_i^q, x_j^r) = \alpha_{ki}x_i^q + \alpha_{kj}x_j^r - \min\{\alpha_{ki}, \alpha_{kj}\}\min\{x_i^q, x_j^r\}. \quad (4)$$

The total system cost can be expressed as:

$$\mathbf{C}(X) = \sum_{f=1}^{F}\sum_{p=1}^{P_f}\beta_f^p x_f^p - \sum_{h=1}^{H}\min\{\alpha_{ki}, \alpha_{kj}\}\min\{x_i^q, x_j^r\} \quad (5)$$

where X is the state of the system and β_f^p is the uncoded path cost (equal to the sum of the link costs on the path) p used by flow f.

Our goal is to build a distributed traffic management scheme in which the source node of each flow f can split its traffic, x_i (packets per unit time), among multiple different paths, so as to reduce the system cost (5), *total number of transmissions per unit time* required to support a given traffic demands. We formulate the objective of minimizing cost, subject to the traffic requirements of each flow, as an optimization problem given below:

$$\min_{X\geq 0} \quad \mathbf{C}(X), \quad (6)$$

$$\text{subject to} \quad \sum_{p=1}^{P_f} x_f^p = x_f \qquad f = 1, \ldots, F.$$

The problem poses major challenges due to the need to achieve a certain degree of coordination among the flows. For example, for the network depicted in Figure 2, increasing of the value of x_3^2 (the decision made by the node n_5) will result in a system-wide cost reduction only if it is accompanied by the increase in the value of x_1^2. In the next section, we develop a distributed traffic management scheme, that does not require any coordination among flows on deciding their traffic splits.

III. Augmented State Space and Hyper-Links

The optimization problem in (6) can be solved efficiently in a centralized manner. But centralized implementations are not practical in large and complex systems. In this section, we propose a simple way of decomposing it into subproblems that can be solved in a decentralized fashion. We do this by means of adding extra state variables to the system, which we refer to as *state-space augmentation*.

It can be observed from (5) that decisions of flows i and j are coupled through the term $\min(x_i^q, x_j^r)$. In general, for any given x_i^q and x_j^r, this term can be expressed as an optimal value of the following optimization problem,

$$\min\{x_i^q, x_j^r\} = \max_{y>0}\Big(y - \lambda_1(y - \min\{y, x_i^q\}) - \lambda_2(y - \min\{y, x_j^r\})\Big), \quad (7)$$

where $\lambda_1, \lambda_2 \geq 1$ are any arbitrary constants. Note that the right hand side of the above equality does not have any coupling term, due to the presence of the augmented variable y. Therefore, we can convert the coupled problem (6) into a de-coupled one by replacing each 'coupled' term $(\min\{x_i^q, x_j^r\})$ with an equivalent 'de-coupled' expression from (7). Since each coupling term is associated with a coding opportunity h, the augmented variable y_h is introduced in association with each coding opportunity. Let $Y = \{y_1, y_2, \cdots, y_H\}$. Now, define $\mathbf{C}(X, Y)$ as

$$\mathbf{C}(X,Y) = \sum_{f=1}^{F}\sum_{p=1}^{P_f}\beta_f^s x_f^s - \sum_{h=1}^{H}(\min\{\alpha_{ki}, \alpha_{kj}\})y_h + \sum_{h=1}^{H}\left(\omega_{1h}(y_h - \min\{y_h, x_i^q\}) + \omega_{2h}(y_h - \min\{y_h, x_j^r\})\right),$$

where $\omega_{1h}, \omega_{2h} \geq \min\{\alpha_{ki}, \alpha_{kj}\}$ are any arbitrary constants. It can be seen that the cost function (5) can be re-written as

$$\mathbf{C}(X) = \min_{Y\geq 0}\mathbf{C}(X, Y). \quad (8)$$

Choosing $\omega_{1h} = \alpha_{ki}$ and $\omega_{2h} = \alpha_{kj}$, we get

$$\mathbf{C}(X,Y) = \sum_{f=1}^{F}\sum_{p=1}^{P_f}\beta_f^s x_f^s - \sum_{h=1}^{H}(\min\{\alpha_{ki}, \alpha_{kj}\})y_h + \sum_{h=1}^{H}\left(\alpha_{ki}(y_h - \min\{y_h, x_i^q\}) + \alpha_{kj}(y_h - \min\{y_h, x_j^r\})\right). \quad (9)$$

The cost function has thus been augmented using the variables y_h. For any fixed value of Y, the cost function only depends on X, and the sources can attempt to modify X find their individually lowest cost solution. The augmented variables Y can then be modified to change the cost function. In Sections V–VI we will formally show how this is accomplished. We now show that our choices for ω's lead to an appealing interpretation for the function $\mathbf{C}(X, Y)$.

Consider coding opportunity $h = n_k[(i, q, n_i), (j, r, n_j)]$, where the node n_k encodes packets coming from i^{th} and j^{th} flows, and then broadcast them to nodes n_i and n_j respectively. Grouping the terms associated with coding opportunity h in (9), we get

$$\begin{aligned} C(h) &= \alpha_{ki}x_i^q + \alpha_{kj}x_j^r - \min\{\alpha_{ki}, \alpha_{kj}\}y_h + \\ &\quad \alpha_{ki}(y_h - \min\{y_h, x_i^q\}) + \alpha_{kj}(y_h - \min\{y_h, x_j^r\}), \\ &= \max\{\alpha_{ki}, \alpha_{kj}\}y_h + \alpha_{ki}(x_i^q - \min\{x_i^q, y_h\}) \\ &\quad + \alpha_{kj}(x_j^r - \min\{x_j^r, y_h\}). \end{aligned} \quad (10)$$

In the above expression, $C(h)$, the first term corresponds to the cost of broadcasting coded traffic, if we restrict the total coded (broadcast) traffic between the two flows at the node n_k to be less or equal to y_h, and the last two terms are the transmission costs associated with the remaining uncoded traffic. This leads to the concept of *hyper-link*, which can be thought of as a broadcast link with capacity y_h. It is composed of physical links (n_k, n_i) and (n_k, n_j) and carries only encoded traffic from flows i and j. And the remaining uncoded traffic is sent through uni-cast links (n_k, n_i) and (n_k, n_j) respectively. Formally, a hyper-link and a hyper-path are defined as follows:

Definition 1: A *hyper-link* is a broadcast-link composed of three nodes and two flows. A hyper-link $h = n_k[(i, q, n_i), (j, r, n_j)]$ at the node n_k can encode packets belonging to flow i (sending packets on path q) with flow j (sending packets on path r). Here, the nodes n_i and n_j are the next-hop neighbors of n_k; for flow i along path q and for flow j along path r, respectively. Also, y_h denotes capacity of the hyper-link (in packets per unit time).

A *hyper-path* $p \in \mathcal{S}_i$ between source n_i^s and destination n_i^d is a virtual path over a physical path between n_i^s and n_i^d. A hyper-path contains zero or more hyper-links on it and at *each node* on the underlying physical path there can be atmost one hyper-link. It follows that the set of all paths are a subset of the hyper-paths.

The cost at hyper-link h, given by (10), can be re-written as:

$$C(h) = \alpha_{ki} x_i^q + \alpha_{kj} x_j^r - T(h), \text{ where} \quad (11)$$

$$T(h) = \alpha_{ki} \min\{x_i^q, y_h\} + \alpha_{kj} \min\{x_j^r, y_h\} - \max\{\alpha_{ki}, \alpha_{kj}\} y_h. \quad (12)$$

Recall that the first two cost terms are the total cost at the node n_k when coding is disabled. The remaining cost, $T(h)$ can be thought of as the *rebate* obtained by using hyper-link $h = n_k[(i, q, n_i), (j, r, n_j)]$. Note that the rebate could be *negative* (hence adding to the total cost), which might happen when one of the flow rate is 0 and the other flow rate is less than the hyper-link capacity.

Now the function $\mathbf{C}(X, Y)$ in (9) can be written as follows:

$$C(X, Y) = \sum_{f=1}^{F} \sum_{p=1}^{P_f} \beta_f^s x_f^s - \sum_{h=1}^{H} T(h), \quad (13)$$

which represents the total system cost without coding minus the total rebate of all the hyper-links. Here, $C(X, Y)$ - total number of transmissions per unit time required to support a given traffic load- is the system cost given the system state (X, Y), where X is the set of traffic vectors of all flows in the system and Y is set of hyper-link capacities. Our objective is to minimize the cost function which can be formally stated as

$$\min_{X, Y \geq 0} \quad \mathbf{C}(X, Y)$$

$$\text{subject to} \quad \sum_{p=1}^{P_f} x_f^p = x_f \quad \forall f = 1, \cdots, F. \quad (14)$$

In the next section, we will also account for the fact that the transmission rate of each node is limited due to scheduling constraints.

IV. Peak Transmission Constraints

In a practical scenario, the maximum number of transmissions per unit time from a wireless node is limited by scheduling. In this section, we assume that the schedule has been predetermined, and imposes a constraint on the maximum amount of traffic that can be accommodated on any particular link. In doing so, we will illustrate the fact that the state space augmentation can be used in conjunction with Lagrange multiplier that enforces a constraint. reformulate problem (14) taking into account the transmission constraints at each node.

Let R_{ki}^{fp} be a routing variable. It takes a value equal to 1 if any path p associated with flow f passes through link (n_k, n_i) and otherwise 0. Similarly, define Z_k^h which takes 1 if hyper-link h is associated with the node n_k and otherwise 0. Let T_k be the maximum number of allowable transmissions per unit time at the node n_k. Then, at each node n_k, the total number of uncoded transmissions minus the saved number of transmissions (using hyper-links) should be less than or equal to T_k. Therefore,

$$\sum_{i=1}^{N} \sum_{f=1}^{F} \sum_{p=1}^{P_f} R_{ki}^{fp} \alpha_{ki} x_f^p - \sum_{h=1}^{H} Z_k^h T(h) \leq T_k. \quad \forall n_k \in \mathcal{N}.$$

Now, incorporating these constraints on transmission rate, the problem (14) can be re-written as

$$\min_{X \geq 0, Y \geq 0} \quad \mathbf{C}(X, Y) = \sum_{f=1}^{F} \sum_{p=1}^{P_f} \beta_f^p x_f^p - \sum_{h=1}^{H} T(h),$$

$$\text{subject to} \quad \sum_{p=1}^{P_f} x_f^p = x_f, \qquad \forall f = 1, \cdots, F, \quad (15)$$

$$\sum_{i=1}^{N} \sum_{f=1}^{F} \sum_{p=1}^{P_f} R_{ki}^{fp} \alpha_{ki} x_f^p - \sum_{h=1}^{H} Z_k^h T(h) \leq T_k, \quad \forall k = 1, \cdots, N, \quad (16)$$

where X is the set of traffic vectors of all flows in the system and Y is set of hyper-link capacities. Note that the augmented cost $C(X, Y)$ is jointly convex in X and Y. The constraint sets are also convex. Therefore, the above problem is convex. We assume that the feasible sets of the above problem -set of traffic vectors X and set of hyper-link capacities Y which satisfy both traffic demands (15) and peak transmission constraints (16)- is nonempty. We can use dual decomposition techniques to construct a distributed algorithm to solve this problem. The Lagrangian function is

$$\mathbb{C}(X, Y, \Sigma) = \sum_{f=1}^{F} \sum_{p=1}^{P_f} \beta_f^p x_f^p - \sum_{h=1}^{H} T(h) + \sum_{k=1}^{N} \sigma_k V_k$$

$$\text{where} \quad V_k = \left(\sum_{f=1}^{F} \sum_{p=1}^{P_f} R_{ki}^{fp} \alpha_{ki} x_f^s - \sum_{h=1}^{H} Z_k^h T(h) - T_k \right). \quad (17)$$

Note that σ_k is a non-negative Lagrange multiplier associated with the transmission constraint of the node n_k. We can interpret σ_k as the 'price' charged by the node n_k for each transmission. Let $\Sigma = [\sigma_1, \cdots, \sigma_N]$ be a set of node-prices.

We define $\mathbb{C}(X, Y, \Sigma)$ as our new system function given the system state (X, Y, Σ), where X is the set of traffic vectors of all flows in the system, Y is the set of hyper-link capacities and Σ is the set of node-prices. Our objective is find an optimal state of the problem given below.

$$\max_{\Sigma \geq 0} \quad \min_{X,Y \geq 0} \mathbb{C}(X, Y, \Sigma),$$
$$\sum_{f=1}^{F} x_f^p = x_f, \quad \forall f = 1, \cdots, F.$$

We propose a bi-level distributed iterative algorithm to find an optimal state for the above problem.

1) **Traffic Splitting:** In this phase, each source node finds the optimum traffic assignment given the hyper-link capacities and node-prices. For any given (Y, Σ),

$$\textbf{TS}: \quad \min_{X \geq 0} \mathbb{C}(X, Y, \Sigma), \qquad \sum_{f=1}^{F} x_f^p = x_f \quad f = 1, \cdots, F.$$

We model this part as a traditional potential game. The reason for our choice is that there exist several simple, well-studied controllers for routing in potential games. Thus, for any fixed value of the augmented variables and Lagrange multipliers, we can use any of these controllers to obtain convergence. Details of our game model and the payoffs used are discussed in Section V. Note that signalling is required to ensure feedback of node-prices and hyper-link rebates to the source nodes, but this overhead is small.

2) **Node Control:** In this phase, we adjust the augmented variables (hyper-link capacities) and Lagrange multipliers (node-prices) assuming that potential game of the sources has attained equilibrium.

$$\textbf{NC}: \quad \max_{\Sigma \geq 0} \min_{Y \geq 0} \mathbb{C}(X^*, Y, \Sigma),$$

where X^* is the assignment matrix at equilibrium. We use gradient decent controllers to modify the optimal hyper-link state and node-price. Details are discussed in Section VI.

We call our controller as *Decoupled Dynamics*. The two phases operate at different time scales. Traffic splitting is done at every *small* time scale and the node-control is done at every *large* time scale. Thus, sources attain equilibrium for given hyper-link capacities and prices, then the hyper-link capacities and prices are adjusted, and this in turn forces the sources to change their splits. This process continues until the source splits, hyper-link capacities and prices converge.

V. Traffic Splitting: Multi-path Network Coding (MPNC) Game

We model the traffic-splitting process of decoupled dynamics as a potential game with continuous action space, which we refer to as the *Multi-Path Network Coding Game* (MPNC Game). A potential game with continuous action space is defined by,

1) a set of players, $\mathcal{F}$,
2) an action space, $\mathbb{X} = \{X_i, \forall i \in \mathcal{F} | X_i \subset R^M, M \in \mathbb{N}\}$, where X_i is an action set of player i,
3) a set of continuously differentiable payoff functions of players, $\mathcal{C} = \{C_i : \mathbb{X} \to R, \forall i \in \mathcal{F}\}$,
4) a continuously differentiable potential function, $\Phi : \mathbb{X} \to R$, such that

$$\nabla_{a_i} \Phi(a_i, a_{-i}) = \nabla_{a_i} C_i(a_i, a_{-i}), \tag{18}$$

where $a_i \in X_i, a_{-i} \in \mathbb{X} \backslash X_i$.

Now, having defined the components of a potential game, we identify the corresponding entities in the case of MPNC game.

First of all, the flows are the players in the MPNC game. Then, the set of players is given by $\mathcal{F} = \{1, 2, \cdots, F\}$. The action set of player i (flow i) is defined as

$$X_i = \{\vec{x_i} = (x_i^1, x_i^2, \cdots, x_i^{P_i}) | \sum_j x_i^j = x_i\},$$

where x_i is the traffic demand of flow i and P_i is the number of hyper paths available to it. Note that each action $\vec{x}_i$ corresponds to, an instance of distribution of traffic demand seen by flow i, over the set of available hyperpaths. Then, the action space, $\mathbb{X}$, is given by $\mathbb{X} = \{X_1, \cdots, X_F\}$.

Finally, the payoff function of a player i is defined as

$$C_i(\vec{x}_i, \vec{x}_{-i}) = \mathbb{C}((\vec{x}_i, \vec{x}_{-i}), Y, \Sigma) - \mathbb{C}((\vec{0}, \vec{x}_{-i}), Y, \Sigma) \tag{19}$$

where $\mathbb{C}$ is the system cost function given by (17). In the above definition, $\vec{x}_i$ is the action of player i, $\vec{x}_{-i}$ is a set of actions of other players and $\vec{0}$ is a null vector. Also Y is the set of hyper link capacities and Σ is the set of node prices which remain invariant during each realization of MPNC game. The utility defined above is sometimes referred to as the *Wonderful life utility (WLU)* [19]. It is well known that payoff as in (19) results in a potential game with potential function $\Phi = \mathbb{C}$ [19].

In the context of MPNC game, it is clear that the payoff function, given by (19), is equal to the total transmission cost incurred by player i, while sending its own traffic over the set of available hyperpaths. Hence, in this game, the objective of each player is to *minimize* its own payoff.

But there is a caveat in using the system cost function $\mathbb{C}$ as the potential function and C_i's as the payoff functions. Recall from the conditions (3) and (4) of the definition of potential game that, the potential function and the utility functions must be differentiable. But, from (17) and (12) note that, the system cost function contains "min" terms over the hyper-link capacity and the flow rates, which makes the function non-differentiable. In order to have a continuously differentiable cost function we approximate these "min" terms using a generalized mean-valued function.

Let $a = \{a_1, \ldots, a_n\}$ be the set of positive real numbers and let t be some non-zero real number. Then the generalized *t-mean* of a is given by:

$$M_t(a) = \left(\frac{1}{n} \sum_{i=1}^{n} a_i^t \right)^{\frac{1}{t}} \tag{20}$$

The "min" function over the set a is approximated using $M_t(a)$ as:

$$\min\{a_1,\ldots,a_n\} = \lim_{t\to-\infty} M_t(a) \tag{21}$$

Substituting for M_t (20), instead of the "min" function in (17) we get the approximated total system function as:

$$\tilde{\mathbb{C}}(X,Y,\Sigma) = \sum_{f=1}^{F}\sum_{s=1}^{S_f}\beta_f^s x_f^s - \sum_{h=1}^{H}\tilde{T}(h) + \sum_{k=1}^{N}\sigma_k\tilde{V}_k, \tag{22}$$

where for a hyper-link $h = n_k[(i,q,n_i),(j,r,n_j)] \in \mathcal{H}$:

$$\begin{aligned}\tilde{T}(h) \;&= \alpha_{ki}\left(\tfrac{(x_i^q)^t+(y_h)^t}{2}\right)^{\frac{1}{t}} + \alpha_{kj}\left(\tfrac{(x_i^r)^t+(y_h)^t}{2}\right)^{\frac{1}{t}} \\ &\quad - \max\{\alpha_{ki},\alpha_{kj}\}y_h\end{aligned} \tag{23}$$

and

$$\tilde{V}_k = \sum_{f=1}^{F}\sum_{s=1}^{S_f}\sum_{m=1}^{N} R_{km}^{fp}\alpha_{ki}x_f^s - \sum_{h=1}^{H} Z_k^h\tilde{T}(h) - T_k. \tag{24}$$

The system function $\tilde{\mathbb{C}}(X,Y,\Sigma)$ is continuous and differentiable. So, we use the approximated function as our potential function. Similarly, the payoff of player i, given by (19), is approximated as follows:

$$\tilde{C}_i(\vec{x}_i,\vec{x}_{-i}) = \tilde{\mathbb{C}}((\vec{x}_i,\vec{x}_{-i}),Y,\Sigma) - \tilde{\mathbb{C}}((\vec{0},\vec{x}_{-i}),Y,\Sigma). \tag{25}$$

The marginal payoff obtained by flow $i \in \mathcal{F}$, given his action, $\vec{x}_i$, and the set of actions of other players, $\vec{x}_{-i}$, is

$$F_i(X,Y,\Sigma) = \nabla_{\vec{x}_i}\tilde{C}_i(X,Y,\Sigma) = \nabla_{\vec{x}_i}\tilde{\mathbb{C}}(X,Y,\Sigma), \tag{26}$$

where $X = (\vec{x}_i,\vec{x}_{-i})$. The above result follows from definition of potential function and (18). Note that F_i is a vector and let its p^{th} component be F_i^p. Then,

$$F_i^p(X,Y,\Sigma) \;=\; \frac{\partial\tilde{\mathbb{C}}(X,Y,\Sigma)}{\partial x_i^p},\ \forall i\in\mathcal{F},\ p\in\mathcal{P}_i \tag{27}$$

$$\begin{aligned} &= \beta_i^p - \sum_{h\in\mathcal{H}_i^p}\frac{\partial\tilde{T}(h)}{\partial x_i^p} + \sum_{k=1}^{N}\sum_{m=1}^{N} R_{km}^{ip}\sigma_k\alpha_{km} \\ &- \sum_{h\in\mathcal{H}_i^p}\sum_{k=1}^{N} Z_k^h\sigma_k\frac{\partial\tilde{T}(h)}{\partial x_i^p}.\end{aligned} \tag{28}$$

where, $\mathcal{H}_i^p$ the set of all hyper-links associated with flow f_i^p. From (23)

$$\frac{\partial\tilde{T}(h)}{\partial x_i^p} = \frac{1}{2}\alpha_{ki}\left(\frac{x_i^p}{M_t(x_i^p,y_h)}\right)^{t-1}, \tag{29}$$

and we have the min-approximation

$$M_t(x_i^p,y_h) = \left(\frac{(x_i^p)^t+(y_h)^t}{2}\right). \tag{30}$$

As we will show below, our algorithm will converge to the optimal state for any given value of $t<0$. Thus, we can attain a solution that is arbitrarily close to the original problem by choosing $|t|$ as large as desired. Also note that the payoff is the marginal cost incurred in using an option, so the players try to minimize their cost. The source node of each flow, $i\in\mathcal{F}$ observes the marginal cost, F_i^p, obtained in using a particular option (particular hyperpath), $p\in\mathcal{P}_i$, and changes the mass on that particular option, x_i^p, so as to attain equilibrium.

Next, we define the concept of equilibrium in potential games. A commonly used concept in non-cooperative games, is the Nash equilibrium. The game is said to be at Nash equilibium, if flows do not have any incentive to unilaterally deviate from their current action states. An action profile, $\hat{X} = (\hat{\vec{x}}_i,\hat{\vec{x}}_{-i}) \in \mathbb{X}$, results in a Nash equilibrium of MNPC game if

$$C_i(\hat{\vec{x}}_i,\hat{\vec{x}}_{-i}) \le C_i(\vec{x}_i,\hat{\vec{x}}_{-i}), \forall\vec{x}_i\in X_i, \forall i\in\mathcal{F}.$$

The above NE condition also implies that

$$F_i^p(\hat{X}) \le F_i^{p'}(\hat{X}) \ \ \forall p,p'\in\mathcal{P}_i, \forall i\in\mathcal{F},$$

where F_i^p is the marginal payoff given by (27). The above result can be interpreted as follows: At NE, for any player $i\in\mathcal{F}$, all the options (hyper paths) being used by that player, yield the same marginal payoff. Also, the marginal payoff that would have been obtained is higher for all those unused options.

The above concept refers to an *equilibrium condition*; the question arises as to how the system actually arrives at such a state. A commonly used kind of population dynamics is *Brown-von Neumann-Nash (BNN) Dynamics* [18]. The source nodes use BNN dynamics to control the mass on each option. But since each source tries to *minimize* its payoff, we use a modified version of BNN dynamics:

$$\dot{x}_i^p = \left(x_i\gamma_i^p - x_f^i\sum_{j=1}^{P_i}\gamma_i^j\right), \tag{31}$$

$$\text{where, } \gamma_i^p = \max\left\{\frac{1}{x_i}\sum_{j=1}^{P_i}F_i^j x_f^j - F_i^p, 0\right\}$$

where F_i^p is the marginal payoff of player i given by (27). In the next subsection, we prove the stability of our inner loop contorol.

A. *Convergence of MPNC Game*

We show in this susection that the multi-path network coding game converges to a stationary point when each source uses BNN dynamics. We will use the theory of Lyapunov functions [20] to show that our population game $\mathcal{G}$, is stable for a given hyper-link state $\breve{Y}$ and node-price state $\breve{\Sigma}$. We use the approximated system function (22) as our candidate Lyapunov function.

Theorem 1: The system of flows $\mathcal{F}$ that use *BNN dynamics* with payoffs given by (27) is globally asymptotically stable for a given hyper-link state $\breve{Y}$ and node-price state $\breve{\Sigma}$.

Proof: We use the approximated system function $\tilde{\mathbb{C}}(X,Y,\Sigma)$ (22) as our Lyapunov function. It is simple to verify that the cost function $\tilde{\mathbb{C}}(X,\breve{Y},\breve{\Sigma})$, is non-negative and convex, and hence is a valid candidate. For a given hyper-link state, $\breve{Y}$, and node-price state, $\breve{\Sigma}$, we define our Lyapunov function as:

$$\mathcal{L}_{\breve{Y}\breve{\Sigma}}(X) = \tilde{\mathbb{C}}(X, \breve{Y}, \breve{\Sigma}).$$

From (27)

$$\frac{\partial \mathcal{L}_{\breve{Y}\breve{\Sigma}}(X)}{\partial x_f^p} = \frac{\partial \tilde{\mathbb{C}}(X, \breve{Y}, \breve{\Sigma})}{\partial x_f^p} = F_f^p(X, \breve{Y}, \breve{\Sigma}).$$

Hence,

$$\begin{aligned}\dot{\mathcal{L}}_{\breve{Y}\breve{\Sigma}}(X) &= \textstyle\sum_{f=1}^{F} \sum_{p=1}^{S_f} \frac{\partial \mathcal{L}_{\breve{Y}\breve{\Sigma}}(X)}{\partial x_f^p} \dot{x}_f^p, \\ &= \textstyle\sum_{f=1}^{F} \sum_{p=1}^{S_f} F_f^p(X, \breve{Y}, \breve{\Sigma}) \dot{x}_f^p.\end{aligned}$$

From (31) we can substitute the value for $\dot{x}_f^p$ and we have

$$\dot{\mathcal{L}}_{\breve{Y}\breve{\Sigma}}(X) = \sum_{f=1}^{F} \sum_{p=1}^{S_f} F_f^p (x_f \gamma_f^p - x_f^p \sum_{j=1}^{S_f} \gamma_f^j),$$

$$= \sum_{f=1}^{F} x_f \left(\sum_{p=1}^{S_f} F_f^P \gamma_f^p - \left(\frac{1}{x_f} \sum_{p=1}^{S_f} F_f^p x_f^p \right) \sum_{j=1}^{S_f} \gamma_f^j \right).$$

We define

$$\begin{aligned}\bar{F}_f &\triangleq \frac{1}{x_f} \sum_{p=1}^{S_f} F_f^p x_f^p, \\ \implies & \sum_{f=1}^{F} x_f \left(\sum_{p=1}^{S_f} F_f^P \gamma_f^p - \sum_{j=1}^{S_f} \bar{F}_f \gamma_f^j \right), \\ &= \sum_{f=1}^{F} x_f \left(\sum_{p=1}^{S_f} \gamma_f^p (F_f^P - \bar{F}_f) \right), \\ &\leq -\sum_{f=1}^{F} x_f \left(\sum_{p=1}^{S_f} (\gamma_f^p)^2 \right) \leq 0.\end{aligned}$$

Thus,

$$\dot{\mathcal{L}}_{\breve{Y}\breve{\Sigma}}(X) \quad \leq \quad 0, \quad \forall\ X \in \mathcal{X}.$$

where equality exists when the state X corresponds to the stationary point of BNN dynamics. Hence, the system is globally asymptotically stable. ■

B. Efficiency

The objective of our system is to minimize the system function for a given load vector $\vec{x} = [x_1, \ldots, x_Q]$ and given hyper-link state $\breve{Y}$ and node-price state $\breve{\Sigma}$. Here the system function $\tilde{\mathbb{C}}(X, \breve{Y}, \breve{\Sigma})$ and is defined in (22). This can be represented as the following constrained minimization problem:

$$\min_X \tilde{\mathbb{C}}(X, \breve{Y}, \breve{\Sigma}) \tag{32}$$

subject to:

$$\sum_{p=1}^{S_i} x_i^p \quad = \quad x_i \quad \forall\ i \in \mathcal{F} \tag{33}$$

$$x_i^p \quad \geq \quad 0.$$

The Lagrange dual associated with the above minimization problem, for a given $\breve{Y}$ and $\breve{\Sigma}$ is

$$\mathcal{L}_{\breve{Y}\breve{\Sigma}}(\lambda, h, X) = \max_{\lambda, h} \min_X \Bigg(\tilde{\mathbb{C}}(X, \breve{Y}, \breve{\Sigma}) \quad - \sum_{i=1}^{F} \lambda_i \Big(\sum_{p=1}^{S_i} x_i^p - x_i \Big) \quad - \quad \sum_{i=1}^{F} \sum_{p=1}^{S_i} h_i^p x_i^p \Bigg) \tag{34}$$

where λ_i and $h_p^i \geq 0$, $\forall\ i \in \mathcal{F}$ and $p \in \mathcal{S}_i$, are the dual variables. Now the above dual problem gives the following Karush-Kuhn-Tucker first order conditions:

$$\frac{\partial \mathcal{L}_{\breve{Y}\breve{\Sigma}}}{\partial x_i^p}(\lambda, h, X^\star) = 0 \quad \forall\ i \in \mathcal{F} \text{ and } p \in \mathcal{S}_i \tag{35}$$

and

$$h_i^p x_i^{\star p} = 0 \quad \forall\ i \in \mathcal{F} \text{ and } p \in \mathcal{S}_i \tag{36}$$

where $X^\star$ is the global minimum for the primal problem (32). Hence from (35) we have, $\forall\ i \in \mathcal{F}$ and $\forall\ p \in \mathcal{S}_i$,

$$\frac{\partial \tilde{\mathbb{C}}}{\partial x_i^p}(X^\star, \breve{Y}, \breve{\Sigma}) - \lambda_i \frac{\partial (\sum_{p=1}^{S_i} x_i^{\star p} - x^\star i)}{\partial x_i^p} + h_i^p \;=\; 0$$

$$\Rightarrow \frac{\partial \tilde{\mathbb{C}}}{\partial x_i^p}(X^\star, \breve{Y}, \breve{\Sigma}) \;=\; \lambda_i + h_i^p \tag{37}$$

$$\Rightarrow F_i^p(X^\star, \breve{Y}, \breve{\Sigma}) \;=\; \lambda_i + h_i^p \tag{38}$$

where the last equation follows from (26).
From (36), it follows that

$$F_i^p(X^\star, \breve{Y}, \breve{\Sigma}) \;=\; \lambda_i \quad \text{when } x_i^{\star p} > 0 \tag{39}$$

and

$$F_i^p(X^\star, \breve{Y}, \breve{\Sigma}) \;=\; \lambda_i + h_i^p \quad \text{when } x_i^{\star p} = 0 \tag{40}$$

$\forall\ i \in \mathcal{F}$ and $\forall\ p \in \mathcal{S}_i$. The above condition (39, 40), implies that the payoff on all the options used is identical and for options not in use the payoff is more, which is equivalent to the NE condition given by (31). Notice that we use a modified definition of Nash equilibrium, since each source tries to minimize it's cost (or payoff). The following theorem proves the efficiency of our system.

Theorem 2: The solution of the minimization problem in (32) is identical to the Nash equilibrium of MPNC game.

Proof: Consider the BNN dynamics (31), at stationary point, $\tilde{X}$, we have $\dot{x}_i^p = 0$, which implies that either,

$$\hat{F}_i = F_i^p(\tilde{X}, \breve{Y}, \breve{\Sigma}) \tag{41}$$

or

$$\hat{x}_i^p = 0,$$

where,

$$\hat{F}_i \triangleq \tfrac{1}{\hat{x}_i} \textstyle\sum_{r=1}^{Q} \hat{x}_i^r F_i^r(\tilde{X}, \breve{Y}, \breve{\Sigma}) \quad \forall\ i \in \mathcal{F}, \tag{42}$$

The above expressions imply that all hyper-paths used by a particular flow $i \in \mathcal{F}$ yield same payoff, $\hat{F}_i$, while hyper-paths not used ($x_i^p = 0$) yield a payoff higher than $\hat{F}_i$.

We observe that the conditions required for Nash equilibrium are identical to the KKT first order conditions (39)-(40) of the minimization problem (32) when

$$\hat{F}_i = \lambda_i \quad \forall\ i \in \mathcal{F}$$

It follows from the convexity of the total system cost that, there is no duality-gap between the primal (32) and the dual (34) problems. Thus, the optimal primal solution is equal to optimal dual solutions, which is identical to the Nash equilibrium. ■

VI. Node Control

Thus far we have designed a distributed scheme that would result in minimum cost for a given hyper-link state or capacities Y, node-price state Σ and for a given load vector $\vec{x} = \{x_1, \ldots, x_f\}$. In this phase of Decoupled Dynamics, the hyper-link capacities and node-prices are adjusted based on the current value of system function. This phase runs at a larger time-scale as compared to the traffic splitting phase described in Section V. It is assumed that during this phase all the flows instantly reach equilibrium, i.e., changing the hyper-link capacities and node-prices would force all the source nodes to attain Wardrop equilibrium instantaneously.

The node control can be formulated as a convex optimization problem as follows:

$$\max_{\Sigma} \min_{Y} \quad Q(Y, \Sigma), \tag{43}$$
$$\text{subject to,} \quad y_h, \sigma_k \geq 0, \ \forall y_h \in Y \text{ and } \forall \sigma_k \in \Sigma.$$

where, $Q(Y, \Sigma)$ is the minimum value of the system function for a given hyper-link state Y and node-price state Σ, i.e., $Q(Y, \Sigma) = \tilde{\mathbb{C}}(X^\star, Y, \Sigma)$, where, for a given Y and Σ, $X^\star$ is an optimal state of the flows that results in minimum cost.[2] We use simple gradient descent:

$$\dot{y}_h = -\kappa \frac{\partial Q(Y, \Sigma)}{\partial y_h} \, \forall y_h \in Y, \tag{44}$$

$$\dot{\sigma}_k = \rho \frac{\partial Q(Y, \Sigma)}{\partial \sigma_k} \, \forall \sigma_k \in \Sigma. \tag{45}$$

The partial derivative, $\frac{\partial Q}{\partial y_h}$, is over the variables $y_h \in Y$. Keeping Σ fixed and changing the hyper-link capacity y_h, of some hyper-link $h \in \mathcal{H}$, would result in a different state of the flows, $X_h^\star$ and hence a different minimum cost, $\tilde{\mathbb{C}}(X_h^\star, Y_h, \Sigma)$, where Y_h corresponds to the changed hyper-link capacity of y_h while other capacities are fixed, as compared to Y. Thus for a hyper-link, $h = n_k[(i, q, n_i), (j, t, n_j)]$ with capacity y_h,

$$\begin{aligned} \frac{\partial Q(Y, \Sigma)}{\partial y_h} &= \tfrac{\partial \tilde{\mathbb{C}}}{\partial y_h}(X^\star, Y, \Sigma) \\ &\quad + \textstyle\sum_{i=1}^{F} \sum_{p=1}^{P_i} \frac{\partial \tilde{\mathbb{C}}}{\partial x_i^p}(X^\star, Y, \Sigma) \frac{\partial x_i^{*p}}{\partial y_h} \\ &= \tfrac{\partial \tilde{\mathbb{C}}}{\partial y_h}(X^\star, Y, \Sigma) + \textstyle\sum_{i=1}^{F} F_i \sum_{p=1}^{S_i} \frac{\partial x_i^{*p}}{\partial y_h}, \end{aligned} \tag{46}$$

where the last expression follows from the definition of F_i^p (Definition 27) and the fact that for changes in the hyper-link state, the sources attain Wardrop equilibrium instantaneously. In other words, before and after a small change in y_h the system is in Wardrop equilibrium. Hence, $F_i^p = F_i \; \forall i \in \mathcal{F}$ and $\forall p \in \mathcal{S}_i$. Finally, $\sum_{p=1}^{S_i} \frac{\partial x_i^{*p}}{\partial y_h} = 0$, since the total load $x_i^* = \sum_{p=1}^{S_i} x_i^{*p}$ is fixed. For hyper-link $h = n_k[(i, q, n_i), (j, t, n_j)]$,

$$\frac{\partial Q(Y, \Sigma)}{\partial y_h} = \tfrac{\partial \tilde{\mathbb{C}}}{\partial y_h}(X^\star, Y, \Sigma) = -(1 + \sigma_k) \tfrac{\partial \tilde{T}}{\partial y_h}(h), \tag{47}$$

where from (23),

$$\begin{aligned} \frac{\partial \tilde{T}}{\partial y_h}(h) &= \tfrac{\alpha_{ki}}{4} \left(\tfrac{y_h}{M_t(x_i^q, y_h)} \right)^{t-1} + \tfrac{\alpha_{kj}}{4} \left(\tfrac{y_h}{M_t(x_j^r, y_h)} \right)^{t-1} \\ &\quad - \max\{\alpha_{ki}, \alpha_{kj}\}, \end{aligned}$$

$$\text{and} \qquad M_t(x_i^q, y_h) = \left(\tfrac{(x_i^q)^t + (y_h)^t}{2} \right)^{\frac{1}{t}}.$$

Similarly, we can show that

$$\frac{\partial Q(Y, \Sigma)}{\partial \sigma_k} = \tfrac{\partial \tilde{\mathbb{C}}}{\partial \sigma_k}(X^\star, Y, \Sigma) = \tfrac{\partial \tilde{V}_k}{\partial \sigma_k}, \tag{48}$$

where, from (24)

$$\frac{\partial \tilde{V}_k}{\partial \sigma_k} = \left(\sum_{m=1}^{N} \sum_{f=1}^{F} \sum_{p=1}^{S_f} R_{km}^{fp} \alpha_{ki} \hat{x}_f^s - \sum_{h=1}^{H} Z_k^h \tilde{T}(h) - T_k \right).$$

Theorem 3: At the large time-scale, the hyper-link capacity control with dynamics (44) and node price control with dynamics (45) is globally asymptotically stable.

Proof: We use the following Lyapunov function

$$G(Y, \Sigma) = \frac{1}{2\kappa} \sum_{h=1}^{H} (y_h - \hat{y}_h)^2 + \frac{1}{2\rho} \sum_{k=1}^{N} (\sigma_k - \hat{\sigma}_k)^2 \tag{49}$$

where $\hat{y}_h \in \hat{Y}$ and $\hat{\sigma}_k \in \hat{\Sigma}$ are optimizers of (43). We will use LaSalle's invariance principle [20] to show stability.

Differentiating G we obtain

$$\dot{G} = \frac{1}{\kappa} \sum_{h=1}^{H} (y_h - \hat{y}_h) \dot{y}_h + \frac{1}{\rho} \sum_{k=1}^{N} (\sigma_k - \hat{\sigma}_k) \dot{\sigma}_k.$$

Now from (44) and (45),

$$\dot{G} = -\sum_{h=1}^{H} (y_h - \hat{y}_h) \frac{\partial Q}{\partial y_h} + \sum_{k=1}^{N} (\sigma_k - \hat{\sigma}_k) \frac{\partial Q}{\partial \sigma_k}. \tag{50}$$

We will show that $\dot{G} \leq 0, \forall Y, \forall \Sigma$.

Note that $Q(Y, \Sigma) = \tilde{\mathbb{C}}(X^*, Y, \Sigma)$, where X^* is a minimizer of approximated cost function defined in (22) for fixed Y and Σ. Also, for any fixed node-price state Σ, the approximated cost function is jointly convex in X and Y. Therefore, minimizing it over a convex set of X yields a convex function. In essence, $Q(Y, \Sigma)$ is convex in Y for any fixed Σ. It can be observed that for any fixed hyper-link state Y and rate vector X, the approximated cost function defined in (22) is a linear function of Σ. Then the minimization of $\tilde{\mathbb{C}}(X, Y, \Sigma)$ over X can be thought of as a point-wise minimization of infinite number of linear functions of Σ which results in a concave function of Σ. Therefore, $Q(Y, \Sigma)$ is concave in Σ for any fixed Y. Therefore, from the convex-concave nature of $Q(Y, \Sigma)$ we can show that

$$Q(\hat{Y}, \Sigma) \leq Q(\hat{Y}, \hat{\Sigma}) \leq Q(Y, \hat{\Sigma}), \forall Y, \forall \Sigma. \tag{51}$$

[2] Notice, there could be many different states, $X^\star$, which result in a minimum cost but the minimum value, $\tilde{\mathbb{C}}(X^\star, Y, \Sigma)$, is unique.

where $\hat{Y}$ and $\hat{\Sigma}$ are optimizers of the problem (43). Now, using the first order properties of convex and concave functions,

$$Q(\hat{Y},\Sigma) \geq Q(Y,\Sigma) + \sum_{h=1}^{H}(\hat{y}_h - y_h)\frac{\partial Q}{\partial y_h}, \quad (52)$$

$$Q(Y,\hat{\Sigma}) \leq Q(Y,\Sigma) + \sum_{k=1}^{N}(\hat{\sigma}_k - \sigma_k)\frac{\partial Q}{\partial \sigma_k}. \quad (53)$$

From equations (50-53), we can write

$$\dot{G} = -\sum_{h=1}^{H}(y_h - \hat{y}_h)\frac{\partial Q}{\partial y_h} + \sum_{k=1}^{N}(\sigma_k - \hat{\sigma}_k)\frac{\partial Q}{\partial \sigma_k} \leq 0$$

In order to apply La Salle's invariance priniciple, let us consider a set of points $\mathcal{E}$ for which the condition $\dot{G} = 0$ is satisfied. The largest invariant set $\mathcal{M}$ is a subset of points such that $\frac{\partial Q}{\partial y_h} = 0, \forall y_h \in Y$ and $\frac{\partial Q}{\partial \sigma_k} = 0, \forall \sigma_k \in \Sigma$. Pick any point $(\tilde{Y},\tilde{\Sigma}) \in \mathcal{M}$. We can show from the properties convex-concave nature of function $Q(Y,\Sigma)$ that $Q(\tilde{Y},\tilde{\Sigma}) \leq Q(Y,\tilde{\Sigma}), \forall Y$ and $Q(\tilde{Y},\tilde{\Sigma}) \geq Q(\tilde{Y},\Sigma), \forall \Sigma$. Therefore, the pair $(\tilde{Y},\tilde{\Sigma})$ satisfies the condition (51) and it is an optimizer of (43). From La Salle's principle, the dynamics converge to the largest invariant set $\mathcal{M}$ and therefore the convergent point is an optimal state of (43). Hence the system is globally asymptotically stable [20].

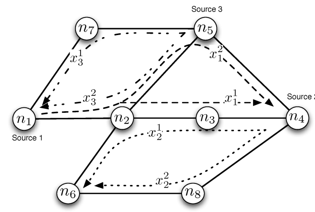
■

VII. Simulations

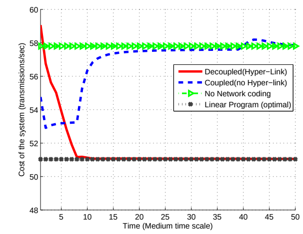
We simulated our system in Matlab to show system convergence. We first performed our simulations for our simple network shown in Figure 3(a). The load at the source nodes 1, 2 and 3 is given as 4.73, 2.69 and 3.56 respectively, which are randomly generated values. We use the following costs on the individual links (α_{ij}): $\alpha_{12} = 2.8$, $\alpha_{23} = 1.6$, $\alpha_{34} = 1.8$, $\alpha_{25} = 1.3$, $\alpha_{54} = 2.1$, $\alpha_{26} = 1.7$, $\alpha_{48} = 2.9$, $\alpha_{86} = 2.2$, $\alpha_{57} = 1.9$, $\alpha_{71} = 2.6$; we assume the costs on the links are symmetric. We use the approximated cost function (22), with a value of $t = -30$ for the approximation parameter (21) for our simulations. We have assumed that the maximum number of transmissions (per unit time) from each node is limited to 15. The simulation is run for 50 large time units, and in each large time scale we have 20 small time units.

We compare the total cost of the system for the following:

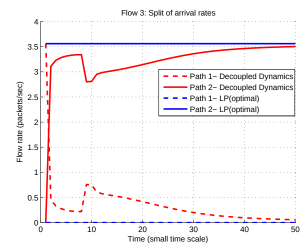
1) Decoupled Dynamics (DD): This is the algorithm that we developed under the augmented potential game framework; we use our hyper-links to decouple the flows that participate in coding.
2) Coupled Dynamics (no hyper-link) (CD): Here, there is coupling between individual flows and coding happens at the minimum rate of the constituent flows. In other words, this is the original potential game without augmentation. We use similar game dynamics as that was used in DD. The total cost is specified in Equation (5).
3) No Coding: In this system no network coding is used. This gives an baseline with respect to which the gains attained by coding can be quantified.



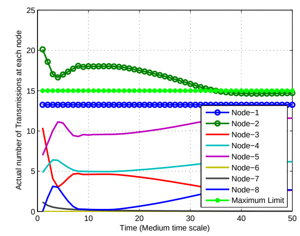
(a) Network topology



(b) Total system cost (per unit rate)



(c) Traffic (packets/sec) from flow-3 using DD converges to optimum



(d) Transmissions (per unit time) of each node converges to below maximum

Fig. 3. Performance evaluation of simple network topology

TABLE I
Comparison of state variables for LP, DD and CD

Variable	x_1^1	x_1^2	x_2^1	x_2^2	x_3^1	x_3^2	y_2	y_3
LP	1.52	3.2	1.52	1.16	0.00	3.56	3.20	1.52
DD	1.6	3.12	1.71	0.97	0.09	3.46	3.29	1.58
CD	4.70	0.00	0.01	2.68	0.62	2.93	N/A	N/A

4) LP Optimal (LP): This is a centralized solution. We formulated our system as a Linear Program (LP) of minimizing cost (17) over X and Y for a given load vector that we obtain using an LP-solver.

As seen in the Figure 3(b), the total cost of the system (number of transmissions per unit time) for our model (decoupled using hyper-link) is close to the optimal solution obtained by solving it in a centralized fashion. We compared the final system state of DD and CD with that of the solution obtained using LP. We observe from Table I that the values for the split (X) and the hyper-link capacities (Y) generated by DD are near-optimal (LP results), but CD is very different. We have plotted time evolution of traffic splits of flow 3, over options 1 and 2, in the Figure 3(c), which shows that they converge to the optimal values obtained by LP solver. In Figure 3(d), we have shown that the number of transmissions from all the nodes is less than or equal to the maximum threshold.

Next, we perform our simulations on a bigger topology shown in Figure 4. This network consists of 30 nodes shared by 6 flows. Flows 1, 2, 3 and 6 have two hyper-paths each and flows 4 and 5 have three hyper-paths each. There are 6 hyper-links in the system. Table II describes the source, destination nodes and the hyper-paths for each flows. Notice, options 2 and 3 of flow 4 have the same physical path but different hyper-links, y_1 and y_2 at node n_7. This is because the sub-flow of x_4 traversing the physical path $(16, 15, 11, 6, 7, 8)$ can be encoded with two different flows, x_1^2 and x_2^2 traversing in the reverse direction at node 7.

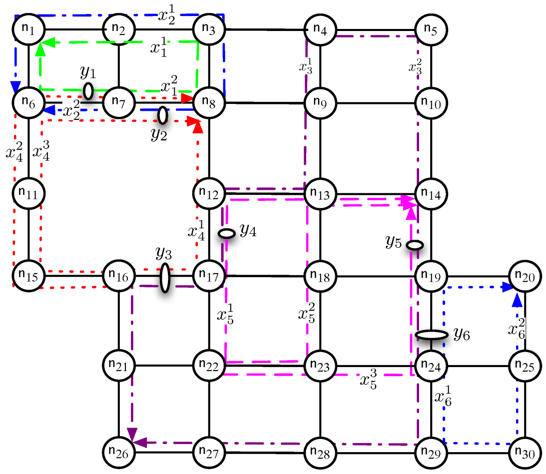


Fig. 4. Complex network

TABLE II
SOURCE, DESTINATION NODES AND HYPER-PATHS CORRESPONDING TO EACH FLOW.

Id	Src Node	Dest. Node	Hyper-Paths
1	8	1	(8,3,2,1) & (8,7,6,1)
2	8	6	(8,3,2,1,6) & (8,7,6)
3	5	26	(5,4,9,13,12,17,16,21,26) & (5,10,14,19,24,29,28,27,26))
4	16	8	(16,17,12,8), (16,15,11,6,7,8) & (16,15,11,6,7,8)
5	23	14	(23,22,17,12,13,14), (23,18,13,14) & (23,24,19,14)
6	29	20	(29,24,19,20) & (29,30,25,20)

We ran our algorithms on this network with random link costs for four example cases. We first present detailed results for one of the example cases. The simulation is run for 150 large time units, and in each large time scale we have 50 small time units. As seen in Figure 5, the total system cost for decoupled dynamics converges to the optimal solution which is obtained by solving the problem in a centralized fashion.

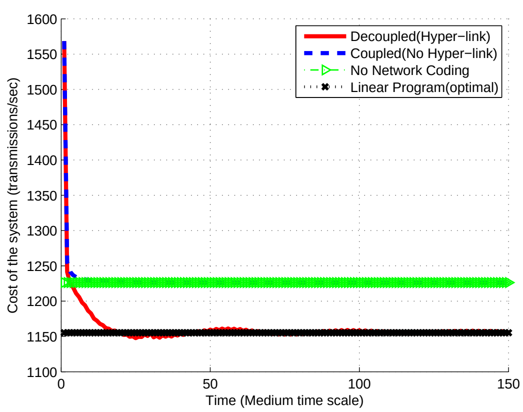


Fig. 5. Comparison of total system cost (per unit rate), for different systems: DD and non-coded against LP.

VIII. CONCLUSION

We considered a wireless network with given costs on arcs, traffic matrix and multiple paths. The objective was to find the splits of traffic for each source across its multiple paths in a distributed manner leveraging the reverse carpooling technique where the peak transmissions (per unit time) at each node is limited. For this we split the problem into two sub-problems, and propose a two-level distributed control scheme set up as a game between the sources and the hyperlink nodes. On one level, given a set of hyperlink capacities and node-prices, the sources selfishly choose their splits and attain a Nash equilibrium. On the other level, given the traffic splits, the hyperlinks and nodes may slightly increase or decrease their capacities and prices using a steepest descent algorithm. We constructed a Lyapunov function argument to show that this process asymptotically converges to the minimum cost solution, although performed in a distributed fashion.

In designing the two level controller, we came up with an interesting formulation that we believe might be useful in other coordination games. The idea is to augment the state space of the system using additional variables that are controlled by unselfish agents. Although these agents only have local information at their disposal, they are able to modify the potential function of the system as a whole, and hence change the actions taken by the selfish routing agents. Essentially, these agents take on some of the system cost on themselves in order to redistribute the overall costs. The system wide cost is minimized as a result. We also showed that the idea can be coupled with a Lagrange multiplier approach to enforce constraints as well.

We performed several numerical studies and found that our two-level controller converges fast to the optimal solutions. Some of the bi-products of our experiments were that: more expensive paths before network coding became cheaper and shortest paths were not necessarily optimal. In conclusion, from a methodological standpoint we have a distributed controller that achieves a near-optimal solution when the individuals are self-interested.

TABLE III
COMPARISON OF STATE VARIABLES FOR NO CODING, LP, DD AND CD.

	No Coding	LP	DD	CD
x_1^1	19.10	19.10	19.09	19.09
x_1^2	0	0	0.01	0.01
x_2^1	0	0	0.01	0.04
x_2^2	21.08	21.07	21.07	21.07
x_3^1	15.32	12.42	12.99	15.32
x_3^2	0	2.90	2.33	0
x_4^1	14.97	15.10	15.02	15.08
x_4^2	0.06	0	0.0087	0.0087
x_4^3	0	0	0	0
x_5^1	0	8.69	8.8	0.05
x_5^2	0	0	0.05	9.19
x_5^3	11.6	2.90	2.79	11.54
x_6^1	18.43	18.43	18.43	18.43
x_6^2	0	0	0	0
y_1	N/A	0	0	N/A
y_2	N/A	0.17	0.63	N/A
y_3	N/A	12.47	13.87	N/A
y_4	N/A	8.69	9.15	N/A
y_5	N/A	2.9	2.68	N/A
y_6	N/A	2.9	3.98	N/A

References

[1] S. Katti, H. Rahul, D. Katabi, W. H. M. Médard, and J. Crowcroft, "XORs in the Air: Practical Wireless Network Coding," in *ACM SIGCOMM*, Pisa, Italy, 2006.

[2] M. Effros, T. Ho, and S. Kim, "A tiling approach to network code design for wireless networks," *Information Theory Workshop, 2006. ITW '06 Punta del Este. IEEE*, pp. 62–66, March 2006.

[3] W. H. Sandholm, "Potential Games with Continuous Player Sets," *Journal of Economic Theory*, vol. 97, pp. 81–108, January 2001.

[4] M. Benaim and W. Sandholm, "Logit evolution in potential games: Reversibility, rates of convergence, large deviations, and equilibrium selection," *Unpublished manuscript, Université de Neuchâtel and University of Wisconsin*, 2007.

[5] J. Marden, G. Arslan, and J. Shamma, "Joint strategy fictitious play with inertia for potential games," *IEEE Transactions on Automatic Control*, vol. 54, no. 2, pp. 208–220, Feb. 2009.

[6] O. Candogan, A. Ozdaglar, and P. Parrilo, "A projection framework for near-potential games," in *Proc. of the 49th IEEE Conference on Decision and Control (CDC 10)*, December 2010, pp. 244–249.

[7] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network Information Flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[8] Y. Sagduyu and A. Ephremides, "Cross-layer optimization of MAC and network coding in wireless queueing tandem networks," *Information Theory, IEEE Transactions on*, vol. 54, no. 2, pp. 554–571, Feb. 2008.

[9] U. K. Hulya Seferogluy, Athina Markopoulouy, "Network coding-aware rate control and scheduling in wireless networks," in *Proc. IEEE International Conference on Multimedia and Expo*, Jun 2009.

[10] Y. Feng, Z. Liu, and B. Li, "Gestureflow: streaming gestures to an audience," in *IEEE INFOCOM 2011*, 2011, pp. 748–756.

[11] S. Sengupta, S. Rayanchu, and S. Banerjee, "An analysis of wireless network coding for unicast sessions: The case for coding-aware routing," in *IEEE INFOCOM 2007*, 2007, pp. 1028–1036.

[12] S. Das, Y. Wu, R. Chandra, and Y. Hu, "Context-based routing: techniques, applications and experience," in *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation table of contents*. USENIX Association Berkeley, CA, USA, 2008, pp. 379–392.

[13] J. Marden and M. Effros, "The Price of Selfishness in Network Coding," in *Workshop on Network Coding, Theory, and Applications*, 2009.

[14] V. Reddy, S. Shakkottai, A. Sprintson, and N. Gautam, "Multipath wireless network coding: A population game perspective," in *IEEE INFOCOM*, San Diego, CA, March 2010.

[15] J. Marden and A. Wierman, "Overcoming limitations of game-theoretic distributed control," in *Proc. of the 48th IEEE Conference on Decision and Control (CDC 09)*, December 2009, pp. 6466–6471.

[16] N. Li and J. Marden, "Designing games to handle coupled constraints," in *Proc. of the 49th IEEE Conference on Decision and Control (CDC 10)*, December 2010, pp. 250–255.

[17] A. ParandehGheibi, A. Ozdaglar, M. Effros, and M. Medard, "Optimal reverse carpooling over wireless networks - a distributed optimization approach," in *CISS*, Princeton, NJ, March 2010.

[18] G. W. Brown and J. von Neumann, "Solution of games by differential equations," *Contributions to the Theory of Games I, Annals of Mathematical Studies*, vol. 24, 1950.

[19] R. Gopalakrishnan, J. Marden, and A. Wierman, "An architectural view of game theoretic control," in *Proc. of ACM Hotmetrics*, June 2010.

[20] H. Khalil, *Nonlinear Systems*. Prentice Hall, 1996.

Vinod Ramaswamy received his Bachelor of Engineering degree in electronics and communication engineering from Kerala University, India, in 2004 and his M.Tech. degree in electrical engineering, with specialization in communication systems, from Indian Institute of Technology, Madras, India, in 2006. He is currently working towards the Ph.D. degree at Texas A&M University, College Station.

His research interests include game theory, optimization methods, and their application to communication networks.

Vinith Reddy Podduturi received dual degrees in Master of Science in Mathematics and Bachelor of Engineering in Computer science, from Birla Institute of Technology and Science, pilani, India in 2006. He received his M.S. degree in Computer Engineering from Texas A&M university in 2009. He is currently working in Microsoft Corp. at Seattle, WA. This work was done as part of his Masters thesis.

His research interests include peer-to-peer systems, pricing approaches to network resource allocation, game theory and network coding.

Srinivas Shakkottai (S '00-M '08) received his Bachelor of Engineering degree in electronics and communication engineering from the Bangalore University, India, in 2001 and his M.S. and Ph.D degrees from the University of Illinois at Urbana-Champaign in 2003 and 2007, respectively, both in electrical engineering. He was Postdoctoral Scholar at Stanford University until December 2007, and is currently an Assistant Professor at the Dept. of ECE, Texas A&M University. He has received the Defense Threat reduction Agency Young Investigator Award (2009) and the NSF Career Award (2012) as well as research awards from Cisco (2008) and Google (2010).

His research interests include content distribution systetems, pricing approaches to network resource allocation, game theory, congestion control, and the measurement and analysis of Internet data.

Dr. Alex Sprintson is an Associate Professor with the Department of Electrical and Computer Engineering, Texas A&M University, College Station. From 2003 to 2005, he was a Postdoctoral Research Fellow with the California Institute of Technology, Pasadena. His research interests lie in the general area of communication networks with a focus on network coding, network survivability and robustness, and support for Quality of Service (QoS).

He received the Wolf Award for Distinguished Ph.D.students, the Viterbi Postdoctoral Fellowship, and the NSF CAREER award. In 2006-2012 he was an Associate Editor of the IEEE Communications Letters and Computer Networks Journal. He has been a member of the Technical Program Committee for the IEEE Infocom 2006-2013.

Natarajan Gautham is an Associate Professor in the Department of Industrial and Systems Engineering at Texas A&M University with a courtesy appointment in the Department of Electrical and Computer Engineering. Prior to joining Texas A&M University in 2005, he was on the Industrial Engineering faculty at Penn State University for eight years. He received his M.S. and Ph.D. in Operations Research from the University of North Carolina at Chapel Hill, and his B.Tech. from Indian Institute of Technology, Madras.

His research interests are in the areas of modeling, analysis and performance evaluation of stochastic systems with special emphasis on optimization and control in computer, telecommunication and information systems. He is an Associate Editor for the INFORMS Journal on Computing, IIE Transactions, and OMEGA.