

Exploiting Route Diversity in Multi-packet Transmission using Mutual Information Accumulation

Yanpei Liu, Jing Yang and Stark C. Draper

Electrical and Computer Engineering, University of Wisconsin Madison

Email: {yliu73@, yangjing@ece, sdraper@ece}@wisc.edu

Abstract—We consider cooperative communication in a given network using rateless codes. Given a network, a source and a destination, we investigate routing and resource allocation for delivering multiple packets through the network. We propose and design a centralized algorithm which minimizes a given objective under different network resource constraints. The solution to the algorithm is a policy that determines which nodes should participate in transmitting which packets, when and with how much resource. Our algorithm efficiently exploits route diversity for packets transmission in both spatial and temporal aspects. Packets may take different routes to reach the destination to exploit the spatial route diversity or they may reuse the same route, one followed by another, to exploit the temporal route diversity. The algorithm we develop also supports transmitting packets with different arrival times.

Index Terms—Wireless relay networks, cooperative communication, mutual information accumulation, rateless codes.

I. INTRODUCTION

In multi-hop cooperative relay networks, relay nodes cooperate with each other to deliver packets from one location to another. The benefits include, but not limited to, energy efficiency, robustness to fading and interference, exploiting other possible routes, and reducing the probability of loss of connectivity [1], [2]. Cooperative communication can be realized by allowing nodes to overhear the transmission from others and forward received message. We therefore design wireless nodes by equipping them with the following functions. First, nodes can overhear the transmission, i.e., neighbor nodes can choose to overhear the transmission originated from other nodes. Second, although the overheard information is noisy, nodes retain whatever they hear and accumulate independent samples of the information until they can successfully determine what the message is. We term the second function as *mutual information accumulation*.

Most of the current routing papers in the literature are based on energy accumulation by using space-time or repetition coding [3]–[5]. The difference between energy accumulation and mutual information accumulation can be easily understood from the following example [1]. Consider binary signaling over a pair of independent erasure channels each having erasure probability p_e from two relays to a single receiver. If the two relays use repetition coding, corresponding to energy accumulation, then each symbol will be erased with probability p_e^2 . Therefore, $1 - p_e^2$ symbols are successfully received on average per transmission. Instead, if we use different codes,

the transmissions are independent and on average $2(1 - p_e)$ novel parity symbols are received per transmission. Mutual information accumulation can be realized through the use of rateless (or “fountain”) codes [1], [6]–[10].

It is important to consider multiple packets going through a given network. With only a single packet transmission, as in [1], [11], the question to be answered is merely which nodes should participate in transmission, when they should participate, and with how much network resource they participate. In multiple packets transmission, the extra factor need to be considered is which packets should a node transmit. This is important. As we show in this paper, by using our algorithm designed for multiple packets transmission, we can exploit the route diversity of the network in both spatial and temporal aspects. Different packets may take different routes to exploit the spatial route diversity. This is a desired feature when packets compete for resource; it may be desirable to route a group of packets through one link and the rest through another to avoid choking the already busy route. To exploit the temporal diversity, packets can be routed on a single route, one followed by another. This usually happens when routes choices are limited or packets arrive at different times. Essentially, our algorithm exploits available resource within the network and distributes it efficiently.

Since our algorithm exploits the spatial route diversity, one may want to packetize a big data chunk into several smaller packets and route these packets simultaneously. As we show in this paper, with per wireless node resource constraint, packetizing usually improves the performance as it fully exploits the resource within the network. Our algorithm also supports the case when packets arrive at the source at different times. One interesting observation is that if the inter-arrival time between two distinct packets is large, then one may want to exploit the temporal route diversity by sending these two packets through the same link one after another, repeating the same routing and resource allocation policy twice.

The main contribution of our work is the design of a joint routing and resource allocation algorithm for multiple packets transmission from the source to the destination in a given network. Our work is a natural extension of the work in [1] where only single packet transmission is considered. The contribution of our work is twofold.

- We first present a formulation of the routing and resource allocation problem. The formulation considers various forms of energy and bandwidth constraints. The problem

formulation is expressed in the form of linear program (LP)

- Second, we design a centralized algorithm which solves the routing and resource allocation *jointly* by solving a sequence of LPs. Each LP solves for the optimal resource allocation given a route decision of all packets. The resource allocation result is then used to update the route decision and the method proceeds iteratively.

There are many existing research on cooperative communication. The authors in [2] design and implement ExOR, a packet-level opportunistic multi-hop routing for wireless networks. It exploits the broadcast nature of wireless communication by allowing sending information through multiple relays concurrently. Network coding approach is another distinct but related perspective. For example, a symbol-level network coding scheme is used in [12]. The relay nodes are designed not only to forward the correct packets, but also to forward parts of corrupted packets which they believe to be correct with the hint from the physical layer. The authors in [13] also design a practical network coding scheme called COPE which mixes packets from different sources to increase the performance. Their scheme incorporates opportunistic listening and coding. A DIVBAR algorithm is developed in [14] which exploits the broadcast nature of wireless transmission and maximizes the throughput in a cooperative manner. The algorithm is based on backpressure algorithm first proposed in [15], [16]. Other works such as [17]–[19] focus on relay nodes selection and assignment. In particular, [19] proposes a proactive link state routing protocol which improves the forward nodes selection in [2].

There are also related works in information accumulation literature. In [6], mutual information is considered for single relay networks. In [20], mutual information is also considered, but without a consideration on resource allocation. Routing and resource allocation with mutual information accumulation is considered in [1], [11], but they only consider single packet routing. To the best of our knowledge, there has been little prior work investigating routing and resource allocation with mutual information accumulation for multiple packets.

The rest of the paper is organized as follows. We present the system model in Sec. II. Problem formulation and algorithm are developed in Sec. III. We provide detailed numerical results in Sec. IV and conclude in Sec. V.

II. SYSTEM MODEL

We consider a system model with L nodes: the source, always labeled 1, the destination, always labeled L and $L - 2$ relay nodes. Suppose K large data chunks arrive at the source at deterministic times and we assume each data chunk can be packetized into packets. Packetizing is widely used in current wireless protocols such as [21]. Since data chunks arrive at the source at different times, the packetized packets are made available to the source at different times as well. This manifests the *arrival process* of packets. The network’s objective is to deliver N such packets, each composed of B bits, from the source to the destination. Each relay may participate in

transmitting a subset of these packets or remain silent. To simplify the analysis we assume that the only significant power expenditure for each node lies in transmission.

The i th node operates at a fixed power spectral density (PSD) denoted as P_i (joules/sec/Hz), uniform across its transmission band. We assume the channel between any pair of nodes is block-fading and frequency non-selective. The channel gain between node i and node j is denoted as $h_{i,j}$. Under these assumptions, the spectral efficiency (bits/sec/Hz) between node i and node j can be expressed as [1], [22]

$$C_{i,j} = \log_2 \left[1 + \frac{h_{i,j} P_i W_i}{N_0 W_i} \right] = \log_2 \left[1 + \frac{h_{i,j} P_i}{N_0} \right], \quad (1)$$

where $N_0/2$ denotes the PSD of the white noise process.

We further denote the time-bandwidth product allocated to node i to transmit packet $c \in \{1, 2, \dots, N\}$ during a given time period s as $A_{i,s}^c$ (sec-Hz). Then the information flow from node i to node j during this interval s is $A_{i,s}^c C_{i,j}$ bits. We first assume that nodes use codes that are ideal in the sense that they fully capture this potential flow, working at the Shannon limit at any rate. We also assume that distinct packets can be simultaneously aired from any node with no interference between packets and receivers have perfect knowledge to distinguish them. Nodes are further designed to use *independently* generated codes. This design leads to our third assumption that a receiver can combine information flows from two or more transmitters for each packet without any rate loss. If, for example, transmitting nodes i and j are allocated time-bandwidth product $A_{i,s}^c$ and $A_{j,s}^c$ respectively during period s , our assumptions mean that node k can decode as long as the mutual information accumulated by it exceeds the size of packet c , i.e.,

$$A_{i,s}^c C_{i,k} + A_{j,s}^c C_{j,k} \geq B. \quad (2)$$

Although this example considers only two nodes, a receiver in general will combine information from all available transmitting sources. It can decode as long as the total aggregated mutual information exceeds B bits [20].

As noted in [1], the use of independently-generated codes is crucial for the mutual information accumulation process reflected in (2) to work. If the transmitters used the *same* code, the receiver would get multiple looks at each codeword symbol. This is “energy accumulation.” By looking at different codes (generated from the same B information bits) the receiver accumulates mutual information rather than energy.

The idea of mutual information accumulation from multiple sources can be naturally realized by the use of rateless (or “fountain”) codes [8]–[10]. Fountain codes encode information bits into potentially infinitely long codewords; additional parity symbols are sent until the receiver is able to decode. For the performance of rateless codes, see [20].

III. CENTRALIZED ALGORITHM

We now consider an arbitrary network with L nodes randomly scattered and N packets of same size packetized from K large data chunks to be delivered from the source to the

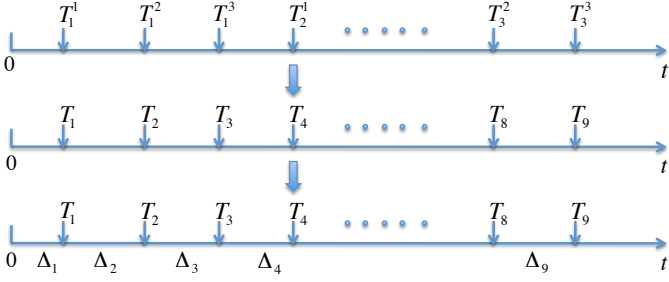


Fig. 1. A sample decoding order for $N = 3$ and $L = 3$.

destination. For a given network with L nodes, we denote the source as 1 and the destination as L . Our algorithm is first led by the definition of “decoding event” and “decoding order”.

Definition 1. A decoding event is either the event that a node decodes a packet, or a packet is made available at the source.

Definition 2. A decoding order is an ordered sequence of timing points at which decoding events occur.

We assume a decode-and-forward relaying strategy. We denote the timing point at which the event of node i decoding packet c occurs as T_i^c . In particular, T_1^c denotes the time at which packet c is made available to the source after sub-packetizing the data chunks. A typical decoding order consists of an ordered sequence of T_i^c 's as shown in Fig. 1. Since a node cannot transmit a packet until it has decoded it, the positions of the timing points put constraints on the resource allocated to nodes for each packet. In particular, the positions of T_1^c , $c \in \{1, 2, \dots, N\}$ must reflect the arrival process of packets induced by the arrival process of the data chunks. We design and implement a centralized resource allocation and routing algorithm iterating between two sub-problems:

- 1) For a given decoding order, a resource allocation scheme is determined. The resource allocation problem can be formulated as a linear programming (LP) problem.
- 2) With a given resource allocation scheme, we update the decoding order.

In the following subsections we first establish and parameterize the problem and analyze the characteristics of the parameterized problem. Then we present the discussion on optimizing the decoding order, a key component in our algorithm which is presented in the last subsection.

A. Problem Parameterization

We note that a decoding order must contain two sets of timing points, i.e., T_1^c and T_L^c for all $c \in \{1, 2, \dots, N\}$. We also note that a decoding order can contain up to NL timing points. Let M to denote the total number of timing points included in a given decoding order. Then the label pair (i, c) of a timing point T_i^c uniquely determines its position s , $s \in \{1, 2, \dots, M\}$ and we denote this mapping as $T_i^c \equiv T_{f(i,c)} \equiv T_s$. A decoding order sample is shown in Fig. 1.

However, instead of working directly with T_s , we find it more convenient to work with “inter-decoding delays”. We present the definition of inter-decoding delays as follows

Definition 3. The inter-decoding delay Δ_s is the time interval between T_s and T_{s-1} and we denote time 0 as T_0 .

Fig. 1 shows the inter-decoding delays. The entire message transmission can be thought of as consisting M phases. The s th phase is of duration Δ_s and is characterized by the fact that at the end of this Δ_s , a decoding event associated with T_s must occur. We let $A_{i,s}^c$ denote the time-bandwidth product allocated to node i to transmit packet c during Δ_s .

For a given decoding order we find the resource allocation minimizing an objective function subject to the following constraints:

- 1) $\Delta_s \geq 0$ for all s .
- 2) $A_{i,s}^c \geq 0$ for all s, i and c .
- 3) Arrival process constraint; the source must have a packet c ready to transmit according to a specified arrival process.
- 4) Decoding constraint; node j must decode packet c at the associated timing point T_s .
- 5) Constraint(s) on energy and bandwidth.

Arrival process constraint puts a constraint on the time intervals between T_1^c 's. Packets are made available by sub-packetizing big data chunks. Thus the arrival process of big data chunks puts the constraint on the availability of data packets at the source; the source must have a packet c ready to transmit at time T_1^c .

Decoding constraint imposes a constraint on other timing points; if T_j^c , $j \neq 1$ is in the decoding order, node j must be able to decode packet c at this time. This constraint can be formally expressed as

$$\sum_{i: f(i,c) < f(j,c)} \sum_{s=f(i,c)+1}^{f(j,c)} A_{i,s}^c C_{i,j} \geq B, \text{ for all } j \neq 1, c. \quad (3)$$

Recall that $C_{i,j}$ is the spectral efficiency (bits/sec/Hz) of the channel connecting node i to node j . Equation (3) says that in order for node j to decode packet c , the total accumulated information at node j for packet c must exceed B bits by the time T_j^c . A node i can contribute to node j only if it has decoded packet c before and can contribute starting from T_i^c to T_j^c .

Constraint(s) on energy and bandwidth can be either sum form which imposes constraints on the system-wide overall resource expenditure, or per-node form which regulates the resource expenditure on each node. We state these constraints in the followings.

- 1) *Per-node bandwidth constraint:* If node i is assigned with bandwidth W_i , its resource allocated during given Δ_s must satisfy the following

$$\sum_{c=1}^N A_{i,s}^c \leq \Delta_s W_i, \text{ for all } i, s. \quad (4)$$

2) *Sum-bandwidth constraint*: If the total bandwidth W_T is allocated, a sum-bandwidth constraint apply across all nodes during any given Δ_s . This can be formally expressed as

$$\sum_{i=1}^L \sum_{c=1}^N A_{i,s}^c \leq \Delta_s W_T, \text{ for all } s. \quad (5)$$

3) *Per-node energy constraint*: If node i with transmission power P_i is assigned with energy budget E_i , we can express per-node energy constraint as

$$\sum_{c=1}^N \sum_{s=f(i,c)+1}^M A_{i,s}^c P_i \leq E_i, \text{ for all } i. \quad (6)$$

4) *Sum-energy constraint*: If a sum energy E_T is assigned for all nodes, a sum-energy constraint can be applied as

$$\sum_{i=1}^L \sum_{c=1}^N \sum_{s=f(i,c)+1}^M A_{i,s}^c P_i \leq E_T. \quad (7)$$

The LP framework can take many objective functions. If there is only one data chunk (thus all packetized packets are available to the source at the same time), one natural objective function to use is the total transmission time

$$T_t = \sum_{s=1}^M \Delta_s. \quad (8)$$

Alternatively, if K data chunks arrive at different times (thus packets are not available to the source at the same time), instead of minimizing total transmission time, one might want to minimize the average transmission time given by

$$T_a = \frac{1}{K} \sum_{\ell=1}^K T_\ell, \quad (9)$$

where T_ℓ is the time period from the time data chunk ℓ arrives at the source to the time destination decodes it. This objective function is a good gauge in the sense that it does not penalize the arrival waiting time.

Other linear programming frameworks are also possible. For example, one may wish to minimize the total energy expenditure given by

$$E_{exp} = \sum_{i=1}^L \sum_{c=1}^N \sum_{s=f(i,c)+1}^M A_{i,s}^c P_i, \quad (10)$$

subject to total transmission time or average transmission time constraint.

Solving the optimization problem with these constraints gives time allocation Δ_s for all s and the resource allocation $A_{i,s}^c$ for all i , c and s .

B. Characteristics of the problem

We now study the properties of routing and resource allocation under different constraints with total transmission time (8) as our objective function. First, under per-node bandwidth constraint, we have the following lemma.

Lemma 1. *Under a given decoding order, suppose Δ_s and $A_{i,s}^c$ are the solution to the problem under per-node bandwidth constraint (4). Then for each s , there exists a node i such that the inequality in (4) becomes equality.*

Proof: We prove by contradiction, suppose for some s inequality is strict for all i , i.e.,

$$\sum_{c=1}^N A_{i,s}^c < \Delta_s W_i, \text{ for all } i.$$

Then we can scale down Δ_s , yielding a smaller objective value. This contradicts the assumption that Δ_s is the optimal solution to the LP. ■

This proposition suggests that the Δ_s can be calculated as

$$\Delta_s = \max_i \frac{\sum_{c=1}^N A_{i,s}^c}{W_i}.$$

Under sum bandwidth constraint, we first have the following lemma.

Lemma 2. *Under a given decoding order, suppose Δ_s and $A_{i,s}^c$ are the solution to the problem under sum bandwidth constraint (5). Then the inequality in (5) must be equality for all s .*

Proof: We prove this by contradiction. Suppose for some s the equality does not hold, i.e.,

$$\sum_{i=1}^L \sum_{c=1}^N A_{i,s}^c = \Delta_s W_s < \Delta_s W_T, \text{ for some } s.$$

Then for these s we can scale down the corresponding Δ_s by $\frac{W_s}{W_T}$ while increase W_s to W_T . We therefore obtain a solution which has smaller objective value. ■

Theorem 3. *Under sum bandwidth constraint and a given decoding order, if $P_i = P$ for all i then the solution that minimizes the objective in (8) also minimizes the sum energy.*

Proof: The total energy expenditure E_{exp} of the entire system is

$$\begin{aligned} E_{exp} &= \sum_{i=1}^L \sum_{c=1}^N \sum_{s=f(i,c)+1}^M A_{i,s}^c P \\ &= \sum_{i=1}^L \sum_{c=1}^N \sum_{s=1}^M A_{i,s}^c P \\ &= \sum_{s=1}^M \Delta_s W_T P \\ &= T_t W_T P. \end{aligned}$$

The first equality holds because a node cannot transmit a given packet (resource allocated to it is zero) before it decodes the packet. The second equality follows from Lemma. 2. Since the objective T_t is proportional to the energy used E_{exp} , minimizing one minimizes the other. ■

Theorem 4. *Under sum bandwidth constraint and a given decoding order, if the minimum transmission time for transmitting one packet is T , then the minimum transmission time for routing N packets of same size is NT .*

Proof: Let \mathcal{R} denote the optimal set of routes on which transmitting a packet takes T amount of time. These routes are equivalent in the sense that transmitting a packet on any of them takes T amount of time, consuming TW_T a amount of total resource. Then transmitting N packets of the same size on \mathcal{R} can be finished within NT amount of time by allocating each packet $\frac{TW_T}{N}$ amount of total resource.

We now show that this is the best the system can do. Suppose otherwise, i.e., not all packets are routed through \mathcal{R} and the total transmission time is smaller than NT . Then there exists at least one packet traveling on a different set of equivalent routes $\mathcal{R}' \neq \mathcal{R}$. Since the transmission time is smaller than NT , each packets traveling on \mathcal{R} should be given more than $\frac{TW_T}{N}$ amount of resource. This implies that the packet on \mathcal{R}' shares less than $\frac{TW_T}{N}$ amount of resource whereas achieves a total transmission time less than NT . This contradicts our assumption that \mathcal{R} is the optimal set of equivalent routes. ■

C. Optimizing the Decoding Order

In this section we show an important theorem that motivates a heuristic algorithm in later section. The theorem tells us how to manipulate the decoding order based on the solution of the LP problem. Given any decoding order of length M , define

$$\mathbf{x} = \left[\Delta_1, \dots, \Delta_M, A_{1,1}^1, \dots, A_{L,M}^N \right]$$

to be the solution to the LP with optimal objective value T_{opt} on this decoding order. We then have the following theorem.

Theorem 5. *If $\Delta_m = 0$ for some m and we swap the positions of T_m and T_{m-1} , then the objective value T_{opt}^* obtained with this swapped decoding order satisfies $T_{opt}^* \leq T_{opt}$.*

Proof: We prove the theorem by showing that when swapped decoding order is used, the original solution \mathbf{x} with optimal objective value T_{opt} is still feasible under new decoding order. To show \mathbf{x} is feasible under the new decoding order, we only need to show the decoding constraint of the swapped decoding order is satisfied by this solution.

If a decoding event associated with T_{m-1} has its decoding constraint satisfied at this time, its decoding constraint is certainly satisfied at a later time T_m . If a decoding event associated with T_m has its decoding constraint satisfied at this time, it actually has the constraint satisfied at an earlier time T_{m-1} since $A_{i,m}^c = 0$ for all i and c . Therefore, solution \mathbf{x}

```

Initialize a DO;
do_again = true

WHILE(do_again)
  result = LP(DO)
  index = search_for_Delta_which_are_zero(result)
  search_for_nodes_decode_after_destination(index)
  IF (find one)
    do_again = true
    drop the decoding event
  ELSEIF (fail AND index == index in previous
    iteration AND no event is dropped in
    previous iteration)
    do_again = false
    break
  ELSE
    do_again = true
    swap_events(result)
  END IF
END WHILE

```

Fig. 2. The algorithm for route selection and resource allocation.

with optimal objective value T_{opt} is feasible under swapped decoding order ■

The idea behind Theorem 5 is based on the following observation. A solution to the LP with $\Delta_m = 0$ indicates that the event associated with T_m takes place at exactly the same time with the previous event T_{m-1} , or actually occurs before it. Therefore, swapping the position of T_m and T_{m-1} typically gives a decrease in the objective value once the LP is solved under swapped decoding order. In the case that T_i^c is swapped with T_L^c for i not equal to L , the timing point T_i^c and its associated decoding event is excluded from the new decoding order by our algorithm.

D. Algorithm

We now state the algorithm to find the routing and resource allocation scheme. Our algorithm iterates between the following two subproblems.

- 1) For a given decoding order, solve the LP problem.
- 2) Based on the solution to the LP problem, update the decoding order.

We present the sketch of the algorithm in Fig. 2. In the following subsections we discuss various aspects of the algorithm in more details.

1) *Initialization:* We initialize the algorithm by using the following decoding order

$$DO = \left[T_1^1, T_1^2, \dots, T_1^N, T_2^1, \dots, T_L^N \right].$$

Since we do not know the preference of each node on every packet, including all NL timing points is a reasonable choice. Other initial orders are also possible. For example, one may randomly reorder the timing points T_i^c for all $i \neq 1, L$.

2) *Decoding order updates*: Based on the solution to the LP, the algorithm first searches for all those m 's such that $\Delta_m = 0$. Then for those m , it checks for the situation where T_i^c is followed by T_L^c for some $i \neq L$. If it finds one, it drops the corresponding T_i^c and restarts the LP. If it does not find one, it swaps the timing events T_{m-1} and T_m . If the swapped events are all the same as the ones in the previous iteration, and if there was no timing point dropped in the previous iteration, the algorithm terminates.

Because of the exponential number of orderings we expect the problem of finding the optimal decoding order to be NP-hard. The sub-optimality of our heuristic algorithm comes from the fact that the excluded decoding events may actually be helpful. Without a mechanism to “re-introduce” the excluded events our algorithm is not expected to achieve global minimum for all networks.

3) *Characteristics of final route*: Our algorithm has the property that different packets may take different paths toward the destination under per-node resource constraint. This exploits the spatial route diversity. For example, the algorithm may schedule the first 3 packets to pass through a set of relay nodes and schedule the next 3 packets to pass through yet another set of relay nodes. The intuition is that instead of choking an existing route and overloading the already busy nodes, the algorithm balances the traffic by invoking other free nodes to fully utilize the network.

Therefore, to fully utilize the resource, it is always wise to packetize a big data chunk into several smaller packets and route them through the network. By doing so, we fully exploit the free resource within the network. As we will see in the simulation section, in transmitting a single data chunk, dividing it into smaller pieces usually decreases the end-to-end total transmission time.

IV. NUMERICAL RESULTS

In this section we present detailed simulation results for the algorithm developed on different networks. These results exemplify the basic properties of using mutual information accumulation in cooperative communication.

A. A simple network

To better illustrate how our algorithm exploits the spatial route diversity, we first consider a simple four nodes example in which a data chunk of size 20 is transmitted in the form of two packets of size 10. Consider the diamond network in Fig. 3. Links with nonzero spectral efficiency (bits/sec/Hz) are shown. Each node is allocated with 1 unit of bandwidth and the source (labeled as 1) transmits these two packets to the destination (labeled as 4).

Running our algorithm returns the following routing and resource allocation scheme. Starting from time 0, node 1 broadcasts packet 1 for 5 units of time. By the end of time 5, node 2 decodes packet 1 and node 3 accumulates half of packet 1. Starting from time 5, node 2 broadcasts packet 1 to node 4 while node 1 broadcasts packet 2. By the end of time 15, node 4 decodes packet 1 and node 3 decodes packet

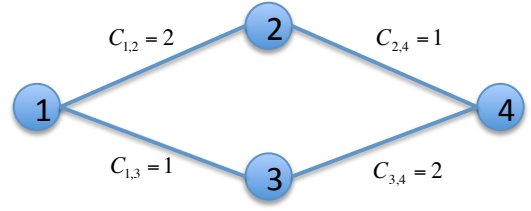


Fig. 3. A diamond network example. Note that transmitting a single packet on route 1 – 3 – 4 or 1 – 2 – 4 takes the same amount of time.

2. Note that by the end of time 10, node 2 decodes packet 2. Then starting from time 15, node 2 and 3 transmit packet 2 to node 4 at the same time and node 4 decodes packet 2 after 3.33 units of time. The total transmission time is thus $5 + 10 + 3.33 = 18.33$ units of time. In this example, node 3 never decode packet 1 thus packet 1 does not pass through route 1 – 3 – 4. Whereas node 3 decodes packet 2 and it cooperates with node 2 to deliver packet 2 to destination.

It is interesting to see that if two packets were routed through the same path, the total transmission time would be $2 \times 15 = 30$. If the data chunk itself is routed without packetizing, the total transmission time would be 30 no matter which route (1 – 3 – 4 or 1 – 2 – 4) it takes. This suggests that possible improvement can be achieved if we divide data chunks into smaller packets. We further illustrate this property in the following section.

B. Simulation on general networks

We now study the performance of our algorithm on more general networks. For better illustration, we choose bandwidth as our single resource element being allocated. Our simulation can take other resource elements such as energy or both energy and bandwidth.

We simulate the performance of our algorithm on 10 randomly scattered wireless nodes within a unit square. The source, denoted as 1, is always located at $[0.2, 0.2]$ and the destination, denoted as L , is always located at $[0.8, 0.8]$. The remaining 8 nodes are randomly placed according to the uniform distribution in the unit square. The channel gain $h_{i,j}$ is deterministically related to the Euclidean distance $d_{i,j}$ between node i and j as $h_{i,j} = d_{i,j}^{-2}$. The spectral efficiency between node i and j is given by (1).

1) *Per-node bandwidth constraint*: We first consider the system with per-node bandwidth constraint. We consider a data chunk of size 20 which is packetized into N packets so that $B = 20/N$. We also let $P_i = P = 1$, $W_i = 1$ and $N_0/2 = 1$. We simulate networks to estimate the cumulative distribution function (CDF) of the total transmission time under $N = 1, 2$ and 3. To further illustrate the benefits of using mutual information accumulation, we also plot the simulation of the transmission performance on the route obtained using Dijkstra’s algorithm [23] with and without mutual information accumulation in Fig. 4.

We see that dividing a data chunk into packets improves the performance by reducing the total transmission time. If

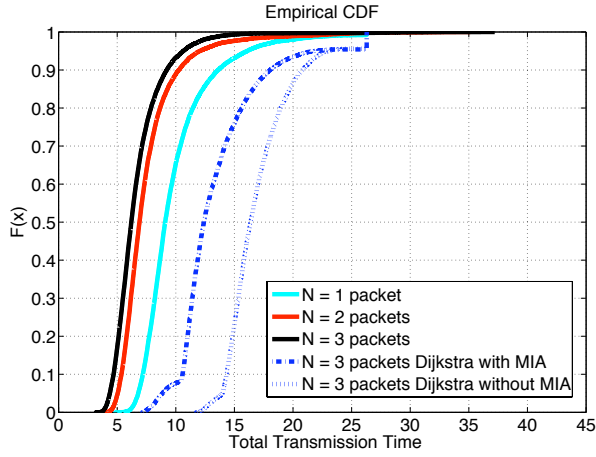


Fig. 4. Cumulative distribution of total transmission time under $N = 1, 2,$ and 3 .

we keep dividing it into more and more packets, we expect that the performance eventually converge to the one with the fluid model where the data chunk itself is treated as fluid. Comparing with Dijkstra’s algorithm, we also notice that the decrease in total transmission time comes from two causes; the use of mutual information accumulation and the ability to exploit other available route and resource within a network, not merely confined in a pre-determined Dijkstra’s route. This illustrates the how our algorithm exploits the spatial route diversity.

2) *Per-node bandwidth constraint with arrival process:* We now consider a deterministic arrival process under per-node bandwidth constraint. Consider a randomly generated network shown in Fig. 5. We now route $K = 3$ data chunks, each is of size 20, from the source to the destination. We also let $P_i = P = 1$, $W_i = 1$ and $N_0/2 = 1$. Instead of using total transmission time as our objective, we try to minimize the average transmission time given in (9). We simulate the average transmission time required when inter-arrival time between data chunks varies from 0 to 15.

For this given network, transmitting a single data chunk of size 20 requires 7.1853 units of time. If the inter-arrival time is larger than this number, transmitting 3 arriving data chunks means repeating the single chunk routing and resource allocation policy three times. Thus the average transmission time is the same as transmission time for single data chunk. This manifests the use of temporal route diversity. When inter-arrival time is small, chunks compete with each other for resource, thus the average transmission time increases as shown in Fig. 6

We now divide each data chunk into 2 packets of size 10 (thus $N = 2K = 6$) and route these packets. Since under per-node resource constraint dividing a big data chunk into packets better utilizes network resource, we have a performance gain shown in Fig. 6. Note that the inter-arrival time is between every two packets of size 10.

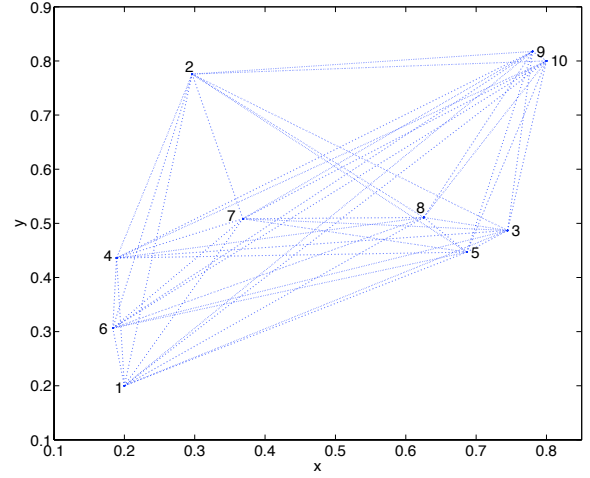


Fig. 5. A randomly generated network. Transmitting a single data chunk of size 20 on this network takes 7.1853 units of time.

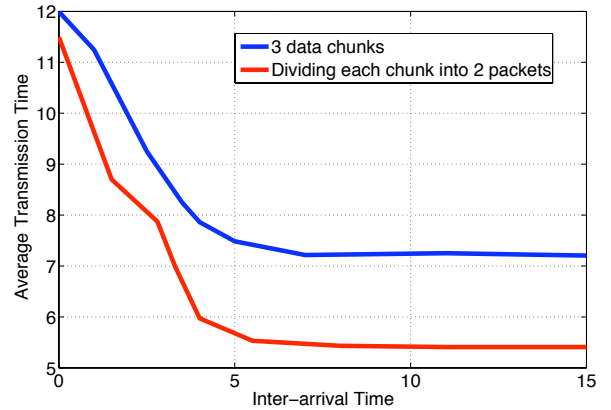


Fig. 6. Average transmission time under different inter-arrival times. The non-smoothness is caused by the non-optimality of our algorithm.

3) *Sum bandwidth constraint:* We next simulate the performance of the system with sum bandwidth constraint. We consider a single data chunk of size 20 which is divided into N packets. We let $W_T = 10$, $P_i = P = 1$, $N_0/2 = 1$. Note that the total bandwidth is the sum of individual bandwidth W_i in per-node bandwidth constraint simulation setting.

The simulation result is shown in Fig. 7. Interestingly, we note that under sum bandwidth constraint, dividing a data chunk into more and more packets would not help to decrease the total transmission time. This is the result of Theorem 4; since each packet is also $\frac{1}{N}$ of the size of the data chunk, the total transmission time would not change. An example of such routing is shown in Fig. 8.

We also notice that in Fig. 7, the performance of sum bandwidth constraint is better than per-node bandwidth constraint. The reason lies in the fact that in allocating each node a fixed amount of bandwidth as in per-node constraint, the nodes which cannot transmit actually waste the resource assigned

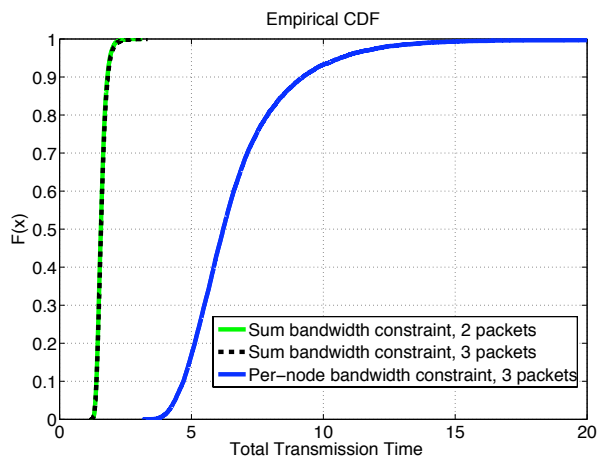


Fig. 7. Simulation of sum bandwidth constraint and comparison with per-node bandwidth constraint.

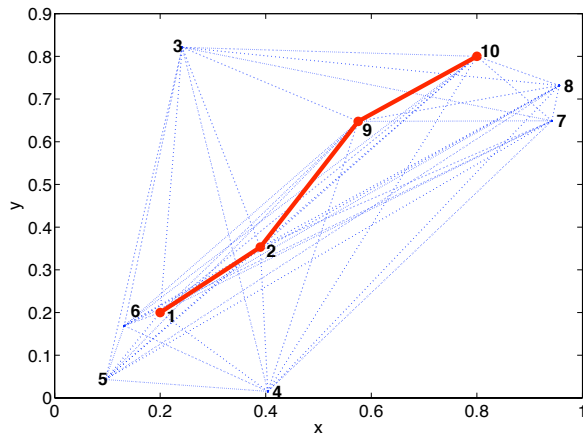


Fig. 8. An example of route under sum bandwidth constraint is shown. All packets follow the route [1, 2, 9, 10]. The total transmission time is 1.4299 and it is the same for any N .

to them. In sum bandwidth constraint, the network gathers all available resource and distributes to serve the most needed nodes; the nodes which are not eligible to transmit are not assigned with resource.

V. CONCLUSION

In this paper we study the problem of cooperative communication using mutual information accumulation. We design and prototype a routing and resource allocation algorithm based on solving a LP based problem iterating between two subproblems; finding the best decoding order and finding the best resource allocation given a decoding order. Our algorithm exploits both the spatial and temporal route diversity and suggests that performance can be improved by dividing a big data chunk into several small packets. Our algorithm also supports packets with different arrival times. Future work will focus on optimizing the power allocation P_i , network with nodes interference, and the impact of non-ideal codes.

REFERENCES

- [1] S. C. Draper, L. Liu, A. F. Molisch, and J. S. Yedidia, "Cooperative routing for wireless networks using mutual-information accumulation," *IEEE Trans. Inform. Theory*, vol. 57, Aug. 2011.
- [2] S. Biswas and R. Morris, "ExOR: Opportunistic multi-hop routing for wireless networks," *ACM SIGCOMM*, vol. 35, pp. 133–144, Aug. 2005.
- [3] I. Maric and R. D. Yates, "Cooperative multihop broadcast for wireless networks," *IEEE J. Select. Areas Commun.*, vol. 22, pp. 1080–1088, Aug. 2004.
- [4] —, "Cooperative multicast for maximum network lifetime," *IEEE J. Select. Areas Commun.*, vol. 23, pp. 127–135, Jan. 2005.
- [5] J. Chen, L. Jia, X. Liu, G. Noubir, and R. Sundaram, "Minimum energy accumulative routing in wireless networks," *Proc. IEEE INFOCOMM*, vol. 3, pp. 1875–1886, Mar. 2005.
- [6] J. Castura and Y. Mao, "Rateless coding for wireless relay channels," *Proc. Int. Symp. Inform. Theory*, pp. 810–814, Sept. 2005.
- [7] R. Gummadi and R. S. Sreenivas, "Relaying a fountain code across multiple nodes," *IEEE Info. Theory Workshop*, pp. 149–153, May 2008.
- [8] M. Mitzenmacher, "Digital fountains: A survey and look forward," *Proc. IEEE Inform. Theory Workshop*, pp. 271–276, Oct. 2004.
- [9] M. Luby, "LT codes," *Proc. Symp. Foundation of Computer Science*, pp. 271–282, 2002.
- [10] A. Shokrollahi, "Raptor codes," *Proc. Int. Symp. Inform. Theory*, vol. 52, pp. 2551–2567, Jun. 2006.
- [11] R. Urgaonkar and M. J. Neely, "Routing with mutual information accumulation in wireless networks," Tech. Rep., Aug. 2010.
- [12] S. Katti, D. Katabi, H. Balakrishnan, and M. Medard, "Symbol-level network coding for wireless mesh networks," *ACM SIGCOMM*, vol. 38, pp. 401–412, Aug. 2008.
- [13] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "XORs in the air: Practical wireless network coding," *ACM SIGCOMM*, vol. 36, pp. 243–254, Sept. 2006.
- [14] M. J. Neely and R. Urgaonkar, "Optimal backpressure routing for wireless networks with multi-receiver diversity," *Proc. Conf. Info. Sci. Sys.*, pp. 18–25, Jan. 2006.
- [15] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Automat. Contr.*, vol. 37, pp. 1936–1948, Apr. 1992.
- [16] —, "Dynamic server allocation to parallel queues with randomly varying connectivity," *IEEE Trans. Inform. Theory*, vol. 39, pp. 466–478, Jun. 1993.
- [17] S. Sharma, Y. Shi, Y. T. Hou, and S. Kompella, "An optimal algorithm for relay node assignment in cooperative Ad Hoc networks," *IEEE/ACM Tran. Networking*, vol. 19, pp. 879–892, Jun. 2011.
- [18] C. K. Lo, S. Vishwanath, and R. W. H. Jr., "Relay subset selection in wireless networks using partial decode-and-forward transmission," *Proc. Vehic. Tech. Soc. Conf.*, p. 2395, May. 2008.
- [19] E. Rozner, J. Seshadri, Y. A. Mehta, and L. Qiu, "SOAR: Simple opportunistic adaptive routing protocol for wireless mesh networks," *IEEE Tran. Mobi. Comp.*, vol. 8, pp. 1622–1635, Dec. 2009.
- [20] A. F. Molisch, N. B. Mehta, J. S. Yedidia, and J. Zhang, "Performance of fountain codes in collaborative relay networks," *IEEE Trans. Wireless Comm.*, vol. 6, pp. 4108–4119, Nov. 2007.
- [21] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Standards Association Std., June 2007.
- [22] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, 1948.
- [23] T. H. Cormen, C. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, 2nd ed. MIT Press and McGraw-Hill, 2001.