

# Network Coding Aware Resource Allocation to Improve Throughput

Dan Zhang, Kai Su and Narayan B. Mandayam  
WINLAB, Rutgers University  
671 Route 1 South, North Brunswick, NJ 08902  
{bacholic, kais, narayan}@winlab.rutgers.edu

**Abstract**—In the past decade, remarkable progress has been made in the area of network coding in terms of theory, code design, and applications. However, from a cross-layer perspective, the understanding of resource allocation for random network coding (RNC) and its impact on network throughput has been limited. In this article we address the problem of resource allocation for RNC in a general wireless network, using the recent framework that models the dynamics of RNC using a system of differential equations. Based on this framework, we design a gradient algorithm that can be used to improve a variety of throughput objectives.

**Index Terms**—Random network coding, differential equation, resource allocation

## I. INTRODUCTION

Resource allocation for random network coding (RNC) aims to find an optimal allocation of communication resource, such as power, bandwidth, time slots and codes, in support of RNC based transport of information. This problem has been extensively addressed with regard to many applications. The methodologies used in the previous studies [1], [2] have a commonality that, in one way or another the problem of coding at the network level is reduced to the traditional commodity-flow [3] problem. For example, in a network where nodes perform RNC, imposing a routing strategy may compromise valuable benefits of RNC because fundamentally RNC is different from routing in exploiting various diversities the network offers. Specifically, a routing strategy fixes a set of routes over which information is transported and this inevitably incurs loss on the so called multicast advantage [1] of wireless. There are a good number of other resource allocation methods tailored for particular topologies or particular applications [4], [5]. These approaches lack generality and most of them only superficially make use of the advantages that RNC provides. In a recent article [6], a dynamical system model for RNC has been proposed to completely characterize the average behavior of rank evolution – a crucial concept of RNC – that contains all information of interest regarding RNC, such as throughput and decoding delay. While this makes plausible an optimal control theoretic strategy, a simpler resource allocation based on gradient methods also works well, as will be discussed in this article. In general, due to the memory of a dynamical system, gradient methods are often not available. The case of RNC comes off as an exception because of two remarkable properties that have been revealed in previous works [6]–[8].

Specifically, we know with fixed MAC and PHY layer, i.e., fixed resource allocation, the rank of any node (or set)

- 1) increases *linearly* at a fixed rate and
- 2) *linearly* depends on the link rates, until it reaches full rank.

The implications are, because of property 1, fixed allocation corresponds to a fixed throughput, thus memory of the system becomes irrelevant. Because of property 2, the system scales in time (doubling the rates doubles the throughput), thus we can put systems of different rates in a common time frame through proper scaling. These implications are so useful that, for the above mentioned dynamical system, they enable the effective use of the gradient method to construct a state feedback to continuously steer the network towards an ever better allocation of resource.

In what follows we first briefly recap the dynamical system view of RNC that has been elaborated in [6]. The resource allocation problem is then cast in this framework and a gradient based resource allocation is proposed. We analyze its optimality, convergence and discuss its practical design considerations. In the end we illustrate the use of the methodology by designing an optimal power control algorithm for the wireless network where nodes perform RNC.

## II. PROBLEM SETUP

### A. Dynamical System View of Random Network Coding

Consider a wireless network modeled as a hypergraph [6]  $G = (\mathcal{N}, \mathcal{E})$  where  $\mathcal{N}$  is the set of transceiver nodes and  $\mathcal{E}$  are a collection of hyperarcs in the form of  $(i, \mathcal{K})$  pairs where  $i \in \mathcal{N}$  and  $\mathcal{K} \subset \mathcal{N}$ . The broadcasting phenomenon from node  $i$  to an arbitrary set of nodes  $\mathcal{K}$  is thus modeled with the hyperarc  $\mathcal{K}$ . We assume every node in the network performs RNC on the finite field  $\text{GF}(q)$  where  $q$  is the size of the underlying finite field. Whenever a node gets to transmit, it transmits a coded packet that is a random linear combination of all the packets available to it, including the packets it originates and the coded packets it has received. Because RNC takes packets as the atomic object to process, it represents a transport strategy that is decoupled from the underlying MAC or PHY technologies, whose operations are considered independent of RNC. Take an arbitrary node  $i$  for example, typically the MAC layer of node  $i$  determines the transmission (broadcast) rate measured in packets per second. The PHY layer of node  $i$

determines the error probability that a transmitted packet can be correctly decoded at any receiver node  $j$ . In fact, according to [6], the respective effect of MAC and PHY in a wireless network can be shown through two variables:

- $\lambda_i$ : the transmission rate at node  $i$ ;
- $P_{i,\mathcal{K}}$ : the probability that a packet sent from node  $i$  can be correctly decoded by at least one node in  $\mathcal{K}$ .

It is possible to summarize the overall effect of MAC and PHY by defining  $z_{i,\mathcal{K}} = \lambda_i P_{i,\mathcal{K}}$ , which can be conceived as the capacity of the hyperarc  $(i, \mathcal{K})$ . Given  $i \in \mathcal{K}^c$ , we further define the min cut capacity for  $(i, \mathcal{K})$  as

$$c_{\min}(i, \mathcal{K}) = \min_{\mathcal{K} \subset \mathcal{T}, i \notin \mathcal{T}} \sum_{j \in \mathcal{T}^c} z_{j,\mathcal{T}}. \quad (1)$$

As decodability of RNC entirely depends on the number of linearly independent coding vectors, i.e., the *rank*, and as there is no explicit routing, it is important to keep track of the rank at any node. We use  $V_i$  to denote the rank at any node  $i$ . It turns out to be rewarding that we also define the rank at any set  $\mathcal{K}$ , denoted by  $V_{\mathcal{K}}$ , as the number of linearly independent coding vectors that the nodes in  $\mathcal{K}$  collectively possess. Apparently  $V_i$  and  $V_{\mathcal{K}}$  are random increment processes. Suppose the number of source packets is  $m$ . Since every coded packet is a random linear combination of  $m$  source packets,  $i$  can start decoding as soon as  $V_i = m$ . In [6] the authors discussed a concentration phenomenon of RNC, that when  $m$  is sufficiently large,  $V_i$  and  $V_{\mathcal{K}}$  behave in a very predictable way, much like deterministic functions of time. Such a concentration phenomenon has also been observed and discussed even earlier in [8]. Its implication triggers the conjecture that  $V_i$  and  $V_{\mathcal{K}}$  may be modeled as a system of differential equations. This is indeed true and, when MAC and PHY are fixed such that  $z_{i,\mathcal{K}}$  are known parameters, these equations are explicitly written down in [6] as

$$\dot{V}_{\mathcal{K}} = \sum_{i \in \mathcal{K}^c} z_{i,\mathcal{K}} (1 - q^{V_{\mathcal{K}} - V_{\{i\} \cup \mathcal{K}}}), \quad \forall \emptyset \neq \mathcal{K} \subset \mathcal{N}. \quad (2)$$

If we stack  $V_{\mathcal{K}}$  into a column vector  $\mathbf{V}$  and stack all  $z_{i,\mathcal{K}}$  in  $\mathbf{z}$ , we may write (2) in a more compact form

$$\dot{\mathbf{V}} = \mathbf{f}(\mathbf{V}, \mathbf{z}) \quad (3)$$

where  $\mathbf{f}$  is the stacking of the right-hand sides of (2). Note the instantaneous throughput of set  $\mathcal{K}$  is given by  $\dot{V}_{\mathcal{K}}$ . Suppose node 1 is the only source node. When  $m \rightarrow \infty$ , [6] showed that

$$\dot{V}_{\mathcal{K}}(t) = c_{\min}(1, \mathcal{K}), \quad \text{when } V_{\mathcal{K}}(t) < m, \quad (4)$$

i.e., the instantaneous throughput is almost a constant if  $m$  is sufficiently large. The fact that when  $\mathbf{z}$  is fixed and  $m$  is large,  $\dot{\mathbf{V}}$  is constant can be written in a more compact form

$$J_{\mathbf{V}} \mathbf{f}(\mathbf{V}, \mathbf{z}) = 0, \quad (5)$$

where  $J_{\mathbf{V}} \mathbf{f}(\mathbf{V}, \mathbf{z})$  is the Jacobian of  $\mathbf{f}(\mathbf{V}, \mathbf{z})$ . Another useful observation from (2) is that  $\dot{V}_{\mathcal{K}}$  only linearly depends on  $z_{i,\mathcal{K}}$ . Mathematically, this implies

$$(J_{\mathbf{z}} \mathbf{f})(\mathbf{V}, \mathbf{z}) = (J_{\mathbf{z}} \mathbf{f})(\mathbf{V}). \quad (6)$$

While (6) is precise, (5) is a good approximation with large  $m$ . Equations (5) and (6) constitute the basis for the resource allocation strategy to be discussed.

### B. Resource Allocation with the Dynamical System Model

Since RNC is decoupled from MAC and PHY layers, it is particularly easy to discuss resource allocation and its effect on RNC at an abstract level. When there is a single resource to be allocated to  $N$  nodes, the allocation scheme is represented by a  $N$ -dimensional column vector  $\mathbf{r}$ , whose  $i$ -th component represents the allocation to node  $i$ . Consequently, the effect of resource allocation is captured in the mapping

$$\lambda_i = \lambda_i(\mathbf{r}) \quad \text{and} \quad P_{i,\mathcal{K}} = P_{i,\mathcal{K}}(\mathbf{r}), \quad (7)$$

or more concisely,  $z_{i,\mathcal{K}} = z_{i,\mathcal{K}}(\mathbf{r})$ . In vector form, we may write  $\mathbf{z} = \mathbf{z}(\mathbf{r})$ .

An allocation scheme is invariably a deliberate design to achieve a certain objective. In this article, we assume the objective is a function of throughputs, which can be solved with the system of differential equations in (2). Specifically, we assume the objective to be maximized can be written as  $T(\dot{\mathbf{V}})$  where  $T$  can be any continuous function in the first quadrant with partial derivatives defined almost everywhere. As a result, we wish to solve

$$\text{maximize} \quad T(\dot{\mathbf{V}}) \quad (8)$$

$$\text{subject to} \quad \dot{\mathbf{V}} = \mathbf{f}(\mathbf{V}, \mathbf{z}) \quad (9)$$

$$\mathbf{z} = \mathbf{z}(\mathbf{r}) \quad (10)$$

Equation (8) takes a distinct form from previous resource allocation problems for RNC. Rather than treating RNC as a static transport strategy, (8) looks at it from the dynamical system point of view, which means the allocation may also be dynamic and changes over time. Moreover, it can take the form of state feedback

$$\dot{\mathbf{r}} = \mathbf{g}(\mathbf{V}, \dot{\mathbf{V}}) \quad (11)$$

where  $\mathbf{g}$  is some function of  $\mathbf{V}$ . The second distinction is that (8) is the first formulation that takes full multicast advantage of a wireless network, while previous formulations inevitably break down the wireless transmission into independent links to approximately convert the wireless network to a wired network. Doing so, they lose a portion of the multicast advantage.

### III. GRADIENT BASED RESOURCE ALLOCATION

While a problem of the form of (8) falls in the traditional category of optimal control and is usually solved using calculus of variations, there are a few reasons that this approach may not be appropriate. First, calculus of variations is computationally intensive as expensive, which can be practically prohibitive in wireless networks. Second, the wireless communication environment can be dynamic (e.g., due to fading or mobility), and may require certain adaptivity in the devices. Fortunately, when we restrict our objective  $T$  to be a function of throughputs, the special problem structure of RNC

can be utilized to derive a simple gradient based control which is adaptive.

Consider the objective function  $T(\dot{\mathbf{V}})$  which can be any function that has a gradient almost anywhere in the first quadrant. Our choice of feedback (see (11)) is  $\mathbf{g} = \nabla_{\mathbf{r}}T(\dot{\mathbf{V}})$ . Note without (13), it is not clear how to compute  $\nabla_{\mathbf{r}}T(\dot{\mathbf{V}})$  because  $\dot{\mathbf{V}}$  is apparently affected by a number of factors including topology, transmission rates and the coding operation. By applying (2), the effect of the resource vector  $\mathbf{r}$  is abstracted in (10), the topology information encoded in the partially ordered index  $\mathcal{K} \subset \mathcal{N}$  and the dynamism represented as the differential operator. The system of equations in (2) thus allow us to find an expression for the feedback as<sup>1</sup>

$$\dot{\mathbf{r}} = aJ_{\mathbf{r}}^{\top} \mathbf{z} J_{\mathbf{z}}^{\top} \mathbf{f} \nabla_{\dot{\mathbf{V}}} T \quad (12)$$

where  $\nabla_{\dot{\mathbf{V}}}T$  is the gradient of  $T(\dot{\mathbf{V}})$ ,  $J_{\mathbf{z}}\mathbf{f}$  the Jacobian of  $\mathbf{f}(\mathbf{V}, \mathbf{z})$  with respect to  $\mathbf{z}$ ,  $J_{\mathbf{r}}(\mathbf{z})$  the Jacobian of  $\mathbf{z}(\mathbf{r})$ . Note we have introduced  $a > 0$  as a feedback gain parameter. Combining (9) and (12) we have a closed-loop system that describes the rank evolution with a dynamic resource allocation strategy

$$\begin{aligned} \dot{\mathbf{V}} &= \mathbf{f}(\mathbf{V}, \mathbf{z}), \\ \dot{\mathbf{r}} &= aJ_{\mathbf{r}}^{\top} \mathbf{z} J_{\mathbf{z}}^{\top} \mathbf{f} \nabla_{\dot{\mathbf{V}}} T. \end{aligned} \quad (13)$$

Strictly speaking, the feedback in (12) is not a gradient of the objective function  $T$ . It is nevertheless capable of continuously computing the instantaneous resource vector  $\mathbf{r}$  in the direction that  $T$  improves.

#### IV. ANALYSIS OF RESOURCE ALLOCATION

In this section we will show that choosing the particular feedback in the form of (12) most effectively improves the objective function  $T$ . As we will see, the system of differential equations in (2) not only facilitates design, but also analysis. The main theorem is the following

*Theorem 1:* By introducing the feedback (12) in (13),  $\dot{T} \geq 0$ . Among all possible feedback  $\dot{\mathbf{r}} = \mathbf{g}$  such that  $\|\mathbf{g}\| = \|aJ_{\mathbf{r}}^{\top} \mathbf{z} J_{\mathbf{z}}^{\top} \mathbf{f} \nabla_{\dot{\mathbf{V}}} T\|$ ,  $\dot{T}$  is maximized by (12).

*Proof:*  $\dot{T} \geq 0$  can be seen from the following computation

$$\begin{aligned} \dot{T} &= (\nabla_{\dot{\mathbf{V}}}T)^{\top} \dot{\mathbf{V}} = (\nabla_{\dot{\mathbf{V}}}T)^{\top} \dot{\mathbf{f}}(\mathbf{V}; \mathbf{z}) \\ &= (\nabla_{\dot{\mathbf{V}}}T)^{\top} \left( J_{\mathbf{V}}\mathbf{f}\dot{\mathbf{V}} + J_{\mathbf{z}}\mathbf{f}\dot{\mathbf{z}} \right) \\ &= (\nabla_{\dot{\mathbf{V}}}T)^{\top} J_{\mathbf{z}}\mathbf{f} J_{\mathbf{r}}\dot{\mathbf{z}} \quad (\text{due to (5)}) \\ &= (\nabla_{\dot{\mathbf{V}}}T)^{\top} aJ_{\mathbf{z}}\mathbf{f} J_{\mathbf{r}}\mathbf{z} J_{\mathbf{r}}^{\top} \mathbf{z} J_{\mathbf{z}}^{\top} \mathbf{f} \nabla_{\dot{\mathbf{V}}} T \\ &= a(\nabla_{\dot{\mathbf{V}}}T)^{\top} J_{\mathbf{z}}\mathbf{f} J_{\mathbf{r}}\mathbf{z} J_{\mathbf{r}}^{\top} \mathbf{z} J_{\mathbf{z}}^{\top} \mathbf{f} \nabla_{\dot{\mathbf{V}}} T \\ &= a\|(\nabla_{\dot{\mathbf{V}}}T)^{\top} J_{\mathbf{z}}\mathbf{f} J_{\mathbf{r}}\mathbf{z}\|^2 \geq 0. \end{aligned} \quad (14)$$

Given any  $\mathbf{g}$  such that  $\|\mathbf{g}\| = \|aJ_{\mathbf{r}}^{\top} \mathbf{z} J_{\mathbf{z}}^{\top} \mathbf{f} \nabla_{\dot{\mathbf{V}}} T\|$ , repeating the computation as shown in (14), we get

$$\dot{T} = a(\nabla_{\dot{\mathbf{V}}}T)^{\top} J_{\mathbf{z}}\mathbf{f} J_{\mathbf{r}}\mathbf{z}\mathbf{g}, \quad (15)$$

so

$$\begin{aligned} |\dot{T}| &\leq \|a(\nabla_{\dot{\mathbf{V}}}T)^{\top} J_{\mathbf{z}}\mathbf{f} J_{\mathbf{r}}\mathbf{z}\| \|\mathbf{g}\| \\ &= a\|(\nabla_{\dot{\mathbf{V}}}T)^{\top} J_{\mathbf{z}}\mathbf{f} J_{\mathbf{r}}\mathbf{z}\|^2. \end{aligned} \quad (16)$$

<sup>1</sup>The superscript  $\top$  stands for matrix transpose.

Note (16) achieves equality when the feedback is chosen as shown in (12). ■

Note Theorem 1 is true regardless whether  $T$  is concave or whether  $T$  is monotonic. It is also true regardless whether  $\mathbf{z} = \mathbf{z}(\mathbf{r})$  is concave or monotonic. Therefore it can be applied to a wide array of problems. Theorem 1 is also true when  $T$  is non-differentiable on a null set since what really matters is  $\int \dot{T} dt$ . As a result, when we have multiple interested destinations  $d_1, d_2, \dots \in \mathcal{D}$ , an objective of the form

$$T(\dot{\mathbf{V}}) = \min_{d \in \mathcal{D}} V_d \quad (17)$$

can still be maximized with the chosen feedback. With Theorem 1, we immediately have

*Corollary 1:* If  $\mathbf{r}$  is contained in a bounded set (i.e., finite resource allocation),  $\mathbf{z}(\mathbf{r})$  is continuous and  $T$  is continuous, with the feedback in (12),  $T$  converges.

*Proof:* We have

$$\dot{V}_{\mathcal{K}} = \sum_{i \in \mathcal{K}^c} z_{i, \mathcal{K}} (1 - q^{V_{\mathcal{K}} - V_{\mathcal{K} \cup \{i\}}}) \leq \sum_{i \in \mathcal{K}^c} z_{i, \mathcal{K}}. \quad (18)$$

Because  $\mathbf{r}$  is bounded, it is contained in a compact set. Therefore  $\mathbf{z}$  is contained in a compact set and from (18)  $\dot{\mathbf{V}}$  is contained in a compact set. Since  $T$  is continuous on this compact set,  $T$  is also bounded. This implies that, if  $T$  is monotonically increasing, it must converge. ■

From (14) we see that  $\dot{T}$  (the rate of convergence) can be calculated if an expression is known for  $J_{\mathbf{r}}(\mathbf{z})$ . In the absence of the exact knowledge of  $J_{\mathbf{r}}(\mathbf{z})$ , we also see the feedback gain  $a$  can be used to speed up convergence. However, the tradeoff is, with  $a$  too large we are at the risk of changing  $\mathbf{z}$  too fast thus invalidating (5). It remains to be answered how different the solution and optimality will be if we vary  $a$  without invalidating (5). The next theorem shows that the solutions for  $\mathbf{V}$  and  $\mathbf{r}$  are basically the same except for a proper scaling and the optimality is insensitive for a range of  $a$ . This is generally not true when dealing with other dynamical systems where different feedback gains lead to very different trajectories. Before we proceed to the theorem, it would be convenient to define  $\mathbf{f}^a$  to be the same with  $\mathbf{f}$ , except the field size  $q$  is replaced by  $q^a$ .

*Theorem 2:* Let  $\mathbf{V}^a, \mathbf{r}^a, \mathbf{z}^a$  satisfy (13) with an arbitrary  $a > 0$ . Let  $\mathbf{V}(t) = a\mathbf{V}^a(t/a)$ ,  $\mathbf{r}(t) = \mathbf{r}^a(t/a)$  and  $\mathbf{z}(t) = \mathbf{z}^a(t/a)$ . Then  $\mathbf{V}, \mathbf{r}$  and  $\mathbf{z}$  satisfy

$$\begin{aligned} \dot{\mathbf{V}}(t) &= \mathbf{f}^{1/a}(\mathbf{V}, \mathbf{r}), \\ \dot{\mathbf{r}}(t) &= J_{\mathbf{r}}^{\top} \mathbf{z} J_{\mathbf{z}}^{\top} \mathbf{f}^a \nabla_{\dot{\mathbf{V}}} T(\dot{\mathbf{V}}). \end{aligned} \quad (19)$$

*Proof:* For rank evolution, we have

$$\begin{aligned} \dot{\mathbf{V}}(t) &= a\dot{\mathbf{V}}^a(t/a)/a = \mathbf{f}(\mathbf{V}^a(t/a), \mathbf{z}^a(t/a)) \\ &= \mathbf{f} \left( \frac{1}{a} \mathbf{V}(t), \mathbf{z}(t) \right) = \mathbf{f}^{1/a}(\mathbf{V}(t), \mathbf{z}(t)). \end{aligned} \quad (20)$$

The last step follows from the fact

$$1 - q^{\frac{1}{a} V_{\mathcal{K}} - \frac{1}{a} V_{\mathcal{K} \cup \{i\}}} = 1 - \left( q^{1/a} \right)^{V_{\mathcal{K}} - V_{\mathcal{K} \cup \{i\}}}. \quad (21)$$

To verify the feedback equation in (19), we assume  $z, r, V$  are an arbitrary component of  $\mathbf{z}, \mathbf{r}, \mathbf{V}$ , respectively. Likewise, assume  $z^a, r^a, V^a$  are an arbitrary component of  $\mathbf{z}^a, \mathbf{r}^a, \mathbf{V}^a$ , respectively. Then we have

$$\frac{\partial z}{\partial r}(t) = \frac{\dot{z}}{\dot{r}}(t) = \frac{\dot{z}^a}{\dot{r}^a}(t/a) = \frac{\partial z^a}{\partial r^a}(t/a), \quad (22)$$

$$\frac{\partial f}{\partial z}(\mathbf{V}, \mathbf{z}) = \frac{\partial f}{\partial z}(a\mathbf{V}^a(t/a)) = \frac{\partial f^a}{\partial z^a}(t/a), \quad (\text{by (6)}) \quad (23)$$

$$\begin{aligned} \frac{\partial T(\dot{\mathbf{V}})}{\partial \dot{\mathbf{V}}} &= \lim_{\Delta t \rightarrow 0} \frac{T(\dot{\mathbf{V}}^a(t/a + \Delta t/a)) - T(\dot{\mathbf{V}}^a(t/a))}{\dot{\mathbf{V}}^a(t/a + \Delta t/a) - \dot{\mathbf{V}}^a(t/a)} \\ &= \frac{\partial T}{\partial \dot{\mathbf{V}}^a} \left( \frac{t}{a} \right). \end{aligned} \quad (24)$$

From (22) – (24) and

$$\dot{\mathbf{r}}^a(t) = a J_{\mathbf{r}^a}^\top \mathbf{z}^a J_{\mathbf{z}^a}^\top \mathbf{f} \nabla_{\dot{\mathbf{V}}^a} T(\mathbf{V}^a), \quad (25)$$

we know

$$\begin{aligned} \dot{\mathbf{r}}(t) &= \frac{1}{a} \dot{\mathbf{r}}^a \left( \frac{t}{a} \right) = J_{\mathbf{r}^a}^\top \mathbf{z}^a \left( \frac{t}{a} \right) J_{\mathbf{z}^a}^\top \mathbf{f} \left( \frac{t}{a} \right) \nabla_{\dot{\mathbf{V}}^a} T(\dot{\mathbf{V}}^a) \left( \frac{t}{a} \right) \\ &= J_{\mathbf{r}}^\top \mathbf{z}(t) J_{\mathbf{z}}^\top \mathbf{f}^a(t) \nabla_{\dot{\mathbf{V}}} T(\dot{\mathbf{V}})(t). \end{aligned} \quad (26)$$

In [6], a concentration result has been obtained regardless of  $q$  as long as  $q$  is sufficiently large. In practice, the effect of finite  $q$  on the concentration speed is minimal if  $q \geq 2$ , because in this case  $q^{-x} \rightarrow 0$  quickly as  $x > 0$  increases. Rank evolution is therefore insensitive to a wide range of  $q$ , with which it concentrates to the same solution. As a result, as long as  $q^a \gg 1$  and  $q^{1/a} \gg 1$ , different  $a$ 's in (13) yield the same system trajectory given by

$$\begin{aligned} \dot{\mathbf{V}} &= f(\mathbf{V}, \mathbf{r}), \\ \dot{\mathbf{r}} &= J_{\mathbf{r}}^\top \mathbf{z} J_{\mathbf{z}}^\top \mathbf{f} \nabla_{\dot{\mathbf{V}}} T(\dot{\mathbf{V}}), \end{aligned} \quad (27)$$

after proper scaling as described in Theorem 2. Hence we have

*Corollary 2:* If  $a$  is chosen such that  $\mathbf{f} \approx \mathbf{f}^a \approx \mathbf{f}^{1/a}$ ,  $T$  converges to the same value.

*Proof:* This is true because after scaling  $\mathbf{V}, \mathbf{z}, \mathbf{r}$  satisfy (27). But

$$\dot{\mathbf{V}}(t) = \dot{\mathbf{V}}^a(t/a). \quad (28)$$

Then

$$\lim_{t \rightarrow \infty} T(\dot{\mathbf{V}}^a(t)) = \lim_{t \rightarrow \infty} T(\dot{\mathbf{V}}^a(t/a)) = \lim_{t \rightarrow \infty} T(\dot{\mathbf{V}}(t)). \quad (29)$$

When choosing  $a$ , we want it to be large for fast convergence, but not so large that  $q^{1/a} \approx 1$  in (20). On the other hand, we want to avoid making  $a$  so small that  $q^a \approx 1$  in (23). Both extremes would undermine the concentration assumption. In general, concentration is essential for RNC to achieve the throughput, and it is also the key assumption for what we have discussed so far to remain valid. To do this, given the real field size  $q$  used, we may set a threshold  $q^{\text{th}} < q$  below which the concern for concentration will arise. Then pick  $a$  such that

$$q^a, q^{1/a} \geq q^{\text{th}}, \quad (30)$$

i.e., from the interval  $[\log q^{\text{th}} / \log q, \log q / \log q^{\text{th}}]$ .

If we wish to know how bad the algorithm would perform with  $a$  so large that (5) becomes invalid (but the concentration assumption still holds), Theorem 3 indicates that (13) may still work.

*Theorem 3:* Assume  $a$  is so large that (5) is not valid. If  $T$  is continuously differentiable in the first quadrant and  $\mathbf{r}$  is constrained in a bounded set,  $\dot{T} \geq 0$  until  $\|J_{\mathbf{r}}^\top \mathbf{z} J_{\mathbf{z}}^\top \mathbf{f}^a \nabla_{\dot{\mathbf{V}}} T(\dot{\mathbf{V}})\| < c/a$  for some constant  $c$  independent of  $a$ .

*Proof:* Since we do not have (5) any more, we must write

$$\begin{aligned} \dot{T} &= (\nabla_{\dot{\mathbf{V}}} T)^\top \dot{\mathbf{V}} \\ &= (\nabla_{\dot{\mathbf{V}}} T)^\top \left( J_{\mathbf{V}} \mathbf{f} \dot{\mathbf{V}} + J_{\mathbf{z}} \mathbf{f} \dot{\mathbf{z}} \right) \\ &= (\nabla_{\dot{\mathbf{V}}} T)^\top J_{\mathbf{V}} \mathbf{f} \dot{\mathbf{V}} + a \|(\nabla_{\dot{\mathbf{V}}} T)^\top J_{\mathbf{z}} \mathbf{f} J_{\mathbf{r}} \mathbf{z}\|^2. \end{aligned} \quad (31)$$

From (18) we already know  $\dot{\mathbf{V}}$  is contained in a compact set.  $\nabla_{\dot{\mathbf{V}}} T$  is thus bounded as  $T$  is continuously differentiable. Moreover, we can bound  $\|J_{\mathbf{V}} \mathbf{f}\|$ . To see this, note for arbitrary row  $\mathcal{K}$  of  $J_{\mathbf{V}} \mathbf{f}$ , all entries are zero except for column  $\mathcal{K}$  and column  $\mathcal{K} \cup \{i\}$  ( $i \in \mathcal{K}^c$ ), whose entries are

$$\frac{\partial \dot{V}_{\mathcal{K}}}{\partial V_{\mathcal{K}}} = -\log q \sum_{i \in \mathcal{K}^c} z_{i, \mathcal{K}} q^{V_{\mathcal{K}} - V_{\mathcal{K} \cup \{i\}}} \geq -\log q \sum_{i \in \mathcal{K}^c} z_{i, \mathcal{K}}; \quad (32)$$

$$\frac{\partial \dot{V}_{\mathcal{K}}}{\partial V_{\mathcal{K} \cup \{i\}}} = \log q z_{i, \mathcal{K}} q^{V_{\mathcal{K}} - V_{\mathcal{K} \cup \{i\}}} \leq \log q z_{i, \mathcal{K}}. \quad (33)$$

Therefore,  $J_{\mathbf{V}} \mathbf{f}$  is also bounded. Hence there exists a constant  $c$  independent of  $a$ , such that

$$\|(\nabla_{\dot{\mathbf{V}}} T)^\top J_{\mathbf{V}} \mathbf{f} \dot{\mathbf{V}}\| \leq c. \quad (34)$$

It is clear that, as long as  $a \|(\nabla_{\dot{\mathbf{V}}} T)^\top J_{\mathbf{z}} \mathbf{f} J_{\mathbf{r}} \mathbf{z}\|^2 \geq c$ ,  $\dot{T} \geq 0$ .

Note when  $\dot{T} < 0$ , it must be true that  $a \|(\nabla_{\dot{\mathbf{V}}} T)^\top J_{\mathbf{z}} \mathbf{f} J_{\mathbf{r}} \mathbf{z}\|^2 < c$ , which can be regarded as the first order stopping criteria. Although Theorem 3 makes the choice of  $a$  even more liberal, it is nevertheless advisable to choose  $a$  reasonably small, otherwise additional nonlinearities may start to enter the picture. This is especially true when we move from the continuous system (13) to a discretized system that adjusts resource allocation only at time instants it is triggered.

## V. NUMERICAL EXAMPLE

In this section, we use the network coding aware power control as an example to demonstrate the performance of the gradient based algorithm as shown in (13). Consider the 6-node network shown in Fig. IV where nodes perform RNC on GF(2) as described in Section II. Node 1 attempts to multicast to nodes in  $\mathcal{D} = \{2, 3, 4, 5, 6\}$  and we wish to maximize the minimum throughput given in (17). We assume MAC has been fixed so that node  $i$  ( $i = 1, \dots, 6$ ) transmits at  $\lambda_i = 1$  packet per millisecond. However, the transmit power of each node can be flexibly adjusted from 0 to 15dBm. The initial transmit powers are set uniformly to 13dBm. We assume the interference is prevalent. Let  $P_i^{\text{Tx}}$  denote the transmit powers

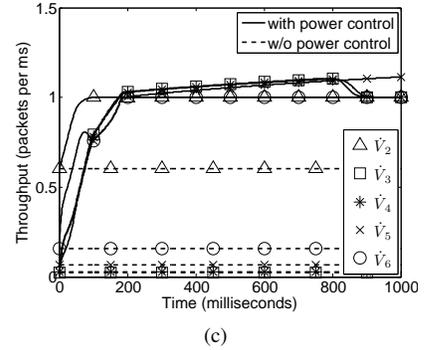
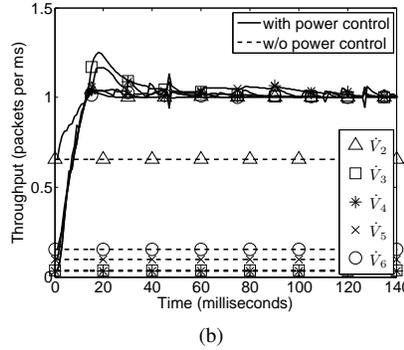
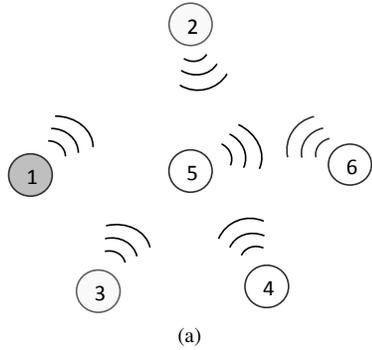


Fig. 1. (a) A 6-node wireless network; (b)  $a = 10$ ; (c)  $a = 0.1$ .

at node  $i$ ,  $h_{ji}$  denote the link gains from node  $i$  to node  $j$ ,  $\sigma^2 = -101\text{dBm}$  denote the thermal noise power at every receiver. We also associate with each node  $i$  a processing gain  $g_i = 8$ . Consequently, we can model the signal-to-interference-noise ratio (SINR) at node  $j$  when  $j$  attempts to decode the transmission from  $i$  as

$$\text{SINR}_{ji} = \frac{P_i^{\text{Tx}} h_{ji}}{\sum_{k \neq i, j} P_k^{\text{Tx}} h_{jk} / g_i + \sigma^2}. \quad (35)$$

For the link gains, we use the ITU indoor attenuation model [9] based on distance. The path loss PL over distance  $d$  is given by

$$\text{PL} = 20 \log f_c + 10n \log d + P_f(n) - 28 \quad (36)$$

where  $f_c$  is the center frequency,  $n$  the path loss exponent and  $P_f(n)$  the floor penetration factor. Consequently the link gain is given by  $h = 10^{\text{PL}/10}$ . In the simulation, we choose  $f_c = 2.4\text{GHz}$ ,  $n = 3$ ,  $P_f(n) = 11$ . We use uncoded BPSK modulation with a codeword/packet length of  $L = 160$  bits. Assuming a Gaussian distribution for the interference, the bit error rate for  $j$  decoding  $i$  is given by

$$P_{ji}^{\text{bit}} = Q\left(\sqrt{\text{SINR}_{ji}}\right). \quad (37)$$

The packet error rate is given by  $P_{ji} = 1 - (1 - P_{ji}^{\text{bit}})^L$ . Assuming independent reception, we thus have

$$P_{i, \mathcal{K}} = 1 - \prod_{j \in \mathcal{K}} P_{ji} \quad \text{and} \quad z_{i, \mathcal{K}} = \lambda_i P_{i, \mathcal{K}}. \quad (38)$$

Using the model specified by (35) – (38), we execute (13) to demonstrate our power control algorithm. Fig. 1b – 1c show the throughputs as functions of time with  $a = 10$ , and  $a = 0.1$ , respectively, in addition to throughputs achieved without power control. With  $a = 10$ , the minimum throughput  $T$  (the lower envelop of all the curves) increases quickly. But as  $\mathbf{z}$  changes too fast, (5) becomes invalid and leads to oscillation of the trajectory of  $T$ . Lowering  $a$  results in a smoother trajectory with slower convergence. However, in both settings,  $T$  eventually achieves the the same value of 1 packet per millisecond. Considering the transmit rate of node 1 (the source) is also 1 packet per millisecond, the gradient based resource allocation has achieved the maximum possible

objective without giving up any multicast advantage and also showed significant improvement over no power control.

## VI. CONCLUSION

In this article we presented a new resource allocation algorithm specifically designed for RNC and throughput based objectives. Using the dynamical system model of RNC, the algorithm evolves by introducing a gradient based state feedback. We proved that this feedback adaptively computes the allocation that continuously improves the objective. We also discussed convergence of the algorithm as well as design considerations for the feedback gain. We illustrated the use of the resource allocation methodology by designing power control for maximizing minimum throughput in a multicast.

## ACKNOWLEDGMENT

This work is supported in part by the NSF under grant no. CCF-1016551.

## REFERENCES

- [1] T. Ho and H. Viswanathan, “Dynamic algorithms for multicast with intra-session network coding,” *IEEE Trans. Inform. Theory*, vol. 55, no. 2, pp. 797–815, Feb. 2009.
- [2] D. Zhang and N. Mandayam, “Resource allocation for multicast in an OFDMA network with random network coding,” in *INFOCOM*, Shanghai, China, Apr. 2011, pp. 391–395.
- [3] D. P. Bertsekas, *Network Optimization: Continuous and Discrete Models*. Belmont, Massachusetts: Athena Scientific, 1998.
- [4] T. Tran, D. Nguyen, T. Nguyen, and D. Tran, “Joint network coding and power control for cellular radio networks,” in *Second International Conference on Communications and Electronics*, Jun. 2008, pp. 109–114.
- [5] T. Tran, D. Nguyen, and T. Nguyen, “A case for joint network coding and power control in wireless linear networks,” in *6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks Workshops*, Jun. 2009, pp. 1–6.
- [6] D. Zhang and N. Mandayam, “Analyzing random network coding with differential equations and differential inclusions,” *IEEE Trans. Inform. Theory*, vol. 57, no. 12, pp. 7932–7949, December 2012.
- [7] T. Ho, M. Médard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, “A random linear network coding approach to multicast,” *IEEE Trans. Inform. Theory*, vol. 52, no. 10, pp. 4413 – 4430, October 2006.
- [8] D. S. Lun, M. Medard, R. Koetter, and M. Effros, “On coding for reliable communication over packet networks,” in *42nd Annual Allerton Conference on Communication, Control, and Computing*, Sept. 2004.
- [9] “Propagation data and prediction methods for the planning of indoor radio communication systems and the radio local area networks in the frequency range 900MHz to 100GHz,” ITU-R Recommendations, 2001.