Dice: a Game Theoretic Framework for Wireless Multipath Network Coding

Xinyu Zhang, Baochun Li*
Department of Electrical and Computer Engineering
University of Toronto
{xzhang,bli}@eecg.toronto.edu

ABSTRACT

Network coding has emerged as a promising approach that enables reliable and efficient end-to-end transmissions in lossy wireless mesh networks. Existing protocols have demonstrated its resilience to packet losses, as well as the ability to integrate naturally with multipath opportunistic routing. However, these heuristics do not take into account the inherent resource competition in wireless networks, thereby compromising the coding advantages. In this paper, we take a game-theoretic perspective towards optimized resource allocation for network coding based unicast protocols. We design decentralized mechanisms that achieve better efficiency-fairness tradeoff, for both cooperative and selfish users. Our framework features a modularized optimization of two subproblems: the multipath routing of coded information flows for each player, and the broadcast and coding rate allocation among competing players. We have implemented the framework on a wireless emulation testbed and demonstrated its high performance in terms of throughput and fairness.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design; C.2.2 [**Computer-Communication Networks**]: Network Protocols

General Terms

Algorithms, Theory

Keywords

Network coding, multipath opportunistic routing, wireless mesh networks, flow control, optimization decomposition, game theory

*This work was supported in part by LG Electronics, and by Bell Canada through its Bell University Laboratories R&D program.

MobiHoc'08, May 26–30, 2008, Hong Kong SAR, China.

1. INTRODUCTION

Existing measurement studies have revealed the prevalence of low-quality wireless links in real-world wireless networks [1]. This inspired routing protocols that are sustainable under lossy conditions. Such protocols have mostly revolved around two major issues: reliability and efficiency (in terms of energy or throughput). An intuitive way towards efficiency is to follow the traditional shortest-path paradigm, with a path metric associated with the reception probabilities of wireless links [2]. Traditional multipath routing has been proposed for the purpose of reliability, but not for throughput or energy efficiency in lossy wireless networks, mainly due to the redundancy and route coupling problem [3].

An opportunistic routing protocol (*e.g.*, ExOR [4]) takes advantage of the broadcast nature of wireless networks, exploring the forwarding capacity of all intermediate nodes that may overhear packets. It avoids overlapping packet transmissions through negotiations among intermediate forwarders. Though claiming high throughput, the protocol requires complex interactions between intermediate forwarders before each scheduling and forwarding decision, which is infeasible in lossy networks. Moreover, ExOR does not ensure end-to-end reliability, and relies on traditional routing to make up for the missing packets.

Network coding departs from the conventional store-and-forward paradigm by allowing mixture of information at the source and intermediate relays. Specifically, the source node splits the original data file into data blocks and then encode them with random linear codes (RLC) [5]. Intermediate nodes can re-encode and forward the linearly independent blocks on hand. The destination is able to decode once it receives a sufficient number of coded blocks. The upshot of such a network code is that full reliability can be achieved without retransmissions — a single ACK can ensure the successful reception of a group of data blocks. Furthermore, the route coupling problem no longer exists since paths are implicitly formed — all nodes that are closer to the destination than their predecessors take part in the forwarding. Following the preliminary proposal in [6], Chachulski *et al.* [5] implemented such a scheme in a wireless mesh network testbed, focusing on the practical problems therein. The core of the protocol in [5] (referred to as MORE) lies in an iterative centralized algorithm that determines how many outgoing packets a relay should generate upon receiving one fresh packet. However, this heuristic does not account for the bandwidth resource competition among neighboring nodes, which results from the inherent interference effect of wireless networks. Consequently, the network tends to become congested, especially when multiple concurrent sessions coexist.

In this paper, we address the problem of resource allocation for multipath network coding (henceforth referred to as MNC) in lossy wireless mesh networks. In particular, we develop *Dice*[1], a game theoretic framework that models the coding, routing, and bandwidth constraints specific to an MNC protocol. *Dice* leads all players (represented by source-destination pairs) in the network towards optimized opportunistic multipath routing and rate allocation. Within this framework, we consider both the case where the players are cooperative, willing to share the resources through negotiation; and the case where players are selfish, greedily maximizing their own utility. In the cooperative framework, social optimum is achieved through a decentralized bargaining algorithm. We design the bargaining algorithm based on optimization decomposition, and prove its optimality as well as convergence rate. In the noncooperative framework, socially efficient Nash equilibrium can still be ensured, assuming wireless nodes in the network enforce a pricing mechanism on all the players. A similar line of analysis can be applied to the case where energy rather than bandwidth is the scarce resource, as in mobile ad hoc networks.

In order to validate the performance of the *Dice* framework, we have implemented *Dice* on a wireless emulation testbed that is designed for computationally extensive application specific protocols like MNC. We observe that the resource allocation mechanism in *Dice* results in a much higher level of throughput and fairness, in comparison with the heuristic in MORE. To our knowledge, there has been no existing work in the literature with a focus on using a *game theoretic* framework to resolve conflicts of interest between competing flows when wireless multipath network coding is used.

The remainder of the paper is structured as follows. In Sec. 2, we present a literature review of related work. Sec. 3 formulates the players' strategy space within which they are running the MNC protocol. Following this, Sec. 4 introduces the cooperative model in *Dice*, derives a decentralized negotiation algorithm for each user, and proves the convergence of the algorithm. Sec. 5 develops the noncooperative model in *Dice*, as well as the corresponding algorithm that leads to an efficient Nash equilibrium. In Sec. 6, we present experimental evaluations in order to verify the *Dice* framework. Finally, Sec. 7 concludes the paper.

2. RELATED WORK

The problem of efficient and fair resource allocation has been extensively explored in the context of wireless ad hoc and mesh networks. This is closely related with the vast literature of cross layer utility optimization [7], where utility functions represent different notion of fairness-efficiency tradeoff. Recently, game theory has been applied to analyze the cooperative and selfish behavior of wireless network users. For instance, [8] applied the Nash bargaining solution approach to model the cooperative relay and bandwidth allocation in wireless multihop networks. Efficiency and fairness are enforced by pricing mechanisms. [9] addressed the problem of MAC layer bandwidth allocation, from both a cooperative and noncooperative perspective.

Our work differs from the above line of research in its unique models. With network coding, the traditional shortest or min-cost single path routing is no longer efficient. Instead, multiple opportunistic paths can be formed by taking advantage of the relays' overhearing capability. To fully explore the wireless broadcast advantage, we need to model a broadcast MAC instead of the traditional unicast MAC scheduling scheme. We are particularly interested in modeling the propagation of coded information flows in a lossy network.

Optimization based approaches to network coding have been extensively studied, but mostly confined to multicast in wireline networks (see *e.g.* [10]). [11] adopted a pricing based algorithm that creates incentives for selfish users to cooperatively achieve the wireline multicast capacity. In [6], the authors pointed out that network coding may also improve energy efficiency for unicast sessions in wireless networks. They proposed a min-cost problem to determine the transmission rate of each node. This inspired the design and implementation of the MORE protocol [5]. Since uncontrolled dissemination of coded packets results in redundant transmissions, MORE uses a heuristic algorithm that tells how many incoming packets a relay should wait before generating a new packet. Unfortunately, this heuristic omits the possible congestion effects caused by multiple forwarders having new packets to transmit. The problem is especially pronounced for multiple unicast sessions which tend to suffer from congestion. In addition, it remains an open problem how many coded packets the source node has to transmit so as to save redundant transmissions while ensuring decodability at the destination. Solving the above problems is the exact objective of the *Dice* framework. Instead of determining the number of packets, we allow each player (a source-destination pair) to determine the encoding and broadcast rate of its source and forwarders in a decentralized manner, taking into account the competition among players, and seeking for optimized bandwidth usage and congestion avoidance.

An unpublished work [12] addressed similar resource allocation problem based on a stochastic network optimization approach. [12] extended the optimal backpressure algorithm for network coding based multicast [13] to the unicast case. The backpressure algorithm assumes that each intermediate forwarder has the queue size information at all downstream nodes, which is infeasible due to the difficulty of real-time feedback in a lossy wireless network. Such queuing information is not required in the *Dice* framework.

To the best of our knowledge, *Dice* represents the first attempt towards a *game theoretic* framework to allocate scarce resources among competing flows with wireless network coding. The upshot of wireless network coding with opportunistic multipath routing is primarily reflected in its *greedy* nature, in that all possible transmission opportunities are fully utilized to maximize throughput. Such a greedy nature can easily lead to bottlenecks and congestion, when multiple "players" are engaged in the game. *Dice* is designed to resolve such conflicts of interest among competing flows.

3. SYSTEM MODELS AND PROBLEM FORMULATION

In this section, we introduce the network models that form the basis for the *Dice* framework. Specifically, we describe the basic operations of a multipath network coding (MNC)

[1]Dice: a code on a piece of instrument used in a game.

protocol, in contrast with the traditional single-path and multipath routing protocols in wireless ad hoc and mesh networks. Such a protocol is further abstracted from a game theory perspective, where the strategies of players corresponds to the network constraints of the protocol.

3.1 Basic Operations of an MNC Protocol

The MNC protocol is designed for long lived unicast sessions in lossy wireless networks. In MNC, the source node continuously generates packet streams from a group of data blocks using a random linear network coding scheme. Coded packet streams flow through multiple paths towards the destination. Intermediate forwarders can refresh the packet streams by re-encoding existing packets and broadcasting the coded packets to downstream nodes. Once a sufficient number of packets accumulates at the destination, the original group of data blocks can be recovered. Subsequently, an uncoded ACK is sent back to the source (preferably using traditional routing), allowing it to start operating on a new group of data blocks. The detailed network coding operations in MNC are as follows.

3.1.1 Basic Encoding and Decoding

In random linear network coding, both the encoding and decoding operations can be regarded as matrix multiplication over a Galois field. Specifically, the source data is grouped into *generations*, and further split into *data blocks*. Each generation is an $n \times m$ matrix B, with rows being the n blocks of the generation, and columns the m bytes of each data block (usually $m \gg n$). The encoding operation produces a linear combination of the original blocks by $X = R \cdot B$, where R is an $n \times n$ matrix composed of randomly selected coefficients in the Galois field $GF(2^8)$. The *coded blocks* (rows in X), together with the *coding coefficients* (rows in R), are packetized and flow as packet streams towards the destination.

The decoding operation at the destination node, in its simplest form, is the matrix multiplication $B = R^{-1} \cdot X$, where each row of X represents a coded block and each row of R represents the coding coefficients accomplished with it. The successful recovery of the original data blocks B requires that the matrix R be of full rank, *i.e.*, the destination must collect n independent coded blocks.

3.1.2 Operations At Intermediate Nodes

Intermediate relays can refresh the packet streams by *re-encoding* incoming packets and *broadcasting* the re-encoded packets to downstream nodes. The re-encoding operation replaces the coding coefficients accomplished with the original coded packets with another set of random coefficients. For instance, consider the existing coded packets at an intermediate node as rows in the matrix Y, which from the viewpoint of the source node was obtained using $Y = R_y \cdot B$ (B is the original uncoded packets and R_y is the random coefficients). Then the intermediate node may produce a new code block by re-encoding existing packets as $Y' = R' \cdot R_y \cdot B = R'_y \cdot B$. As a result, the original coefficients R_y are replaced by R'_y.

To reduce futile transmissions, a relay stores an incoming packet only if it is a fresh packet independent of existing received ones, *i.e.*, it is *innovative*. This ensures that if a relay accepts a new packet, it is also able to produce and contribute an independent coded packet to the packet streams.

3.1.3 Why Resource Allocation?

The very nature of randomized network coding makes it possible to guarantee full reliability even under severe losses, since the probability of decoding failure approaches 0 as more and more packets accumulate at the destination [6]. However, it is a nontrivial task to tailor the random linear network coding for efficient unicast, given the possible redundancy induced by linearly dependent packets, and congestion caused by competing sessions. The key contribution of *Dice* lies in its ability to manage the encoding, broadcasting and multipath routing in an optimized manner, in order to maximize the performance of lossy wireless networks. This is mainly achieved by the game theoretic framework which optimizes the broadcast/coding rate for relay and source nodes, and thus resolves the conflict among competing flows.

3.2 A Game Theoretic Formulation

Game theory analytically models the interactions among individual decision makers called *players*. To play the game, each player selects a *strategy* from a set of possible strategies (referred to as *strategy space*). The outcome of the game is evaluated by payoff or utility functions representing the preferences of individual players. If each player tends to selfishly move towards his own beneficial point, this is called a *noncooperative game*. A noncooperative game reaches *Nash equilibrium* if no player can increase his own payoff by varying his strategies while other players' strategies remain unchanged. The primary application of noncooperative game theory has centered around the existence, uniqueness and efficiency of a Nash equilibrium. On the other hand, *cooperative game theory* models the situations where players coordinate their actions so as to reach a solution of public interest. It is concerned with the formalization of fair resource sharing and cost allocation problems.

In the *Dice* framework, each *player* is a network user that can manipulate a single *session*, *i.e.*, the end-to-end transmission from a source to a destination node. The player commands intermediate relay nodes to behave according to his preference. For player k, the utility function is $U(\lambda^k)$, where λ^k is the throughput of the corresponding session. His strategy is to assign encoding and broadcast rates to all transmitters (including the source and intermediate relays), and to allocate an information flow rate to each forwarding link, in order to increase his payoff. The encoding rate and broadcast rate are the same, *i.e.*, a node attempts to broadcast a coded packet immediately after it is encoded. Under both the cooperative and noncooperative framework, the strategy space of participating players is constrained by the underlying network models, which we detail in the sequel.

3.2.1 The Opportunistic Multipath Routing Model

Before the actual transmission, an MNC protocol performs a *node selection* procedure in a decentralized manner [5], such that each intermediate relay is closer to the destination than its predecessors. Denote the resulting topology as $G(V^k, E^k)$ for each player k, where E^k is the set of directed links and V^k is the set of nodes involved in session k. Note that unlike traditional multipath routing protocols, no explicit path selection is needed. All nodes selected by k will opportunistically contribute to the unicast.

Without loss of generality, we focus only on the model for player (session) k, as reflected in the superscripts of all variables. Denote x_{ij}^k as the information flow rate from node i to j, which is the average injection rate of innovative packets on link (i, j). Since fresh information must be conserved at each relay, we have:

$$\sum_j x_{ij}^k - \sum_j x_{ji}^k = \pi(i, k), \forall i \in V^k, (i, j) \in E^k \quad (1)$$

where

$$\pi(i,k) = \begin{cases} \lambda^k & \text{if } i = S^k, \\ -\lambda^k & \text{if } i = T^k, \\ 0 & \text{otherwise.} \end{cases}$$

S^k, T^k and λ^k denote the source, destination and throughput of player k.

3.2.2 *The Broadcast MAC Model*

The MAC layer is responsible for scheduling all transmitters, so as to avoid the collisions at the receiver caused by interfering transmissions. In a unicast MAC protocol, the primary concern is to avoid the interference among *links*. In contrast, a broadcast MAC must ensure that collision does not happen for any of the transmitting *nodes*. Formally, a broadcast transmission from node i is *collision free* if and only if all other transmitters are outside the range of any downstream receiver of node i. Unlike the traditional unit-disk model, we define *range* as the distance where reception probability falls below a small threshold. With this definition, it is fair to assume that the interference range equals to the transmission range. Note that the notion of collision free lies only within the MAC layer, and is independent of packet *loss* due to path-loss or fading effects.

Our MAC model determines the feasible broadcast rate that each transmitter can achieve, subject to wireless interference. For unicast MAC protocols, it is well known that the maximal clique model gives a necessary condition for feasible link rate, while the independent set model imposes a sufficient condition [14]. However, both problems are NP-hard and the exact characterization of the rate region remains an open problem. Similar results apply to the broadcast MAC, where the basic element in a clique (or independent set) is a node rather than a link. To gain insights, we have to tradeoff accuracy of the models for the simplicity of the resource allocation algorithm running at each player. Specifically, we propose a necessary condition for a broadcast MAC that is similar to the clique model. A sufficient condition can be derived following similar analysis for the unicast MAC [14]. In Sec. 4, we will discuss how the *Dice* framework can be generalized to other MAC models.

Assume an ideal collision-free broadcast MAC protocol exists and is based on slotted scheduling. Let $B_i^k[t]$ be the decision variable indicating whether node i is transmitting player k's data in slot t. According to the above definition of collision, a necessary and sufficient condition for collision free schedule is:

$$\sum_k B_i^k[t] + \sum_k \sum_{j \in R(i)} B_j^k[t] \leq 1, \forall i \in V^k \backslash S^k, \quad (2)$$

i.e., in each time slot, any receiver i allows the broadcast transmission from at most one transmitter within its range (including itself), denoted as $R(i)$. Note that the source node S^k is excluded because all nodes in V^k are receivers except S^k. Assume the period of a schedule is T, then the *broadcast rate* of node i is:

$$\sum_k b_i^k = \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \sum_k B_i^k[t], \quad (3)$$

Apply (3) to (2), we have:

$$\sum_k b_i^k + \sum_k \sum_{j \in R(i)} b_j^k \leq C, \forall i \in V^k \backslash S^k \quad (4)$$

where $C = \frac{1}{T}$ is the MAC layer capacity, which equals to the maximal broadcast rate of a node when no interferer presents. In consequence, for any feasible broadcast schedule, (4) must be necessarily satisfied. It is necessary but insufficient as we transformed an integer variable $B_i^k[t]$ into a continuous one b_i^k by averaging.

3.2.3 *The Coding Model*

It is straightforward that the injection rate of innovative information flow along a link (i, j) must not exceed the corresponding unicast transmission rate, *i.e.*,

$$b_i^k \cdot p_{ij} \geq x_{ij}^k, \quad (5)$$

where p_{ij} is the reception probability of link (i, j). However, this is not a tight bound for x_{ij}^k. It is possible that a collision-free, loss-free packet arriving from (i, j) is linearly dependent with existing packets stored on j, which may have been delivered from some other link (k, j). Nevertheless, (5) still represents a reasonable approximation in a lossy environment. To see this, consider an elementary scenario where S pushes the coded packet streams to T through two paths, each containing one forwarder, denoted as u and v ($u \notin R(v)$), respectively. If u, v have different set of linearly independent packets from S, then they can generate linearly independent packets for T with high probability. Furthermore, when links are lossy, the probability for u, v to have the same set of linearly dependent packets is as low as $q = (p_{Su} \cdot p_{Sv})^t$, where t is the sequences of packets broadcasted from S, which increments from 1 to as large as the generation size. Obviously, q remains small most of the time, thus it is fair to assume u and v can independently contribute information to T. In [6], an exact characterization of the coding model is provided, but it involves an exponential number of constraints, making the problem intractable.

We remark that the above models involve approximations to the behavior of an actual wireless mesh network. This is largely due to the undetermined MAC layer rate region and probabilistic nature of lossy wireless networks. However, from the player's point of view, this formulation includes all the tractable information that they can manipulate to induce a better payoff. In effect, the essential objective of the *Dice* framework is not to compute an absolutely optimal value, but to derive optimized MNC algorithms that highlight the importance of resource allocation for the players. We will demonstrate in Sec. 6 that these models can indeed lead the players towards a much higher level of efficiency and fairness tradeoff than existing heuristic algorithms.

4. THE COOPERATIVE GAME THEORETIC FRAMEWORK

As a first contribution in the *Dice* framework, we model the MNC resource allocation problem as a cooperative Nash bargaining game in which the players select strategies through collaboration.

4.1 The Nash Bargaining Game Formulation

A pivotal element of the Nash bargaining game is a *disagreement point* which takes effect when no agreement can be reached through negotiation. The disagreement point provides an incentive for the players to jointly move towards an optimal point called *Nash bargaining solution* (NBS) [15]. NBS captures the notion of social efficiency and fairness using a set of axiomatic definitions. In this paper, we highlight the following properties of the NBS:

Pareto optimality. This implies that there is no other point that leads to strictly higher utility than the NBS.

Fairness. The notion of fairness for NBS involves symmetry, scale invariance and independence of irrelevant alternatives (for a formal definition, see [15]). These properties establish a sufficient condition for the well known concept of *proportional fairness.*

Assume there are n players, whose strategy space $\mathcal{S} = S_1 \times S_2 \times \cdots \times S_n$. Each player k has a metric function $f_k(\boldsymbol{s})$ evaluating the efficiency of a strategy $\boldsymbol{s} \in \mathcal{S}$. If $f_k(\cdot)$ is concave, and $\mathcal{S}$ is a convex and compact set, then the NBS satisfying the above properties is unique, and is the solution to the following optimization problem [16]: max $\prod_{k=1}^{n}(f_k(\boldsymbol{s}) - f_k^d)$, subject to: $\boldsymbol{s} \in \mathcal{S}^0$, where f_k^d is the disagreement point and $\mathcal{S}^0$ is the subset of strategies that achieve higher payoff than the disagreement point.

In the MNC resource allocation problem, it is natural to set the metric function as the throughput of a player, *i.e.*, $f_k(\boldsymbol{s}) = \lambda^k$. The disagreement point can be set to a null vector, in order to prevent the failure of consensus. Consider the NBS framework under opportunistic multipath routing (Sec. 3.2.1), broadcast MAC scheduling (Sec. 3.2.2) and RLC coding constraints (Sec. 3.2.3): $\max \prod_{k=1}^{n} f_k(\boldsymbol{s})$, subject to $\boldsymbol{s} \in \mathcal{S}$, which is equivalent to:

$$\max \quad \sum_{k=1}^{n} \ln(\lambda_k), \tag{6}$$

subject to:

$$\begin{aligned} \sum_j x_{ij}^k - \sum_j x_{ji}^k &= \pi(i,k), i \in V^k && (7)\\ x_{ij}^k &\geq 0, (i,j) \in E^k && (8)\\ \sum_k b_i^k + \sum_k \sum_{j \in R(i)} b_j^k &\leq C, i \neq S^k && (9)\\ b_i^k p_{ij} &\geq x_{ij}^k, i \neq T^k && (10) \end{aligned}$$

We consider λ_k as a special element x_{ts}^k in the vector $\boldsymbol{x}$, indicating the conceptual flow from the destination node back to the source. All vectors (bold characters) are obtained by stacking the corresponding variables in a row. Thus the strategy of a player k is a vector consisting of the information flow rate $x_{ij}^k, (i,j) \in E^k$, and broadcast rate $b_i^k, i \in V^k$ which is also the rate that node i re-encodes the data blocks for player k. Since the strategy space $\mathcal{S}$ defined by constraints (7), (8), (9) and (10) is affine and compact, and the metric function $f_i(\boldsymbol{s})$ is concave, the above optimization will result in the unique Nash bargaining solution.

Unfortunately, the NBS concept itself does not explicitly specify the procedure leading to a consensus. Here we use the dual decomposition method [17] to solve the problem (6) and derive a decentralized algorithm that converges to the NBS.

4.2 The Cooperative Dice

Consider the partial Lagrangian of the cooperative *Dice* framework (6):

$$\begin{aligned} &L(\boldsymbol{x}, \boldsymbol{b}, \boldsymbol{\beta}, \boldsymbol{\gamma}) \\ &= \ln(\lambda_k) + \sum_i \beta_i (C - \sum_k b_i^k - \sum_k \sum_{j \in R(i)} b_j^k) \\ &\quad + \sum_k \sum_{(i,j) \in E^k} \gamma_{ij}^k (b_i^k p_{ij} - x_{ij}^k) \\ &= \ln(\lambda_k) - \sum_i \sum_k [(\beta_i + \sum_{j \in R(i)} \beta_j - \sum_{(i,j) \in E^k} \gamma_{ij}^k p_{ij}) b_i^k] \\ &\quad - \sum_k \sum_{(i,j) \in E^k} \gamma_{ij}^k x_{ij}^k + \sum_i \beta_i C \end{aligned} \tag{11}$$

which is obtained by relaxing the scheduling constraint (9) and the coding constraint (10).

Then the primal optimization problem is:

$$d(\boldsymbol{\beta}, \boldsymbol{\gamma}) = \max_{\boldsymbol{x}, \boldsymbol{b}} L(\boldsymbol{x}, \boldsymbol{b}, \boldsymbol{\beta}, \boldsymbol{\gamma}) \tag{12}$$

subject to the routing constraint (7). The corresponding dual problem is $\min_{\boldsymbol{\beta}, \boldsymbol{\gamma}} d(\boldsymbol{\beta}, \boldsymbol{\gamma})$. According to decomposition theory, the subgradients for the dual variables β_i and γ_{ij}^k are:

$$\begin{aligned} G_{\beta_i} &= C - \sum_k b_i^k - \sum_k \sum_{j \in R(i)} b_j^k \\ G_{\gamma_{ij}^k} &= b_i^k p_{ij} - x_{ij}^k \end{aligned}$$

Therefore, the dual variables can be updated using the subgradient method:

$$\begin{aligned} \beta_i(t+1) &= [\beta_i(t) - \theta_2 \cdot G_{\beta_i}]^+ && (13)\\ \gamma_{ij}^k(t+1) &= [\gamma_{ij}^k(t) - \theta_1 \cdot G_{\gamma_{ij}^k}]^+ && (14) \end{aligned}$$

Here $[\cdot]^+$ denotes projection on the non-negative domain. θ_1 and θ_2 are the step sizes. The rules for choosing step sizes will be discussed in Lemma 1.

Both dual variables have economic interpretations. β_i is the price used to charge wireless node i for its violation of the bandwidth resource constraint, *i.e.*, the MAC constraint (4). The adjustment of such a price can be performed by a MAC level protocol that is independent of individual players. The players negotiate with each other in order to reduce the MAC price, *i.e.*, the cost of using the bandwidth. γ_{ij}^k is the price that each player uses to adjust the amount of information flows on the forwarding links. The minimum price corresponds to the opportunistic multipath routing scheme with maximum efficiency.

At the same time, all players need to solve the primal problem (12) in order to maximize their payoff. Observe that (12) can be decomposed with respect to the broadcast rate vector $\boldsymbol{b}$ and the information flow rate vector $\boldsymbol{x}$, resulting in two subproblems: *SUB1*:

$$\max_{\boldsymbol{x}} \sum_{k=1}^{n} \ln(\lambda_k) - \sum_k \sum_{(i,j) \in E^k} \gamma_{ij}^k x_{ij}^k \tag{15}$$

subject to: constraints (7) and (8). And *SUB2*:

$$\max_{\boldsymbol{b}} \sum_k \sum_{j:(i,j) \in E^k} \gamma_{ij}^k p_{ij} b_i^k - \sum_k \sum_i [(\beta_i + \sum_{j \in R(i)} \beta_j) b_i^k]$$

subject to: $0 \leq b_i^k \leq C.$ (16)

Note that (16) is an implicit constraint in the original problem (6). Owning to the above decomposition, we obtain a modularized optimization of two subproblems: the multipath opportunistic routing problem (*SUB1*), and the broadcast/encoding rate allocation problem (*SUB2*). These two problems are solved separately and coordinated by the pricing vector $\boldsymbol{\gamma}$.

4.2.1 The multipath opportunistic routing problem (SUB1)

It is straightforward to see that problem *SUB1* is decomposable with respect to each player. In particular, both the objective and the constraint can be decoupled, thus each player performs an individual optimization:

$$\max_{\boldsymbol{x}^k} \quad \ln(\lambda_k) - \sum_{(i,j)\in E^k} \gamma_{ij}^k \cdot x_{ij}^k \tag{17}$$

subject to: constraint (7) and (8).

We observe that the above optimization tends to give a single path solution, as we relaxed the capacity constraint (10). To see this, consider the equivalent flow-path formulation (P is the set of paths):

$$\max_{\boldsymbol{y}} \quad \ln(\sum_{r\in P} y_r) - \sum_{r\in P} p_r \cdot y_r \tag{18}$$

where y_r indicates the amount of flow on path r; p_r is the cost of routing a unit flow, which is obtained by adding up the link cost γ_{ij}^k along the path r. Let $Y = \sum_r y_r$ and $p_{\min}$ be the cost of the min-cost path, then:

$$\ln(\sum_r y_r) - \sum_r p_r y_r \le \ln(Y) - p_{\min} Y \le \ln(\frac{1}{p_{\min}}) - 1$$

equality is achieved iff we choose the min-cost path.

The loss of a multipath solution is mainly due to the non-strict concavity of the primal Lagrangian function (17) (since obviously its Hessian with respect to $\boldsymbol{x}$ is not negative definite), which implies that a dual optimal solution does not necessarily produce a primal optimum. Thus we design the following mechanism to recover the primal solution. Following the above flow-path reasoning, we send $\frac{1}{p_{\min}}$ units of traffic through the shortest path in each optimization iteration. Correspondingly, for each link (i,j) along this path, $x_{ij}^k = \frac{1}{p_{\min}}$. Then we use an averaging based primal recovery method [18] to retain the feasibility of the primal solution. Specifically, we take an equally-weighted average of the resulting flow rate in each iteration t:

$$\bar{x}_{ij}^k(t) = \frac{1}{t}\sum_{m=1}^{t} x_{ij}^k(m) \tag{19}$$

where $x_{ij}^k(m)$ is the solution for link (i,j) in iteration m. Note that the link cost γ_{ij}^k may vary throughout the process of iterative optimization. Within each iteration, only a single shortest path is selected. However, with (19), we not only obtain a primal feasible solution, but also a multipath routing scheme that appropriately assigns rates to all links.

4.2.2 The rate allocation problem (SUB2)

Problem *SUB2* can be solved separately for each b_i^k:

$$\max_{b_i^k} \quad (\sum_{j:(i,j)\in E^k} \gamma_{ij}^k p_{ij}) b_i^k - (\beta_i + \sum_{j\in R(i)} \beta_j) b_i^k \tag{20}$$

Obviously this is a linear objective, which will cause oscillations if solved directly. Thus we add a quadratic regulation term to make it strictly concave. Specifically, at each step, the b_i^k is updated as follows:

$$\begin{aligned} b_i^k(t+1) &= \arg\max_{b_i^k} \Big\{ (\sum_{j:(i,j)\in E^k} \gamma_{ij}^k p_{ij}) b_i^k \\ &\qquad - (\beta_i + \sum_{j\in R(i)} \beta_j) b_i^k - \epsilon |b_i^k - b_i^k(t)|^2 \Big\} \\ &= b_i^k(t) + \frac{1}{2\epsilon}(\sum_{j:(i,j)\in E^k} \gamma_{ij}^k p_{ij} - (\beta_i + \sum_{j\in R(i)} \beta_j)) \end{aligned}$$

where ϵ is a small positive constant. As the algorithm converges, the quadratic term approximates zero, thus the above update approximates the optimal value of b_i^k. For a validation of such a method, see Sec. 3.4 in [19].

Note that the opportunistic multipath routing (*SUB1*) algorithm is based on iterations of the shortest path algorithm, which has well-established decentralized solutions. The rate allocation scheme (*SUB2*) only involves localized updates for each player and the corresponding forwarding nodes. Furthermore, the dual pricing mechanism uses constant step sizes, and only requires nodes in the same neighborhood to communicate the dual variable β_i to each other. Overall, the cooperative *Dice* can be realized in a decentralized manner. Another noteworthy point is that the term G_{β_i} is essentially the *congestion price* in the neighborhood of node i. Such a congestion price can be locally generated by a MAC protocol itself (for such a MAC, see [20]). Therefore, the above *Dice* algorithm can be extended to other MAC models, as long as the MAC protocol can generate the congestion price by itself.

4.2.3 Proof of optimality and convergence

We proceed to prove that the above dual decomposition based algorithm, together with the primal recovery and regulation mechanism, indeed results in a feasible and optimal solution corresponding to the NBS. In particular, the above algorithm convergences to the NBS at a linear rate. The proof shares a similar spirit with recent work on the convergence of primal recovery techniques when using the subgradient method [18]. The difference is that our primal problem is ill conditioned, especially in the linear part for the vector $\boldsymbol{b}$ where we added a quadratic term to regularize the primal function. Our proof borrows the following important result from [18]: If Slater's condition holds and the subgradients are bounded, the sequence of Lagrange multipliers are bounded, *i.e.*, $||\boldsymbol{\beta}(t)|| \le B, ||\boldsymbol{\gamma}(t)|| \le R$, The bounds B and R depend on the dual optimal solution, the constraint violation at iteration t, and the value of an initial point satisfying the Slater's condition.

In our problem, the Slater's condition holds since we can easily construct a feasible solution by assigning each player a small amount of information flows so that their aggregate bandwidth requirements do not violate the broadcast MAC constraint (4) and the coding constraint (5). Furthermore, the subgradients are bounded since the problem is concave, and the constraint set is affine and compact [18].

For brevity, we reformulate the *Dice* optimization problem (6) as: $\max_{\boldsymbol{x}} f(\boldsymbol{x})$, s.t., $r_i(\boldsymbol{x}) = 0$, $g_i(\boldsymbol{b}) \ge 0$, $h_i(\boldsymbol{x}, \boldsymbol{b}) \ge 0$, corresponding to the routing constraint, MAC constraint and coding constraint, respectively. In addition, we have $\boldsymbol{x} \ge \boldsymbol{0}$, and $\boldsymbol{C} \ge \boldsymbol{b} \ge \boldsymbol{0}$, in consistent with the original problem (6). Further, denote $h_i(\boldsymbol{x}(t), \boldsymbol{b}(t)) = h_i(t)$ where t is the iteration index. Obviously $h_i(\cdot)$ can be decomposed: $h_i(\boldsymbol{x}, \boldsymbol{b}) = h_i^{(1)}(\boldsymbol{x}) + h_i^{(2)}(\boldsymbol{b})$. Denote the bounds for subgradients as $||\boldsymbol{h}(t)||^2 \le H, ||\boldsymbol{g}(t)||^2 \le G$. We precede the final result with three lemmas.

Lemma 1: *The primal solutions generated by the cooperative Dice are feasible as* $t \to \infty$.

Proof: The routing constraint $r_i(\boldsymbol{x}) = 0$ is always satisfied since we used iterations of the shortest-path algorithm which maintains flow conservation. Further, when solving problem (15) (*i.e.*, *SUB1*), we used the average value of the vector $\boldsymbol{x}$. Thus we have:

$$h_i(\bar{\boldsymbol{x}}(t), \boldsymbol{b}(t)) = h_i^{(2)}(\boldsymbol{b}(t)) + \frac{1}{t}\sum_{j=1}^{t} h_i^{(1)}(\boldsymbol{x}(j))$$

$$\ge \frac{1}{t}\sum_{j=1}^{t} h_i^{(2)}(\boldsymbol{b}(j)) + \frac{1}{t}\sum_{j=1}^{t} h_i^{(1)}(\boldsymbol{x}(j)) \tag{21}$$

$$= \frac{1}{t}\sum_{j=1}^{t} h_i(j) \ge \frac{1}{t}\sum_{j=1}^{t}\frac{1}{\theta_1}[\gamma_i(j) - \gamma_i(j+1)] \tag{22}$$

$$= \frac{1}{t\theta_1}(\gamma_i(0) - \gamma_i(t+1)) \ge -\frac{1}{t\theta_1}\gamma_i(t+1) \tag{23}$$

Denote $h_i(t)^-$ as the amount of constraint violation for subgradient h_i at iteration t. It is straightforward that:

$$h_i(t)^- = \min(0, h_i(\bar{\boldsymbol{x}}(t), \boldsymbol{b}(t))) \ge -\frac{1}{t\theta_1}\gamma_i(t+1) \tag{24}$$

Since the sequences of Lagrange multipliers are bounded by R, we have:

$$||\boldsymbol{h}(t)^-|| \le \frac{1}{t\theta_1}||\boldsymbol{\gamma}(t+1)|| \le \frac{R}{t\theta_1}$$

which approximates zero as $t \to \infty$. Note that (22) can be derived from the update rule (14). (21) stands under either of the following two technical conditions. First, (20) has non-zero solution and thus $\boldsymbol{b}(t)$ is non-decreasing in general. This can be satisfied by adjusting the step sizes for dual variables $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$, or by allowing multiple sub-iterations of $\boldsymbol{\gamma}$ for each iteration of $\boldsymbol{\beta}$. Second, we can use the same averaging based primal recovery algorithm to approximate a feasible solution for the vector $\boldsymbol{b}$.

Similarly, we can prove that under the above conditions the amount of constraint violation for $g(\cdot)$ approximates zero as $t \to \infty$, *i.e.*, $\lim_{t\to\infty} ||\boldsymbol{g}(\boldsymbol{b}(t))^-|| = 0$. □

Lemma 2: *The cooperative Dice approximates the NBS solution from above at a linear rate.*

Proof: Denote $\boldsymbol{f^*}$ and $\boldsymbol{q^*}$ as the optimal primal and dual value, respectively. Since the objective function $f(\cdot)$ is concave, we have:

$$f(\bar{\boldsymbol{x}}(t)) \ge \frac{1}{t}\sum_{j=1}^{t} f(\boldsymbol{x}(t)) = \frac{1}{t}\sum_{j=1}^{t}[f(\boldsymbol{x}(j)) + \boldsymbol{\gamma}(j)\boldsymbol{h^T}(j)$$

$$+ \boldsymbol{\beta}(j)\boldsymbol{g^T}(j)] - \frac{1}{t}\sum_{j=1}^{t}[\boldsymbol{\gamma}(j)\boldsymbol{h^T}(j) + \boldsymbol{\beta}(j)\boldsymbol{g^T}(j)]$$

$$\ge \boldsymbol{q^*} - \frac{1}{t}\sum_{j=1}^{t}[\boldsymbol{\gamma}(j)\boldsymbol{h^T}(j) + \boldsymbol{\beta}(j)\boldsymbol{g^T}(j)] \tag{25}$$

From Proposition 2(b) in [18], we know that:

$$\boldsymbol{\gamma}(t)\boldsymbol{h^T}(t) \le \frac{1}{2\theta_1}(||\boldsymbol{\gamma}(t)||^2 - ||\boldsymbol{\gamma}(t+1)||^2) + \frac{\theta_1}{2}||\boldsymbol{h}(t)||^2$$

Similar results can be obtained for the multiplier $\boldsymbol{\beta}$. Furthermore, since the *Dice* optimization problem is concave, there is no duality gap, *i.e.*, $\boldsymbol{f^*} = \boldsymbol{q^*}$. Then the inequality (25) can be simplified as:

$$f(\bar{\boldsymbol{x}}(t)) \ge \boldsymbol{q^*} + \frac{||\boldsymbol{\gamma}(t+1)||^2 - ||\boldsymbol{\gamma}(1)||^2}{2t\theta_1} - \frac{\theta_1}{2t}\sum_{j=1}^{t}||\boldsymbol{h}(t)||^2$$

$$+ \frac{||\boldsymbol{\beta}(t+1)||^2 - ||\boldsymbol{\beta}(1)||^2}{2t\theta_2} - \frac{\theta_2}{2t}\sum_{j=1}^{t}||\boldsymbol{g}(t)||^2$$

$$\ge \boldsymbol{f^*} + \frac{||\boldsymbol{\gamma}(t+1)||^2}{2t\theta_1} + \frac{||\boldsymbol{\beta}(t+1)||^2}{2t\theta_2} - \frac{\theta_1 H}{2} - \frac{\theta_2 G}{2}$$

Thus as $t \to \infty$, the iterative cooperative *Dice* algorithm approaches the optimal objective $\boldsymbol{f^*}$ at a linear rate. The distance to the optimal objective $\boldsymbol{f^*}$ depends on the step size θ_1 and θ_2, as well as the subgradient bounds H and G. □

Lemma 3: *The cooperative Dice approximates the NBS solution from below at a linear rate.*

Proof:
$$f(\bar{\boldsymbol{x}}(t)) = f(\bar{\boldsymbol{x}}(t)) + \boldsymbol{\gamma^*}\boldsymbol{h^T}(t) + \boldsymbol{\beta^*}\boldsymbol{g^T}(t) - \boldsymbol{\gamma^*}\boldsymbol{h^T}(t) - \boldsymbol{\beta^*}\boldsymbol{g^T}(t)$$

$$\le \boldsymbol{f^*} - \boldsymbol{\gamma^*}\boldsymbol{h^T}(t) - \boldsymbol{\beta^*}\boldsymbol{g^T}(t)$$

$$\le \boldsymbol{f^*} + ||\boldsymbol{\gamma^*}|| \cdot ||\boldsymbol{h^T}(t)^-|| + ||\boldsymbol{\beta^*}|| \cdot ||\boldsymbol{g^T}(t)^-||$$

Since both $||\boldsymbol{\gamma^*}||$ and $||\boldsymbol{\beta^*}||$ are bounded, while $||\boldsymbol{h^T}(t)^-||$ and $||\boldsymbol{g^T}(t)^-||$ approximate zero at a linear rate as $t \to \infty$, Lemma 3 follows directly. □

Theorem 1. *The cooperative Dice approaches the unique solution of the NBS problem at a linear rate.*

Proof: From Lemma 2 and Lemma 3, we know that the *Dice* algorithm approximates the optimal solution $\boldsymbol{f^*}$ of the NBS problem at a linear rate. Furthermore, Lemma 1 states that the primal solution vector $\boldsymbol{x}$ and $\boldsymbol{b}$ generated by the *Dice* algorithm are feasible. Since the original NBS problem has a concave objective function and affine constraints, it has a unique optimal solution. Therefore, the *Dice* algorithm approximates the unique solution of the NBS problem at a linear rate. □

5. THE NONCOOPERATIVE FRAMEWORK

When players selfishly improve their own payoff without considering the social welfare, it is important to determine the existence, efficiency and fairness of consensus points. For such a noncooperative case, the consensus is characterized by the well known Nash equilibrium point (NEP). The game arrives at a NEP if no player can unilaterally increase his own payoff, *i.e.*,

$$f_i(s_1^*, \cdots, s_i^*, \cdots, s_n^*) \ge f_i(s_1^*, \cdots, s_i, \cdots, s_n^*), \forall s_i \in S_i$$

where s_i^* is the strategy of player i at the NEP. Unlike the Nash bargaining framework for cooperative games, there is no guarantee that a Nash equilibrium can be unique or socially optimal. For instance, consider a one-shot static game where each player attempts to satiate the channel by maximizing their broadcast rate, then all players will achieve zero throughput due to severe collisions. Unfortunately, this catastrophic result is a NEP since no player can increase his throughput by adjusting his own broadcast rate.

To avoid such unfavorable NEPs, we need to design mechanisms that lead all players to a socially optimal point. Here we propose a price based scheme that regulates the behavior of the players. The key problem is to design a payoff function (referred to as the *net payoff*) that relates the utility of each player with the price for achieving the utility. By analogy with the cooperative *Dice*, we assume a MAC level pricing protocol exists and charges each player for accessing the channel. In other words, the noncooperative game is a hierarchical game [21], in which the pricing protocol is the leader at the higher level, while the players are followers at the lower level, which selfishly update their net payoff. In addition, each player needs a pricing mechanism for his forwarding links, so as to efficiently use the bandwidth allocated to him. This involves no competition with other players, and is referred to as the *routing price*. With such a setup, the net payoff of player k can be defined as:

$$f(k) = U(\lambda_k) - \sum_{i \in V^k} P_i^k b_i^k - \sum_{(i,j) \in E^k} Q_{ij}^k x_{ij}^k \qquad (26)$$

where the vector $\boldsymbol{x}$ satisfies the routing constraint (1). According to the economic interpretation of diminishing returns, the utility function $U(\lambda_s)$ should be concave and monotonically increasing. The $\ln(\cdot)$ function is a conventional choice. The routing price Q_{ij}^k is exactly the same as the dual variable γ_{ij}^k for the multipath routing problem in the cooperative *Dice*. As for the MAC price P_i^k, we can derive it directly from the cooperative framework, according to equation (11):

$$P_i^k \;\; = \;\; \beta_i + \sum_{j \in R(i)} \beta_j - \sum_{j:(i,j) \in E^k} \gamma_{ij}^k p_{ij}$$

where $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ are the dual variables.

With the above formulation, the noncooperative *Dice* is reduced to an optimization problem that is exactly the same as the cooperative *Dice*, thereby achieving the same level of efficiency and fairness. In the following theorem, we claim that the resulting solution indeed achieves a Nash equilibrium.

Theorem 2. *Assume the players' net payoff is defined by (26), and a pricing protocol charges each player for using the bandwidth resource, then the selfish players can achieve a Nash equilibrium following the same decentralized optimization as in the cooperative Dice.*

Proof: As shown in Sec. 4, the payoff function of each player is concave, and their strategy space is affine and compact. By Theorem 3.1 in [15], the noncooperative game admits at least one NEP.

We proceed to show that when the price vectors $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ reach optimality (*i.e.*, the duality gap becomes zero), the corresponding set of strategies $\boldsymbol{x}$ and $\boldsymbol{b}$ form a NEP. First of all, the utility function and constraint functions are concave and continuously differentiable. By Theorem 3 in [21], a Nash equilibrium can be achieved by decomposing the original game into a higher-level optimization problem that adjusts the dual variables (the pricing vectors), and a lower level Nash game with the individual payoff function defined in (26). Furthermore, with the primal recovery and quadratic regularization method in Sec. 4, we are able to guarantee a primal feasible and optimal strategy for each player. As a result, the final strategies $\boldsymbol{x}$ and $\boldsymbol{b}$ form a feasible Nash equilibrium. Since we used exactly the same decomposition method to solve for the optimal point of the cooperative *Dice* and the noncooperative *Dice*, the two frameworks achieve the same level of utility. We thus complete the proof for Theorem 2. □

6. EXPERIMENTAL RESULTS

In this section, we first introduce *Drift*, the emulation testbed that we use to implement the multipath network coding protocol. We then present experiments on the performance of the *Dice* framework, in order to validate its performance under practical settings.

6.1 Experimental Environment — The Drift Emulator

Drift is a high performance emulation testbed that we designed for prototyping and validating application layer protocols in large-scale wireless networks. Compared with existing emulators, the main feature of *Drift* is a better tradeoff between scalability and accuracy, which rests on its distributed architecture, efficient packet processing unit and analysis based lower layer models. Running in a server cluster, *Drift* is able to accommodate hundreds of nodes and several MBytes/second traffic in a single server host. Such advantages enable *Drift* to operate efficiently even for those computationally intensive algorithms such as network coding.

As in existing wireless emulation testbeds, application algorithms developed in *Drift* run in real-time and real operating systems. *Drift* directly employs the IP and transport layer protocol stacks in the emulation hosts, simulates the wireless PHY and MAC with specific models, and emulates wireless transmissions over a Gigabit Ethernet. The lower layer models consist of a PHY model that captures the lossy nature of the actual wireless environment, and a MAC model that captures the channel competition among neighboring nodes. We next provide more details of both models and justify their sufficiency for the purpose of evaluation.

PHY model. To model opportunistic packet reception in a lossy wireless environment, the widely used unit-disk graph model, which assumes perfect reception within transmission range, no longer holds. Instead, we use a PHY model based on real-world traces from [22], which empirically maps link distance to the reception probability. With this model, the transmission range is defined as the distance where reception probability falls below a small threshold. Therefore, we can define interference range as the same as transmission range, in consistent with the notion of *range* in the *Dice* framework (Sec. 3.2.2).

MAC model. To model bandwidth allocation among neighboring nodes by the MAC layer, we partition the network into groups, each including an active receiver and all transmitters that may send packets or cause interference to it. The group size equals to the interference range. A node cannot receive packets if it falls in the interference range of an interfering node, and the total broadcast rate of all group members must not exceed the channel capacity. The packet transmission of each member is enabled in a randomized round-robin fashion in order to avoid starvation. Such an abstract scheduling model corresponds to a generic MAC protocol in which interfering nodes can optimally multiplex the channel. Although such a generic MAC does not model protocol details such as RTS/CTS, it serves as the foundation for predicting the performance trends, and ensuring a *fair comparison* among various application-specific protocols.

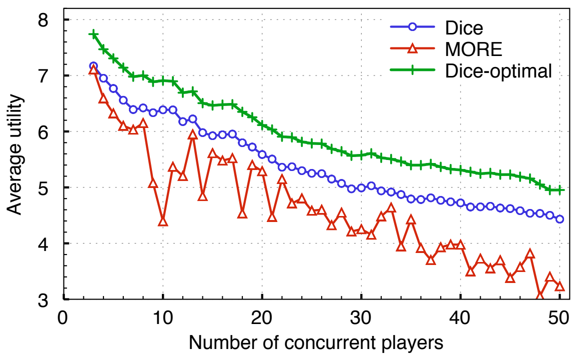

Figure 1: The average per-player utility in a lossy network.

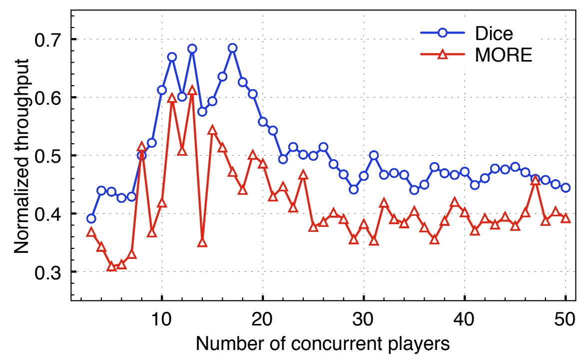

Figure 2: The aggregate throughput as a function of the number of concurrent players.

To evaluate the performance of the MNC protocols, including *Dice* and MORE, we have implemented them within *Drift*. We now proceed to present the experimental results obtained from the *Drift* testbed.

6.2 Performance Evaluation

We explore the potential of *Dice* for multiple unicast sessions on a 50-node random topology with node density 6 (an average of 5 neighbors per node). The transmission and interference range is defined as the distance where reception probability drops to 0.1. The reception probabilities of all links range from 0.11 to 0.93, with an average of 0.48. The channel capacity is 10^4 bytes/second. In the coding module, each generation contains 40 data blocks and each data block is of 1 KB. The source and destination of each player are randomly chosen, with a path length constraint of 4 to 10 hops. The CBR rate is set to half of the channel capacity. Throughput (excluding overhead) is calculated immediately after the source receives the "successfully decoded" ACK from the destination, and then averaged over the entire session.

Fig. 1 demonstrates the average per-player utility as a function of the number of concurrent players. In general, the average utility decreases as more players join. With *Dice*, the players always achieve higher utility than that of MORE, especially when a large number of concurrent players compete for the bandwidth resource. The average utility improvement is 12%. It is worth noting that the actual emulation results of *Dice* have lower utility than those predicted by the optimization framework (6). The main reason lies in the fact that the *Dice* framework is built upon an approximate coding model that overestimates the amount of innovative information flow injecting into each node. In contrast, in the emulation, each node performs independence check on each incoming data block. A data block is dropped unless it is innovative.

Since utility reflects a tradeoff between efficiency (reflected by throughput) and fairness, we further evaluate the performance of both *Dice* and MORE in terms of network throughput and fairness. Fig. 2 plots the aggregate throughput of all players when we vary the number of concurrent players. We observe that *Dice* consistently achieves higher throughput than MORE. The throughput improvement can be up to 47%, with an average of 17%.

A more important performance metric demonstrating the advantage of the game theoretic framework is fairness. As

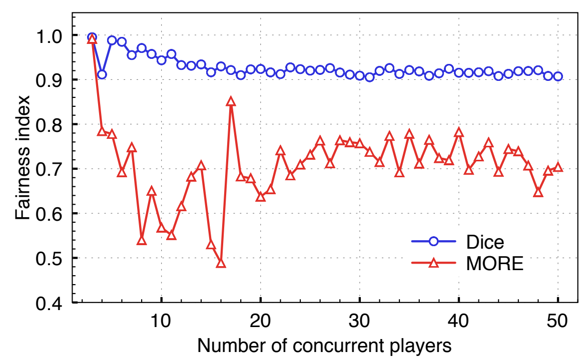

Figure 3: The fairness index of *Dice* and MORE, respectively.

a quantitative evaluation, we compare the Jain's fairness index [23] of *Dice* and MORE. Denote the throughput of player i as T_i and the number of players as n, the fairness index is then defined as $F = \frac{(\sum_{i=1}^{n} T_i)^2}{n \cdot \sum_{i=1}^{n} T_i^2}$. From the results (Fig. 3), we observe that the fairness of MORE varies quite significantly with the number of players. In comparison, *Dice* is able to maintain a high level of fairness. It's fairness index can be up to 91% higher than that of MORE, and 32% higher on average.

The above experiments have verified the claim that *Dice* can lead the players towards a higher level of throughput-fairness tradeoff. The main reason for such advantage over MORE is that it performs optimized bandwidth resource allocation. As an intuitive explanation, we quantify the network congestion status by monitoring the average queue size of relay nodes when running *Dice* and MORE, respectively. Specifically, we start 4 randomly selected players, which require the service of 22 relay nodes for the corresponding end-to-end sessions. We sample the queue size of each relay and then calculate its time average. The results are sorted and plotted in Fig. 4. We see that the time averaged queue size of each relay in *Dice* is lower than 1, with an average of 0.16, implying that *Dice* matches the coding and broadcast rate of intermediate forwarders with the network congestion status. In contrast, the queue sizes of nodes in MORE are largely unbalanced, ranging from 0.05 to 97 (the average is 18.4). Some nodes may not even be able to transmit their encoded data before the data expire. In summary, MORE

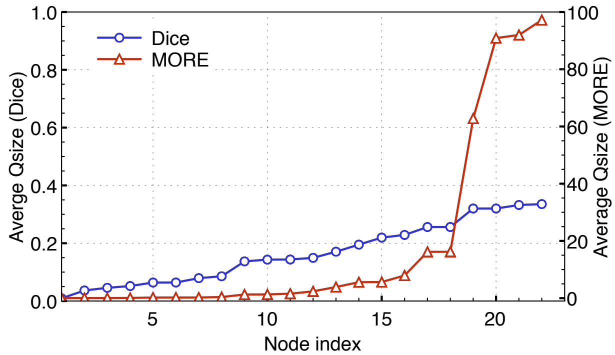

Figure 4: The sorted time averaged queue size of relay nodes.

is not able to match the encoding rate of intermediate forwarders to the amount of resource that they can obtain from the network, hence leading to performance degradation.

7. CONCLUSION

With the *Dice* game theoretic framework, we raised and addressed the problem of resolving conflicts of interest among multiple competing flows with wireless multipath network coding. In *Dice*, we model the problem as a network game, in which participating players share the bandwidth resource through negotiation or competition. In case when players are willing to cooperate, a Nash bargaining solution can be achieved through a decentralized negotiation algorithm. When players are selfish, socially optimal equilibrium point can still be achieved by enforcing pricing mechanisms. For both cases, the players perform a localized optimization of two subproblems: multipath opportunistic routing and broadcast/coding rate allocation. With extensive experiments on an emulation testbed, we demonstrated the effectiveness and potential of *Dice*, in comparison with heuristics in previous work. To our knowledge, this is the first attempt towards a game theoretic framework to negotiate multiple competing flows in the context of wireless network coding.

8. REFERENCES

[1] J. Bicket, D. Aguayo, S. Biswas, and R. Morris, "Architecture and Evaluation of an Unplanned 802.11b Mesh Network," in *Proc. of ACM MobiCom*, 2005.

[2] D. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A High-throughput Path Metric for Multi-hop Wireless Routing," in *Proc. of ACM MobiCom*, 2003.

[3] M. R. Pearlman, Z. J. Haas, P. Sholander, and S. Tabrizi, "On the Impact of Alternate Path Routing for Load Balancing in Mobile Ad Hoc Networks," in *Proc. of ACM MobiHoc*, 2000.

[4] S. Biswas and R. Morris, "ExOR: Opportunistic Multi-hop Routing for Wireless Networks," in *Proc. of SIGCOMM*, 2005.

[5] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading Structure for Randomness in Wireless Opportunistic Routing," in *Proc. of ACM SIGCOMM*, 2007.

[6] D. S. Lun, M. Medard, and R. Koetter, "Network Coding for Efficient Wireless Unicast," in *Proc. of IEEE International Zurich Seminar on Communications*, 2006.

[7] X. Lin, N. B. Shroff, and R. Srikant, "A Tutorial on Cross-Layer Optimization in Wireless Networks," *IEEE Journal on Sel. Areas in Comm.*, vol. 24, no. 8, 2006.

[8] Y. Xue, B. Li, and K. Nahrstedt, "Channel-Relay Price Pair: Towards Arbitrating Incentives in Wireless Ad Hoc Networks," *Wirel. Commun. Mob. Comput.*, vol. 6, no. 2, 2006.

[9] Z. Fang and B. Bensaou, "Fair Bandwidth Sharing Algorithms Based on Game Theory Frameworks for Wireless Ad-hoc Networks," in *Proc. of IEEE INFOCOM*, 2004.

[10] L. Chen, T. Ho, S. Low, M. Chiang, and J. Doyle, "Optimization Based Rate Control for Multicast With Network Coding," in *Proc. of IEEE INFOCOM*, 2007.

[11] Z. Li, "Min-Cost Multicast of Selfish Information Flows," in *Proc. of IEEE INFOCOM*, 2007.

[12] B. Radunovic, C. Gkantsidis, P. Key, P. Rodriguez, and W. Hu, "An Optimization Framework for Practical Multipath Routing in Wireless Mesh Networks," Tech. Rep. MSR-TR-2007-83, Microsoft Research, July 2007.

[13] T. Ho and H. Viswanathan, "Dynamic Algorithms for Multicast With Intra-session Network Coding," in *Allerton*, 2005.

[14] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of Interference on Multi-hop Wireless Network Performance," in *Proc. of ACM MobiCom*, 2003.

[15] F. Forgo, J. Szep, and F. Szidarovszky, *Introduction to the Theory of Games*, Kluwer Academic, 1999.

[16] H. Yaïche, R. R. Mazumdar, and C. Rosenberg, "A Game Theoretic Framework for Bandwidth Allocation and Pricing in Broadband Networks," *IEEE/ACM Trans. Netw.*, vol. 8, no. 5, 2000.

[17] N. Z. Shor, K. Kiwiel, and A. Ruszcayǹski, *Minimization Methods for Non-differentiable Functions*, Springer-Verlag, 1985.

[18] A. Nedic and A. Ozdaglar, "Approximate Primal Solutions and Rate Analysis for Subgradient Methods," Tech. Rep., MIT LIDS, March 2007.

[19] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Prentice-Hall, 1989.

[20] L. Chen, S. Low, M. Chiang, and J. Doyle, "Joint Congestion Control and Media Access Control Design for Ad Hoc Wireless Networks," in *Proc. of IEEE INFOCOM*, 2005.

[21] L. Pavel, "An Extension of Duality to a Game-theoretic Framework," *Automatica*, vol. 43, no. 2, 2007.

[22] J. Camp, J. Robinson, C. Steger, and E. Knightly, "Measurement Driven Deployment of a Two-tier Urban Mesh Access Network," in *Proc. of MobiSys*, 2006.

[23] R. Jain, D. Chiu, and W. Hawe, "A Quantitative Measure of Fairness and Discrimination For Resource Allocation in Shared Computer Systems," Tech. Rep. TR-301, DEC Research, September 1984.