

Embracing Distributed MIMO in Wireless Mesh Networks

Apurv Bhartia*, Yi-Chao Chen*, George P. Nychis†, Lili Qiu*

The University of Texas at Austin*

CMU⁺

Abstract – This paper proposes a novel routing protocol, DM^+ , to achieve distributed spatial multiplexing gain in wireless mesh networks. It lets multiple nodes simultaneously send and receive different streams over each hop. To realize this goal, we propose an optimization framework that jointly optimizes spatial multiplexing, routing, and rate limits while taking into account wireless interference. We further design and implement a practical routing protocol that (i) enforces the optimized multiplexed routes, (ii) synchronizes transmissions from different senders, (iii) encodes and decodes analog signals to support simultaneous transmissions, and (iv) compensates for the frequency offset incurred over a multihop path. Using Qualnet simulation and USRP implementation, we show it significantly out-performs state-of-the-art shortest path routing and opportunistic routing protocols. To our knowledge, this is the first routing protocol and prototype that achieves distributed spatial multiplexing in a real multihop network.

I. INTRODUCTION

The popularity of wireless mesh has increased rapidly due to ease of deployment and low cost. It is also being considered as one of the key technologies for 5G (*e.g.*, [1], [32]). Significant work has been devoted to developing routing protocols in such networks, such as designing routing metrics (*e.g.*, [6], [7]), network coding (*e.g.*, [14]), and opportunistic routing (*e.g.*, [3], [4], [25]). While these approaches differ in details, they all aim to reduce the number of transmissions required to send traffic end-to-end. Orthogonal to these approaches, we propose a new routing, called DM^+ , which creates virtual MIMO nodes in a *multihop* setting, to support multiple simultaneous transmissions, thereby reducing the time required for the transmissions. DM^+ goes beyond the existing work on virtual MIMO systems in single-hop networks, and brings MIMO multiplexing gain to multihop wireless networks.

Illustrative examples: To illustrate the idea, consider a simple example in Figure 1, where source A and destination F have 2 antennas each and the intermediate nodes have one antenna. Node A wants to send to F , and all links interfere with each other. In traditional routing, even though the source has two antennas, since none of its neighbors have multiple antennas, it can only send out one frame at a time. Similarly, the destination can only receive one frame at a time since each of its neighbors have one antenna. As the shortest path includes 3 hops, it takes 3 time slots to transmit a packet from A to F . In comparison, in DM^+ , we leverage the antennas belonging to different nodes to increase multiplexing opportunities. Node A transmits 2 frames simultaneously to its forwarders B and C . Even though B and C only get one signal each and cannot decode the two unknown transmissions individually, they collectively still get two signals. As long as they forward these signals to the destination, the destination can still decode them. So we let B and C forward them simultaneously to D and E (after

linear transformation), which further forward to F together. We transmit two frames in 3 timeslots, doubling the throughput.

This example shows that traditional routing is bottlenecked by the minimum number of antennas at the two end points of a link. In comparison, DM^+ treats multiple next hop nodes as a single virtual node with multiple antennas and increases the number of concurrent streams. So traditional routing supports $\min(n_i, n_{i+1})$ streams on the i -th hop, while DM^+ supports $\min(\sum_j m_{i,j}, \sum_k m_{i+1,k})$ streams, where n_i is the number of antennas at the i -th hop node, and $m_{i,j}$ is the number of antennas at the j -th node of the i -th hop. Its benefit over traditional routing in theory can become arbitrarily large. For example, the gain is N when the source and destination each has N antennas and are separated by N intermediate nodes each with 1 antenna since shortest path routing (SPP) sends 1 frame at a time whereas the new routing sends N frames simultaneously. The gain is especially high when nodes have different numbers of antennas. This is common, as mesh networks are formed organically and different nodes are likely to have heterogeneous hardware. The antenna heterogeneity tends to increase with 802.11n and 802.11ac. While the above example lets the source and destination have more antennas than relayers, this is not required. A relay with multiple antennas can also transmit simultaneously to multiple downstream nodes, each with one antenna.

Moreover, heterogeneous numbers of antennas are *not* required to reap performance gain. In the example in Figure 1, when all nodes have one antenna each, the traditional routing takes 3 time slots to deliver 1 packet end-to-end, whereas the new routing takes 5 time slots to deliver 2 packets end-to-end. The latter is because on the first hop the two packets have to be sent sequentially due to one antenna at the source, which takes 2 time slots; on the second hop, two transmissions can be sent together from B and C to D and E , which takes 1 time slot; and on the third hop, two transmissions have to be sent sequentially due to one antenna at the destination. This gives 2.5 slots per packet and yields 20% improvement. The improvement increases with the number of nodes on each hop and with the number of hops with multiple forwarders.

Challenges: There have been several theoretical studies on distributed MIMO. For example, [20] extends the classic wireless capacity paper by Gupta and Kumar [9] to analyze the asymptotic performance of distributed MIMO in multihop

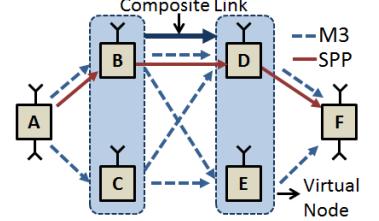


Fig. 1. Source A simultaneously sends two streams to B and C , which simultaneously forward to D and E , which further deliver to destination F .

networks. This approach is useful to understand how wireless capacity scales as the network grows to infinity when nodes are uniformly distributed and have random communication patterns. Different from the existing work on asymptotic wireless capacity, we optimize distributed MIMO for a specific network with specific traffic demands. We make no assumption about the homogeneity of nodes regarding their radio ranges, interference patterns, numbers of antennas, or traffic demands. Instead, this information is measured from the network and used as input to our system. We go beyond optimization, and develop a protocol to realize multiplexing gain in practice. Designing such a routing protocol poses the following challenges:

1. *Optimization challenges:* Different from existing routings, where we may simply prefer shortest paths, DM^+ should take into account not only the number of transmissions and the extent of wireless interference among them but also the level of spatial multiplexing. How to simultaneously incorporate all the factors to optimize network throughput for a given topology with given traffic demands?
2. *Protocol design and implementation challenges:* We need to address how to (a) enforce the routes computed from our optimization, (b) synchronize between multiple senders, (c) develop a coding design to support analog forwarding over multiple hops so that a destination can successfully decode, and (d) develop a technique to compensate for the frequency offset incurred over multiple hops. This calls for a new optimization framework to support distributed spatial multiplexing and accurately capture its impact on the end-to-end throughput.

Our approach: We develop a practical protocol, DM^+ , which performs joint spatial multiplexing, routing, rate limiting to effectively harness the MIMO multiplexing and diversity gain in multihop wireless networks.

1. *Optimization:* We propose a theoretical framework to optimize throughput under distributed MIMO routing. The formulation simultaneously harnesses both multiplexing gain and diversity gain to achieve high efficiency and resilience. The framework serves two purposes: (i) the throughput derived from the framework can be achieved under optimal scheduling, which gives a useful baseline; (ii) the optimized MIMO routes can be incorporated into a practical routing protocol (Section III).
2. *Protocol design and implementation:* We design DM^+ , the first prototype that supports distributed spatial multiplexing in a multihop network. DM^+ has several significant components: (i) a mechanism to enforce the sources' sending rates and intermediate nodes' forwarding rates according to the optimized routes and rate limits, (ii) synchronizing transmissions from different nodes for spatial multiplexing, (iii) an effective encoding scheme to combine concurrent analog signals and support successful decoding after multihop forwarding, and (iv) correcting frequency offset incurred along a multi-hop path. DM^+ is compatible with different MAC protocols including IEEE 802.11 (Section IV).
3. *Evaluation:* We demonstrate the feasibility and effectiveness of DM^+ using Qualnet simulation and a USRP testbed

experiments. Our results show it significantly out-performs state-of-the-art shortest path routing and opportunistic routing protocols (Section V and Section VI).

II. RELATED WORK

We broadly classify related work into: (i) routing in wireless mesh networks, (ii) network coding, and (ii) MIMO.

Routing in wireless mesh networks: Many routing protocols and techniques have been proposed for wireless mesh networks. For example, various routing metrics have been proposed for single path routing (*e.g.*, ETX [6] and WCETT [7]). Single path routing uses only one node on each hop. To take advantage of other nearby nodes, researchers propose opportunistic routing to leverage multiple forwarders to enhance robustness. Most existing opportunistic routing schemes exploits receiver diversity (*e.g.*, [4], [25]). SourceSync [21], an advanced version of opportunistic routing, exploits transmitter diversity in addition to receiver diversity by letting multiple transmitters send simultaneously to strengthen the received signal. [17] also exploits both transmitter and receiver diversity using multihop amplify-and-forward, where multiple sources amplify and forward simultaneously the analog signals they receive from the previous hops. The capacity growth is logarithmic with the diversity order. So opportunistic routing including SourceSync and amplify-and-forward can at best obtain logarithmic gain. In comparison, spatial multiplexing, which is the focus of this paper, supports multiple different streams simultaneously and increases throughput linearly with the number of transmit/receive antennas, much higher than opportunistic routing including [21], [17]. Distributed spatial multiplexing is much harder to realize since a forwarder may not be able to decode its received frames (as it is just one of multiple receivers) but should forward useful information to support end-to-end decoding at the destination.

Network Coding: MIMO sends more transmissions in a given time. An orthogonal line of routing research studies inter-flow network coding, which packs more information into each transmission. For example, COPE [14] develops a practical inter-flow network coding for unicast in multi-hop wireless networks. Analog Network Coding (ANC) [13] further extends the network coding concept to the analog level. Both approaches let a relay decode only if it knows beforehand all packets involved in coding except one packet. Network coding is complementary to distributed MIMO and the two can potentially be combined to further enhance network capacity by sending more transmissions in parallel and packing more information into each transmission.

MIMO: MIMO is a widely successful technology and many wireless devices and standards have adopted MIMO. The research on distributed MIMO in multi-hop networks has just started. [15] considers the practical aspect of MAC protocol design for distributed MIMO in multihop networks, whereas DM^+ supports different MAC protocols including WiFi. In terms of routing, the first category is opportunistic routing. It achieves logarithmic diversity gain instead of linear multiplexing gain, which we focus on. The second category of existing works studies spatial multiplex using theoretical analysis (*e.g.*, [31], [8], [30], [11], [2], [17], [19], [5]). They extend the classic

paper by Gupta and Kumar [9] to support distributed MIMO. [5] is closest to our work, but its formulation does not capture wireless interference and losses. Our paper differs from the existing work by (i) optimizing end-to-end throughput in distributed MIMO routing for a given network (instead of deriving asymptotic bounds based on geometry), (ii) supporting general topologies with arbitrary loss rates and interference patterns, and (iii) going beyond theoretical analysis to design a *practical* routing protocol and realize it in real networks.

Cooperative Relaying: There are five types of cooperative forwarding: amplify-and-forward, decode-and-forward, compress-and-forward, compute-and-forward, and quantize-map-forward. DM^+ is related to amplify-and-forward [17], but differs from [17] in that we achieve multiplexing gain whereas [17] achieves diversity gain. Moreover, DM^+ is the first protocol and implementation that realizes the spatial multiplexing gain in real multihop wireless networks.

III. DISTRIBUTED MIMO ROUTE OPTIMIZATION

In this section, we develop a framework to optimize end-to-end performance of distributed MIMO routing. To support distributed MIMO, we introduce a composite link cl , denoted as $(\{s_i\}, \{r_j\})$, which indicates the senders in the set $\{s_i\}$ are transmitting to the receivers in the set $\{r_j\}$. A composite link includes (i) one sender sending to one receiver in traditional communication, (ii) one sender sending to multiple receivers in opportunistic routing (*e.g.*, [3], [4], [25]), where multiple forwarders are chosen and whoever receives the frame can forward it, and (iii) multiple senders sending to multiple receivers for spatial multiplexing (*e.g.*, $(\{B, C\}, \{D, E\})$ in Figure 1). Only nodes within a composite link cooperate.

The main challenges in route optimization include (i) how to support distributed MIMO links, (ii) how to quantify the amount of information transmitted on distributed MIMO links especially under wireless losses, and (iii) how to capture information conservation on distributed MIMO links.

Given the composite links, this problem can be formulated as a linear program (LP) that maximizes the total network throughput subject to throughput constraints, information conservation constraints, opportunistic constraints, and interference constraints. The formulation supports both unicast and multicast flows, though we only focus on unicast flows in the protocol design and implementation. The optimization problem can be solved efficiently using solvers such as cplex. The optimization can be done periodically or whenever the network condition changes. Incremental computation can be used to reduce the computation cost. The routes can be computed at a central location, such as in Tesseract [29], or in a fully distributed manner. The distribution is similar to that in link-state protocols such as OSPF, where all nodes apply the same algorithm over the same data to arrive at consistent routes. This strategy works well for static networks. The input required for optimization is small and cheap to distribute. More specifically, the input includes traffic demands, link loss rates, and the conflict graph, which are $O(F)$, $O(E)$, $O(E^2)$, respectively, where F is the number of flows and E is the number of composite links. Among these three terms, $O(E^2)$ is the dominating term. The output is credit table as introduced

| | |
|----------------------------------|-------------------------------------------------------------------------------------------------------------|
| <i>Flows</i> | the set of unicast or multicast flows |
| $src(f)$ | source of flow f |
| $dest(f, d)$ | d -th destination of flow f |
| $Demand(f)$ | flow f 's traffic demand |
| $G(f)$ | throughput of flow f |
| $T(f, cl)$ | composite link cl 's sending rate for flow f |
| $Y(f, d, cl, r)$ | node r 's information receiving rate from cl for d -th destination in flow f ($d = 1$ for unicast) |
| $P(i, j)$ | inherent loss rate of link $i \rightarrow j$ |
| $NA(i)$ | number of antennas on node i |
| $NA(cl)$ | # antennas on all senders or receivers in whichever is smaller |
| $\mathcal{M}(i)$ | a subset of i 's neighbors |
| $S(senders(cl), \mathcal{M}(i))$ | delivery rate from cl 's senders to $\mathcal{M}(i)$ |

TABLE I. NOTATIONS FOR OPTIMIZING DISTRIBUTED MIMO ROUTES.

- ▷ *Input* : $Flows$, $Demand(f)$
- ▷ *Output* : $T(f, cl)$, $Y(f, d, cl, r)$
- maximize**: $\sum_{f \in Flows} G(f) - \beta \sum_{f, i} T(f, cl)$
- subject to**:
- [C1] $G(f) \leq Demand(f) \quad (\forall f)$
- [C2] $G(f) \leq \sum_{dest(f, d) \in receivers(cl)} Y(f, d, cl, dest(f, d)) \quad (\forall f, d)$
- [C3] $Y(f, d, cl, src(f)) = 0 \quad (\forall f, d, cl : src(f) \in receivers(cl))$
- [C4] $Y(f, d, cl, k) = 0 \quad (\forall f, d, k, cl : dest(f, d) \in senders(cl))$
- [C5] $\sum_{cl} Y(f, d, cl, i) \geq \sum_{cl'} \frac{NA(i)}{NA(cl')} Y(f, d, cl', j)$
 $\quad (\forall f, d, i : i \neq src(f), i \neq dest(f, d), i \in receivers(cl)$
 $\quad i \in senders(cl'), j \in receivers(cl'))$
- [C6] $S(senders(cl), \mathcal{M}(i)) NA(cl) T(f, cl) \geq \sum_{k \in \mathcal{M}(i)} Y(f, d, cl, k)$
 $\quad (\forall f, d, \mathcal{M}(i), cl : k \in receivers(cl))$
- [C7] $interference\ constraints\ on\ T(cl) \triangleq \sum_f T(f, cl)$

Fig. 2. Problem formulation to optimize distributed MIMO routes, where notations are introduced in Table I.

in Section IV-B, whose size is $O(F \times L^2)$, where L is the maximum number of composite links going through a node. In practice, most entries in the credit table are zeros and only non-zero entries should be stored, so they are much smaller.

Figure 2 shows the resulting LP. It takes as input the traffic demands of all flows $Demand(f)$ and inherent link loss rate $P(i, j)$ (*i.e.*, losses under no interfering traffic and can be efficiently measured), and outputs traffic and information receiving rate on each composite link (*i.e.*, $T(f, cl)$ and $Y(f, d, cl, r)$, respectively). The outputs will be converted to a routing configuration in DM^+ to realize the optimized routes as described in Section IV-B.

Optimization objective: Figure 2 shows one example of objective function, *i.e.*, the total throughput over all flows $\sum_{f \in Flows} G(f)$ minus the total amount of wireless traffic scaled by a small weighting factor β . It reflects our goal of (i) maximizing total throughput and (ii) preferring the least amount of traffic among all solutions that support the same total throughput (*e.g.*, avoiding loops and unnecessary traffic). We use a small $\beta = 0.001$ since (i) is our primary objective and we prefer the least traffic only when the total throughput is the same. In addition, our framework can also optimize other objectives, such as a linear approximation of proportional fairness, defined as $\sum_{f \in Flows} \log G(f)$, to take into account both fairness and throughput [22] (evaluated in Section VI), or maximizing total revenue, which is a function of throughput.

Throughput constraints: To ensure $G(f)$ is the throughput of flow f , it has to satisfy [C1] and [C2] in Figure 2, which ensure the throughput of a flow should be no more than its traffic demand and no more than the total information delivered from all links incident to the destination of flow f .

Information conservation constraints for composite links: A feasible routing solution should satisfy information conservation, given by constraints [C3–C5] in Figure 2. Constraint [C3] ensures no incoming information to a traffic source, constraint [C4] ensures no outgoing information from a destination, and constraint [C5] represents flow conservation at an intermediate node i , *i.e.*, the total amount of incoming information is no less than the total amount of out-going information. In order to capture that the composite link may involve multiple transmitters, we let $Y(f, d, cl', j)$ denote the total information transmitted by all the senders in the composite link cl' . The amount of information transmitted by a node i in cl' is only a fraction of $Y(f, d, cl', j)$, and the fraction is determined by the number of its antennas divided by $NA(cl')$, which is the total number of antennas of all the senders or receivers in cl' , whichever is smaller. That is, $NA(cl') = \min(\sum_{i \in \text{senders}(cl')} NA(i), \sum_{j \in \text{receivers}(cl')} NA(j))$. For example, two senders send to two receivers in cl' and they have one antenna each. Each sender only transmits half of the information on the composite link.

Opportunistic constraints for composite links: As wireless is broadcast medium, we let nodes to extract different information from the same transmission. We capture this notion using opportunistic constraints in [C6], which relate traffic to the amount of information delivered under multiple streams.

One sender to multiple receivers: Suppose a sender s has K neighbors. We enumerate all subsets of its neighbors. For each neighbor set $\mathcal{M}(i)$, $S(i, \mathcal{M}(i))T(f, i) \geq \sum_{k \in \mathcal{M}(i)} Y(f, d, i, k)$, where $S(i, \mathcal{M}(i))$ denotes the delivery probability from node i to at least one node in $\mathcal{M}(i)$. The inequality indicates the total traffic successfully delivered to at least one neighbor in $\mathcal{M}(i)$ is no less than the total non-overlapping information delivered to $\mathcal{M}(i)$. When i has many neighbors, we limit the number of such constraints by only enumerating neighbor sets of size 1, 2, and K (*i.e.*, enumerate only $O(K^2)$ instead of $O(2^K)$ neighbor sets). If loss rates of different links are independent, which holds for some networks [27], $S(i, \mathcal{M}(i)) = 1 - \prod_{k \in \mathcal{M}(i)} P(i, k)$. When the link losses are correlated, we empirically measure $S(i, \mathcal{M}(i))$.

Multiple senders to multiple receivers: To further generalize to multiple senders in the composite link, we observe that the amount of information delivered to $\mathcal{M}(i)$ increases linearly with $NA(cl)$. Moreover, when multiple antennas transmit, a signal is received successfully only when all of the transmitters in cl succeed in sending to at least one receiver. (cl is defined as the set of nodes that actually transmit in the composite link.) So we change $S(i, \mathcal{M}(i))$ to $S(\text{senders}(cl), \mathcal{M}(i))$, which denotes the probability of successfully delivering traffic from all senders in cl to at least one receiver in $\mathcal{M}(i)$. $S(\text{senders}(cl), \mathcal{M}(i)) = 1 - \prod_{k \in \mathcal{M}(i)} P(\text{senders}(cl), k)$ where the loss rate from any sender in $\text{senders}(cl)$ to k is $P(\text{senders}(cl), k) = 1 - \prod_{i \in \text{senders}(cl)} (1 - P(i, k))$ if the

delivery rates of these links are independent. Otherwise, we can again empirically measure the joint delivery rate.

Interference constraints: Wireless interference prevents certain composite links from being active simultaneously. To capture these constraints, we extend the conflict graph model developed for point-to-point links in [12] to support distributed MIMO links as follow. While conflict graph is used for route optimization, the routing protocol can run on any MAC, including WiFi, used in our Qualnet implementation.

Two composite links $(\{s_i\}, \{r_j\})$ and $(\{s'_k\}, \{r'_l\})$ interfere if either of the following conditions holds: (i) if there exist s_i and s'_k such that they are within carrier sense range of each other, or (ii) if there exists r'_l within s_i 's interference range or exists r_j within s'_k 's interference range. Alternatively, one can use measurements to determine if two composite links interfere with each other based on whether the delivery rates of the receivers in the two composite links differ when they send separately versus they send simultaneously.

We construct a conflict graph similar to [12]. The only difference is that links are now composite links. We use a vertex in a conflict graph to denote a composite link. We draw an edge between two vertices if the links corresponding to the vertices interfere according to the above definition. We derive lower bounds using independent set constraints similar to [12].

Rate selection: DM^+ should select rate for each composite link. A lower rate may increase diversity and multiplexing gain due to more neighbors. So simply using the highest rate may not be the best, and we should take into account the entire network when picking the data rate. Rate selection can be cast as a routing problem and optimized in the same framework by considering a node has multiple links to each of its neighbors, one for each rate. So we can select rate by replacing a composite link cl in the above formulation with a virtual composite link (cl, rate) , and let all the constraints apply to the virtual composite link. In addition, we add interference constraints among these virtual composite links – traffic from the same physical composite link over all rates interferes. The optimization result $T(f, cl, \text{rate})$ specifies how fast a composite link cl should transmit for flow f at a given rate. It may let cl split traffic across multiple rates. We evaluate our rate selection in Section VI.

Enumerating composite links: Since a composite link is defined by a set of senders and receivers, the number of composite links grows exponentially with the node degree. To enhance scalability, instead of enumerating all possible sender and receiver pairs, we enumerate the most useful composite links. To pick useful composite links, we make the following observations. Let S denote the sender set and R denote the receiver set in a composite link. We have the following requirements on the composite links: (i) All nodes in R should hear all nodes in S ; (ii) All nodes in S should hear each other so that they can synchronize their transmissions; (iii) To efficiently route a flow, we require traffic to make progress on each hop. That is, the receivers in R should all be closer to the destination than the senders in S (*e.g.*, in terms of the expected number of transmissions to reach destination (ETX) [6]).

Based on the observations, for every node i that is closer to the destination than the source, we enumerate one or

more composite links involving i as follows. We partition a subset of $\{i, \mathcal{N}(i)\}$ into a sender set and a receiver set in a composite link, where $\mathcal{N}(i)$ denotes i 's neighbors. Two nodes are neighbors if their delivery rate is high enough. To do that, we enumerate subsets of i 's neighbors, denoted as $\mathcal{M}(i)$, where $\mathcal{M}(i) \subseteq \mathcal{N}(i)$. We add i to the sender set, and iteratively add nodes from $\mathcal{M}(i)$ to the sender set as long as the nodes can hear every node in the current sender set. This ensures (ii). We then form a receiver set by adding nodes from i 's remaining neighbors that hear from everyone in the sender set and are closer to the destinations than the senders in the composite link. The former condition ensures (i) and the latter condition ensures (iii). To further enhance scalability, if $\|\mathcal{N}(i)\| = K$ is too large, instead of enumerating all possible subsets $\mathcal{M}(i)$, we enumerate subsets of size 1, 2, and K (*i.e.*, enumerate $O(K^2)$ instead of $O(2^K)$ neighbor subsets). Other heuristics of selecting composite links can be used without affecting the optimization framework or protocol design.

IV. DM^+ PROTOCOL DESIGN

A. Overview

In this section, we develop a novel *practical* routing protocol. In the following, we use a source and destination to refer to the two end-points of a flow, and use a sender and receiver to refer to the two end-points of a wireless link (*e.g.*, a forwarder serves as both a sender and a receiver).

At a high level, a source in DM^+ divides packets into batches and broadcasts linear combinations of the packets in each batch. Its sending rate is determined by the optimization result $T(f, cl)$ (derived in Section III), where cl refers to any composite link that includes the source in the sender list for flow f . Every packet header specifies cl , denoting which senders transmit to which receivers, and the coding matrix used so far in order for the destination to reconstruct the packets based on the received signals.

Upon receiving a packet, the forwarder checks if it is one of the intended receivers. If so, it extracts the coding coefficients from the coding matrix specified in the header, re-encodes data to avoid duplicates, and updates the coding matrix. We use a *simple* credit based scheme to control the forwarding rate.

The destination extracts coding coefficients from all received packets in the current batch and inserts them into a coding matrix. When the coding matrix becomes full rank, it decodes the packets and sends an end-to-end ACK via MAC-layer unicast to inform the sender to move to the next batch.

Realizing such a protocol poses significant challenges:

- How to enforce the optimized MIMO routes and rate limits computed from our optimization framework?
- How do senders in a composite link synchronize their transmissions? How does synchronization interact with a MAC protocol? Synchronization is important for OFDM as it has tight timing requirements. Closely related, how do we perform automatic gain control?
- To support spatial multiplex, DM^+ performs coding at the physical layer to combine analog signals. How to encode and decode the analog signals to support simultaneous transmissions and successful decoding at a destination, which can be multiple hops away? How to take into

account effects of wireless channel, since received signal is affected by both coding at nodes and wireless channel?

- There exists frequency offset between any two nodes. Unlike traditional routing, which resolves the offset every hop, how should a destination compensate for the frequency offsets incurred over multiple hops from multiple senders?
- How to compress the coding matrix, whose values change across OFDM subcarriers?

Below we address these issues in turn.

B. Enforcing Forwarding Rates

The forwarding rates (*i.e.*, $T(f, cl)$ and $Y(f, d, cl, r)$) derived from the optimization procedure in Section III represent the distributed MIMO routes. We enforce forwarding rates based on the optimization results using a credit-based scheme. The credit table entry is $[f, cl_{in}, cl_{out}, credit]$, which denotes how many transmissions to generate on the outgoing composite link cl_{out} upon receiving a packet from cl_{in} for the flow f .

We derive the credit table based on the optimization results from Section III as follows. The credit is $U \cdot R$, where U is amount of information should be transmitted and R is the amount of redundancy to include in order to successfully deliver the signals to the receivers of the composite link. They can be computed as follows:

$$U(r) = \frac{Y(f, d, cl_{in}, r)}{T(f, cl_{in})(1 - P(send(cl_{in}), r))} \frac{T(f, cl_{out})}{\sum_{r \in send(cl_{out}')} T(f, cl_{out}')} \quad (1)$$

$$R(r) = \frac{T(f, cl_{out}) NA(cl_{out})}{\sum_{r \in recv(cl_{out})} Y(f, d, cl_{out}, r)} \quad (2)$$

where $NA(cl) = \min(\sum_{i \in send(cl)} NA(i), \sum_{j \in recv(cl)} NA(j))$, which represents the maximum number of concurrent streams that can be supported on a composite link cl . We omit the derivation in the interest of brevity.

When a node receives a new packet, it extracts the tuple $[f, cl_{in}]$ from the packet header, finds all entries $[f, cl_{in}, cl_{out}]$ that match the tuple in the credit table, increments the credits associated with $[f, cl_{out}]$ by the amounts specified in the credit table entries. Whenever any $[f, cl_{out}]$ has credit no less than 1, the node generates a forwarding record $[f, cl_{out}]$, puts it into a queue, and decrements the credit by 1. This continues until the credit becomes less than 1. When a node detects the medium is idle, it pulls the first record from the queue, generates a linear combination of all signals it receives from the flow within a batch, and transmits on the composite link cl_{out} . Deferring actual signal generation till the time of transmission allows a node to include information from its latest receptions and improves the chance of generating linearly independent transmissions since it will receive more frames by then.

When rate adaptation is used, the entry is $[f, cl_{in}, rate_{in}, cl_{out}, rate_{out}, credit]$, where $rate_{in}$ and $rate_{out}$ are incoming and outgoing rates. The credit derivation is similar except that we use $(cl_{in}, rate_{in})$ and $(cl_{out}, rate_{out})$ instead of cl_{in} and cl_{out} . If any $[f, cl_{out}, rate_{out}]$ has at least 1 credit, a node generates a forwarding record so that a packet will be sent on cl_{out} at $rate_{out}$.

C. Synchronizing Transmissions

Triggered-based synchronization: DM^+ works well on top of IEEE 802.11. Any node with traffic to transmit carrier



Fig. 3. Synchronization in DM^+ .

senses as usual. The node winning the channel checks to see if the packet to be transmitted should involve more senders. If not, it transmits as usual. Otherwise, it invites other senders to join its data transmission while silencing all the other nearby senders. This is achieved using a trigger message which contains $[f, cl_{out}]$ and network allocation vector (NAV) that specifies the time till the end of data transmission to prevent other nodes from interrupting the join transmission. NAV can be set based on the common packet size (if there is one), otherwise MTU. Upon receiving the trigger message, a node checks its queue for a matching record (which is generated based on its credits). If it finds one, it transmits a header and preamble in clean according to an increasing order of sender node IDs and sends the data frame together, as shown in Figure 3. If a node x in cl_{out} does not have a packet to transmit, it skips transmission. The downstream nodes can easily tell which upstream nodes have participated in the transmission based on whose headers and preambles are present. Similarly, loss of a trigger does not affect the correctness of the protocol, but only reduces efficiency since fewer streams are transmitted and the next hop node(s) can also detect missing senders based on the headers and preambles. When rate adaptation is enabled, the sender transmits packets at proper rates according to the credits computed in Section IV-B. We implement our protocol in Qualnet on top of IEEE 802.11 as described above. Our implementation supports *multiple flows*.

To support successful delivery of simultaneous transmissions, different transmissions should be *loosely synchronized*. The synchronization error at different receivers should be within the Cyclic Prefix (CP), a guard interval in OFDM (*e.g.*, $0.8 \mu s$ in IEEE 802.11a). This ensures an FFT window contains information from all simultaneous transmissions. To achieve this, the transmitters need to time their transmissions accordingly. The timing of their transmissions depend on several factors, such as propagation delay, hardware turnaround delay, packet detection delay [21]. As SourceSync [21], we measure these delay and calculate the time at which a transmission should be sent in order for it to synchronize with the lead sender. We leverage the timestamp mechanism in USRPs to tag a transmission with the time that it should leave the front-end.

As described in [21], it is infeasible to perfectly synchronize at multiple receivers. But we can formulate a linear program to compute the transmission time at each sender to minimize the maximum pairwise misalignment across all senders and receivers in a composite link. If the maximum misalignment is larger than the standard CP, we can use a larger CP by using a larger FFT without increasing CP overhead [28].

Automatic Gain Control (AGC): A receiver uses AGC to adjust the incoming signal level to an appropriate operating range. Normally, the AGC is tuned based on the received power of a preamble. However, in DM^+ the preamble is transmitted individually by each sender whereas the data symbols are transmitted by multiple senders simultaneously and have

higher energy than the preambles transmitted by individual senders, which causes AGC to be tuned inappropriately. To support AGC, we let all the participating senders send an extra preamble synchronously and the receiver adjust the gain based on the energy of this extra preamble.

D. Encoding and Decoding

In DM^+ , a source modulates bits into analog signals and encodes the analog signals (*e.g.*, by scaling the signals or performing linear combination of different signals). The signals are not only coded by physical nodes but also inherently by the wireless channel as the channel attenuates, rotates, and combines different signals. When the signals arrive at an intermediate node, it re-encodes the signal and propagates the updated coding coefficients in the packet header. The destination extracts these coefficients from the packet header, constructs a linear system, and decodes the analog signals to digital symbols by finding the solution that best satisfies the linear system. In order to achieve successful decoding at a destination, the destination should (i) correctly construct the linear relationship between the original signals and received signals and (ii) receive enough linearly independent constraints to solve the system. Below we specify coding at a source, forwarder, and destination. To simplify description, we focus on how to encode and decode the first symbol in a packet, since the operation is identical for all symbols.

Source: A source s divides traffic into batches of K packets. It then modulates the j -th packet in a batch into analog signal a_j , and transmits $t_{s,i}$ that is linear combinations of these analog signals (*i.e.*, $t_{s,i} = \sum_{j=1..K} c_{s,i,j} a_j$, where $c_{s,i,j}$ is a coding coefficient selected for the j -th packet in the i -th transmission). The source includes $c_{s,i,j}$ in the header to facilitate decoding at the destination.

Forwarder: Upon receiving a new signal, a forwarder constructs the relationship between the original signal and received signal based on the coding coefficients in the packet header (sent in clean) and channel estimates (calculated using the received preamble). It then re-encodes the signal to avoid duplicate transmissions, and propagates the updated coefficients and re-encoded signals.

Different from digital coding (*e.g.*, in MORE [4]), here coding is performed by both physical nodes and wireless channel. Let $t_{s,i}$ denote the i -th signal transmitted by sender s . It is attenuated and rotated by the channel and becomes $h_{s,r} t_{s,i}$ as it traverses from the sender s to a receiver r , where $h_{s,r}$ is a complex number, representing the channel coefficient from s to r . Moreover, when there are multiple transmitted signals, they naturally add up in the air and the final resulting signal that the receiver gets is the sum of the signals coming from all the active senders s , which is $\sum_s h_{s,r} t_{s,i} = \sum_s h_{s,r} \sum_j c_{s,i,j} a_j$. Therefore, at each hop the signal is first transformed by encoding at a physical node and then by the wireless channel. We should keep track of the aggregate transformation of an original signal from the source to the destination so that the destination can correctly reconstruct a linear system for decoding.

Based on this observation, let us examine how a forwarder updates the coding coefficients. Specifically, the coefficient

$c_{s,i,j}$ included in the header of the i -th transmission from node s denotes the scaling and rotation that have been done to the j -th original packet by the physical nodes and wireless channel from the source up to node s . The forwarder r extracts $c_{s,i,j}$ from the packet header of every active previous hop s involved in the transmission, and estimates the channel coefficient $h_{s,r}$ using the preamble from s as in traditional communication. This is possible because both packet headers and pREAMbles are transmitted in clean and only data transmissions are concurrent. The forwarder r derives the relationship between its received signal R and original signal as $R = \sum_s h_{s,r} \sum_j c_{s,i,j} a_j$. When the forwarder is allowed to transmit as determined by the credit-based routing mechanism (Section IV-B), it generates and forwards a linear combination of all the signals it has received so far from the current batch (*i.e.*, k -th transmission contains $\sum_i R_i \beta_{k,i}$, where $\beta_{k,i}$ is the new coding coefficient) and includes the updated coefficient $c_{r,k,j}$ in the packet header of its k -th transmission, where $c_{r,k,j}$ is a reduced expression that captures all the linear transformation performed so far in the k -th transmission for the j -th original packet and $c_{r,k,j} = \sum_i \sum_s \beta_{k,i} h_{s,r} c_{s,i,j}$.

Destination: The destination r decodes the signals by extracting the coding coefficient used so far till the previous hop s from the current packet header and estimating the channel coefficient on the last hop using the preamble. The received signal $R = \sum_s h_{s,r} \sum_j c_{s,i,j} a_j$. So the destination inserts the following row $[\sum_s h_{s,r} c_{s,i,1}, \sum_s h_{s,r} c_{s,i,2}, \dots, \sum_s h_{s,r} c_{s,i,j}]$ to a coding matrix M . When M becomes full rank, it decodes by solving $Mx = R$, where R is the received signal and x is the transmitted symbol. Since x takes discrete values (*e.g.*, x is either -1 or $+1$ in BPSK), we can solve x by enumerating all combinations of x and finding the combination whose Mx gives the closest match to the actual received signal.

Selecting coding coefficients: A simple way is to let every node randomly select coding coefficients. Several papers (*e.g.*, [10], [4] have demonstrated the effectiveness of random codes. In our context, we need to address an additional issue – random coding coefficients may make the decision boundary of the combined signal too small and result in decoding errors. For example, two coded signals: $c_1 h_{11} x_1 + c_2 h_{21} x_2$ and $c_1 h_{11} x'_1 + c_2 h_{21} x'_2$, where $(x_1, x_2) \neq (x'_1, x'_2)$, may be too close to distinguish and cause decoding errors. In order to prevent this, we impose the following requirement on the coding coefficients. When k nodes transmit simultaneously, their coding coefficients should satisfy: $\forall x_{1..k} \neq x'_{1..k}: \min_{x_{1..k}, x'_{1..k}} \text{dist}(\sum_i c_i h_{i1} x_i, \sum_i c_i h_{i1} x'_i) > \text{threshold}$. Essentially it tries to ensure the minimum Euclidean distance between two different combined signals exceeds a threshold to avoid ambiguity during decoding. To achieve this condition, we let all senders except the last sender randomly select their coding coefficients and broadcast the selected coefficients, and the last sender searches for the coefficient that satisfies the condition based on the channel coefficient h , which is fed back by the next hops through piggyback either periodically or upon significant channel changes. If it cannot find a solution, among all the coding coefficients it searched so far, it selects the one that gives the largest minimum distance. This selection significantly improves the delivery rate over random selection.

To ensure transmitted symbols have the same power and avoid saturating the receiver's ADC, every node scales the amplitude of outgoing signal by $\sqrt{1/(maxMag * n_{tx})}$, where n_{tx} is the number of MIMO transmitters and $maxMag$ is the maximum magnitude of all symbols in the current packet.

Error propagation: A relay node in DM^+ forwards analog signals before demodulation. Analog forwarding works well under high SNR, but suffers under low SNR. Therefore, one can selectively enable DM^+ on portions of a network with high SNR to enjoy spatial multiplexing gain. In this work, we use DM^+ in an entire network. In future work, we plan to explore how to automatically decompose a large network into several building blocks, where DM^+ is applied to each block independently and the nodes on the boundary of a block demodulate first before forwarding to limit error propagation.

E. Frequency Offset Correction

Frequency offset is an artifact of radio hardware due to the unintentional drift in the frequency of the numerically controlled oscillator (NCO) driving the radio. Suppose node A transmits x_a to node B . B receives y_b : $y_b = h_{ab} e^{2\pi \Delta f t} x_a$. This frequency offset $\Delta f = (f_b - f_a)$ causes a phase shift, which increases with the packet size. If unaccounted for, it can lead to decoding failures. Since frequency offsets remain relatively stable, nodes in DM^+ measure and report their pairwise frequency offsets along with propagation delay during the network startup time, and then multiplies the outgoing symbol at time t by $e^{-2\pi \Delta f t}$. However, it is impossible to correct Δf perfectly, there still remains a residual offset, which can build up over time resulting in large phase errors.

While several methods have been proposed to address this issue in one link (*e.g.*, using pilot to track this phase error throughout the packet), how to track and compensate for residual offset after multihop forwarding and for multiple senders poses a new challenge.

A natural solution is to track the residual offset for each hop along the path and passes the information to the destination for decoding. However, this incurs significant overhead since the frequency offset may change every symbol and conveying coding coefficients on per-symbol basis is too expensive. The cost is even higher when multiple senders transmit, since their frequency offsets are different and hard to compress.

Instead of compensating for the frequency offset per-hop, we show it is sufficient to perform an end-to-end compensation of frequency offset. We estimate the end-to-end frequency offset using the pilot symbols, where each relay amplifies and relays the pilot symbols it receives from upstream. The pilot symbol arriving at the destination now contains the phase rotation from the frequency offset accumulated end-to-end. So the destination can decode the data by first compensating for the end-to-end frequency offset using the pilot and then plugging the data signal after compensation to the linear system for decoding. For this to work, the pilot symbols should be sent without overlap so that the exact offset is tracked. When multiple relay nodes forward data simultaneously, they should send the pilots in turn. Since pilot signal tracks the frequency offset over time, this approach works even when frequency offset changes. In comparison, the existing works

(e.g., [18]) only work for one sender and do not support multiple senders, which is necessary to achieve multiplexing.

F. Reducing Header Overhead

An intermediate node should convey coefficients so that the destination can correctly reconstruct the linear system. The coefficients are determined by the coefficients chosen by each sender and the channel coefficients. In OFDM, the channel coefficients vary across different subcarriers, and specifying coefficients for each subcarrier incurs too much overhead (e.g., IEEE 802.11n has 54 data subcarriers). We use least square to compress the coefficients across subcarriers using a degree 4 polynomial, which requires transmission of only 5 numbers. It yields a small approximation error ($\approx 4 - 10\%$). We also evaluate using 802.11 20MHz channel state information (CSI) traces, and observe similarly low approximation error.

V. TESTBED IMPLEMENTATION

We implement DM^+ on USRP. We create a testbed of 8 USRP N200 nodes randomly placed, each equipped with an XCVR2450 frontend. Some experiments involve multi-antenna nodes, which are created by connecting them using a MIMO cable. We use 1MHz bandwidth due to the high hardware turnaround time when a node acts as a half-duplex transceiver, required for relay nodes. We use 64 OFDM subcarriers, FFT window of size 96, cyclic prefix of 24 samples, 2 packets per batch, and FPGA running at 100MHz. 2.49GHz is used to avoid external interference from campus network. We modify the default GNU Radio OFDM implementation to realize all the functionalities described in Section IV. We support Reed-Solomon based FEC. We use a combination of Schmidl-Cox auto-correlation algorithm [26] followed by an additional cross-correlation step, which cross correlates the incoming signal with a PN sequence corresponding to a preamble, to detect a correlation peak and improve the accuracy of preamble detection. We run the experiments using BPSK and QPSK modulations under varying levels of FEC.

Since implementing carrier sense in USRPs involves significant modification to the FPGA and is orthogonal to our work, for simplicity, in the testbed implementation, we use time division multiple access (TDMA) as the medium access mechanism. All nodes are scheduled by an external scheduler and only transmit during the assigned time slot to avoid interference. To accommodate USRP significant turn around time, the source is rate limited so that relay nodes have enough time to switch between transmission and receiving modes. Our experiments run in real time. Our routing protocol can work with other MAC and our Qualnet simulation uses IEEE 802.11.

Synchronization accuracy: To measure this accuracy, we let 3 transmitters, T_1 , T_2 , and T_3 send to one, two, or three randomly placed receivers R . We measure the propagation and turnaround delays of the node using probes. T_1 transmits packets to R at a fixed rate. After T_1 transmits its preamble and header, it waits for time δ for T_2 to join. Similarly, T_2 waits for T_3 to join after δ time. R then counts the number of samples (gap) between the received preambles from T_1 , T_2 , and T_3 . Figure 4(a) shows synchronization errors when 3 senders transmitting to 1, 2, or 3 receivers. The maximum misalignment is 3 samples for 1 receiver, 4 samples for 2,

| Topology | Mod.+FEC | PDR | SDR | Mod.+FEC | PDR | SDR |
|----------|------------------------|----------------|--------------|------------------------|----------------|--------------|
| (i) | BPSK,3/4 QPSK,1/2 | 638.3 512.9 | 0.97 0.97 | BPSK,7/8 QPSK,3/4 | 551.3 501.0 | 0.97 0.97 |
| (ii) | BPSK,3/4 QPSK,1/2 | 715.2 657.2 | 0.97 0.95 | BPSK,7/8 QPSK,3/4 | 615.1 623.6 | 0.98 0.95 |
| (iii) | BPSK, 3/4 QPSK, 1/2 | 881.4 869.1 | 0.98 0.97 | BPSK, 7/8 QPSK, 3/4 | 844.7 831.1 | 0.98 0.96 |
| (iv) | BPSK, 3/4 QPSK, 1/2 | 975.0 963.2 | 0.98 0.96 | BPSK, 7/8 QPSK, 3/4 | 932.1 915.2 | 0.98 0.96 |
| (v) | BPSK, 3/4 QPSK, 1/2 | 940.9 929.1 | 0.98 0.96 | BPSK, 7/8 QPSK, 3/4 | 915.2 884.1 | 0.97 0.96 |

TABLE II. PACKET DELIVERY RATE (PDR) AND RAW SYMBOL DELIVERY RATE (SDR) IN OUR USRP TESTBED.

and 6 samples for 3 receivers, well under the CP length of 24 samples. An offset in the time domain results in a phase shift in the frequency domain, which is reflected in both preamble and data symbols and get canceled during the channel compensation step. A synchronization error within CP does not affect decoding accuracy. For larger propagation delays, we may increase CP by using a larger FFT window without increasing CP overhead.

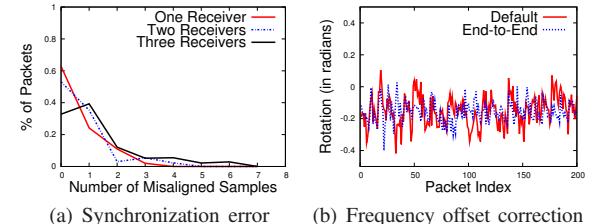


Fig. 4. Micro benchmarks.
Frequency offset correction: Next we quantify the accuracy of end-to-end frequency offset correction. We select two random nodes A and B , and measure the rotation due to the residual offset between them experienced over 150 packets. Then we add two hops between A and B , and measure the rotation using our end-to-end frequency correction. Figure 4(b) shows these two approaches yield similar estimations with an average difference of 0.067 radians.

Testbed performance: We construct the following five multihop topologies by carefully placing nodes to have the required numbers of hops: (i) Figure 1, with source and destination having two antennas each, and spatial multiplex (SM) taking place every hop, (ii) Figure 1, each node with one antenna and spatial multiplex (SM) taking place on the second hop, (iii) Figure 5 with the source having two antennas and SM taking place on the first hop, (iv) Figure 5 with the destination having two antennas and SM taking place on the second hop, (v) Figure 5 with both the source and destination having two antennas and SM taking place on both hops. These topologies are interesting beyond their current forms, since they may serve as building blocks to compose a larger network.

In each case, the source sends 500 packets and uses a batch size of 2, BPSK and QPSK with varying coding rates. The SNR on the various hops ranges between 18dB and 26dB. Table II reports the average across five runs. Both PDR and SDR are high in all cases. The destination can sometimes

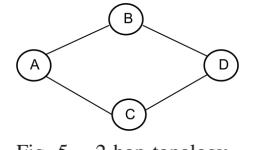


Fig. 5. 2-hop topology.

decode more packets correctly than what the source transmits, because it is possible to extract two native symbols from one transmitted coded symbol when the rate is low. Our implementation also supports multiple flows. We evaluate two flows in some of the topologies, and observe similar decoding rates since each flow is routed independently. For example, using BPSK-7/8, PDR and SDR in topology (iii) are 841.07 and 0.97, respectively, and they are 929.3 and 0.98 in topology (iv), respectively. This demonstrates the feasibility of DM^+ .

VI. SIMULATION

We implement DM^+ in QualNet v3.9.5 using 802.11 MAC. We compare with the following routing protocols: (i) Shortest-path routing (SPP) using the ETX routing metric, which minimizes the total number of expected transmissions from a source to its destination [6]; (ii) Shortest-path routing with rate-limiting (SPP-RL), the same as (i) except the flows' sending rates are optimized using the conflict graph interference model, as described in [16]; (iii) MORE, a state-of-the-art opportunistic routing protocol; and (iv) Optimized opportunistic routing (optimized OR), which improves MORE by optimizing opportunistic routing and rate limiting. The objective function maximizes total throughput. All protocols except SPP and SPP-RL use the batch size of 12 packets, which yields similar performance as a larger batch size.

We consider the following topologies: (i) 5x5 grid topologies, (ii) 25-node random topologies, (iii) Roofnet topology with 35 nodes [24], (iv) UW testbed topologies with 14 nodes [23]. Topologies (i) and (ii) use IEEE 802.11a and occupy $500 \times 500m^2$ area. Roofnet uses an IEEE 802.11b testbed and UW traces contain measurements from 802.11a and 802.11b testbeds. For UW topologies, we report the results from 802.11b since the results from 802.11a are similar. In IEEE 802.11a, every sender uses transmission power of $10dBm$ (QualNet default), and a fixed rate of 6 Mbps. This gives $230 m$ communication range, $253.6 m$ carrier sense range, and $460 m$ interference range, which we use in the interference model to determine interfering links. In IEEE 802.11b, every sender uses transmission power of $15dBm$, and a fixed rate of 1 Mbps. This gives $1012.3 m$ communication range, $1090.5 m$ carrier sense and interference range. The senders generate 1024-byte CBR traffic and always have traffic to send.

For each scenario, we report the average and standard deviation of 10 random trials and vary the number of concurrent flows from 1 to 8. In each trial, flow sources and destinations are picked randomly provided they are at least 2 hops away, as the performance of all schemes should be the same for 1-hop flows. The reported throughput accounts for the overhead in each protocol (*i.e.*, header and trigger overhead in DM^+ and header overhead in the other protocols). Since all these protocols are designed for wireless networks with lossy links, we extend QualNet to generate directional inherent packet losses ranging 0-50%. We also vary the maximum link loss to understand its effects.

Unless otherwise specified, all nodes have one antenna. In order to further understand how the performance varies under heterogeneous numbers of antennas, we also consider all nodes have 1 antenna, but the sources and destinations have multiple antennas, and we vary the number of antennas we place on

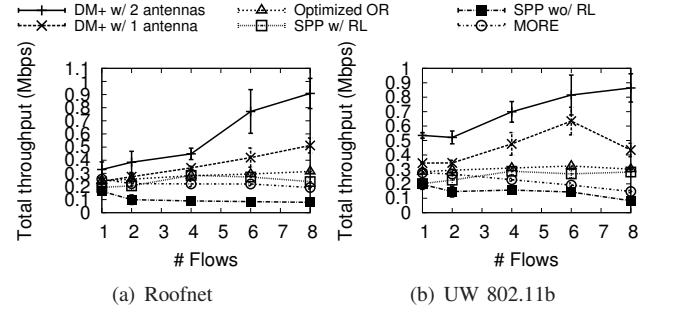


Fig. 6. Total throughput under a varying # flows.

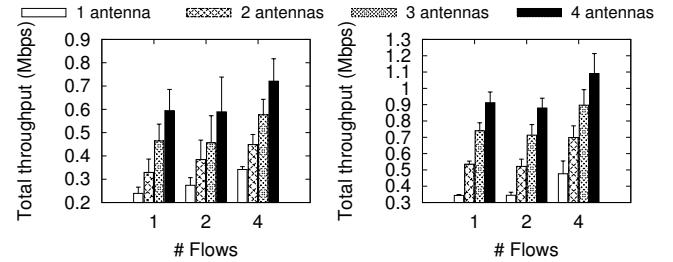


Fig. 7. Total throughput under a varying # antennas.

the sources and destinations. This is realistic since Internet gateways in a mesh network typically serve as sources or destinations of the flows and are appropriate to have multiple antennas, while clients with lots of traffic to send also have incentives to deploy multiple antennas.

Varying number of flows: As shown in Figure 6, DM^+ with 2 antennas > DM^+ with 1 antenna > optimized OR > SPP with RL > MORE and SPP without RL. DM^+ with 2 antennas outperforms SPP without RL by 437-530%, SPP with RL by 138-170%, MORE by 168-241%, and optimized OR by 100-126% using spatial multiplexing. DM^+ with 1 antenna out-performs SPP without RL by 232-290%, SPP with RL by 51-74%, MORE by 67-115%, and optimized OR by 27-45%. DM^+ outperforms the others by leveraging spatial multiplexing. DM^+ with 2 antennas performs best due to more opportunities for spatial multiplexing.

Varying number of antennas: Figure 7 shows the results from real topologies as we vary the number of antennas at the sources and destinations while keeping the other nodes to have 1 antenna. As expected, the performance improves with the number of antennas at the source and destination due to higher concurrency. DM^+ with 4 antennas leads to 20%-23% improvement over 3 antennas, 54%-64% improvement over 2 antennas, and 116%-153% over 1 antenna cases. Moreover, DM^+ with 4 antennas also out-performs SPP without RL by 462%-484%, SPP with RL by 172%-306%, MORE by 165%-275%, and optimized OR by 134%-222%. Similar results are observed from synthetic topologies and omitted for brevity.

Proportional fairness: This objective is non-linear. In order to optimize it, we approximate it using a piecewise linear, increasing, convex function. We select s points on $\log(x)$, and approximate $\log(x)$ using s line segments, each connecting two adjacent points. We perform two different point selections and observe similar performance. In the interest of space,

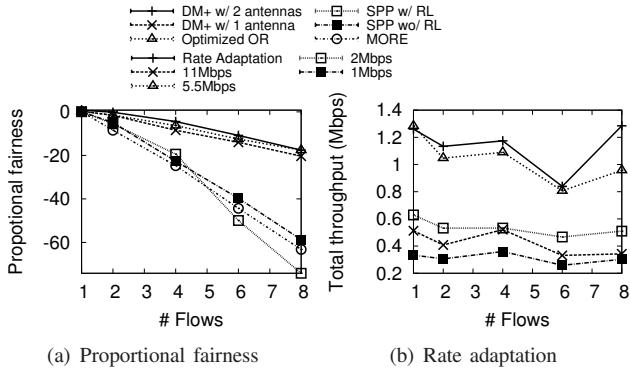


Fig. 8. Proportional fairness and rate adaptation in random topologies under a varying number of flows.

Figure 8(a) plots the proportional fairness in random topologies from one selection: $x = 0.001, 0.01, 0.1, \sqrt{0.1}, 1, \sqrt{10}, 10$. As we can see, DM^+ with 2 antennas achieves the highest proportional fairness, while DM^+ with 1 antenna and optimized opportunistic routing achieves comparable fairness, both outperforming all the other schemes. This shows the optimization framework is flexible in supporting different objectives, including total throughput and fairness.

Rate Adaptation: Figure 8(b) compares the performance of DM^+ with rate adaptation against the fixed rates version. As it shows, our rate adaptation out-performs any of the fixed rate by letting each physical link use most appropriate rate(s) for each flow and different links may use different rates.

VII. CONCLUSION

We develop a distributed MIMO routing protocol. It consists of a novel optimization framework that jointly optimizes spatial multiplexing, routing, and rate limiting, and a new protocol that enforces the optimized MIMO routes, synchronizes transmissions from different senders, encodes and decodes analog signals to support distributed spatial multiplexing, and compensates for the frequency offset incurred over multiple hops. To our knowledge, it is the first distributed MIMO routing protocol/prototype in real multi-hop networks. Our evaluation shows its significant gain in various scenarios including single and multiple flows. For example, on average DM^+ with 2 antennas out-performs SPP without RL by 603%, SPP with RL by 301%, MORE by 769%, and optimized OR by 115%.

Acknowledgements: This work is supported in part by NSF Grants CNS-1343383 and CNS-1017549. We thank reviewers especially Aaron Striegel for their insightful feedback.

REFERENCES

- [1] 5G Mesh Internet Services. <http://www.5gmesh.com>.
- [2] BHATIA, R., AND LI, L. Throughput optimization of wireless mesh networks with MIMO links. In *Proc. of IEEE INFOCOM* (2007).
- [3] BISWAS, S., AND MORRIS, R. ExOR: opportunistic multi-hop routing for wireless networks. In *Proc. of ACM SIGCOMM* (Aug. 2005).
- [4] CHACHULSKI, S., JENNINGS, M., KATTI, S., AND KATABI, D. Trading structure for randomness in wireless opportunistic routing. In *Proc. of ACM SIGCOMM* (Aug. 2007).
- [5] CHU, S., AND WANG, X. MIMO-aware routing in wireless mesh networks. In *Proc. of IEEE INFOCOM* (2010).
- [6] COUTO, D. D., AGUAYO, D., BICKET, J., AND MORRIS, R. A high-throughput path metric for multi-hop wireless routing. In *Proc. of ACM MobiCom* (Sept. 2003).
- [7] DRAVES, R., PADHYE, J., AND ZILL, B. Routing in multi-radio, multi-hop wireless mesh networks. In *Proc. of ACM MobiCom* (2004).
- [8] DUONG, T. Q., AND ZEPERNICK, H. J. Performance analysis of cooperative spatial multiplexing with amplify-and-forward relays. In *Proc. of PIMRC* (2009).
- [9] GUPTA, P., AND KUMAR, P. R. The capacity of wireless networks. *IEEE Transactions on Information Theory* 46, 2 (Mar. 2000).
- [10] HO, T., KOETTER, R., MEDARD, M., KARGER, D. R., AND EFFROS, M. The benefits of coding over routing in a randomized setting. In *Proc. of IEEE Symposium on Info. Theory* (2003).
- [11] JAAFAR, W., AJIB, W., AND TABBANE, S. The capacity of MIMO-based wireless mesh networks. In *Proc. of ICON* (2007).
- [12] JAIN, K., PADHYE, J., PADMANABHAN, V. N., AND QIU, L. Impact of interference on multi-hop wireless network performance. In *Proc. of ACM MobiCom* (Sept. 2003).
- [13] KATTI, S., GOLLAKOTA, S., AND KATABI, D. Embracing wireless interference: Analog network coding. In *Proc. of SIGCOMM* (2007).
- [14] KATTI, S., RAHUL, H., HU, W., KATABI, D., MEDARD, M., AND CROWCROFT, J. XORs in the air: Practical wireless network coding. In *Proc. of ACM SIGCOMM* (Sept. 2006).
- [15] KORGER, U., AND HARTMANN, C. Spatial multiplexing for heterogeneously equipped nodes in wireless ad hoc networks. In *VTC* (2011).
- [16] LI, Y., QIU, L., ZHANG, Y., MAHAJAN, R., AND ROZNER, E. Predictable performance optimization for wireless networks. In *Proc. of SIGCOMM* (2008).
- [17] MARIC, I., GOLDSMITH, A., AND MEDARD, M. Multihop analog network coding via amplify-and-forward: The high SNR regime. *IEEE Transactions on Information Theory* (Feb. 2012).
- [18] MURPHY, P., SABHARWAL, A., AND AAZHANG, B. On building a cooperative communication system: testbed implementation and first results. *Special issue on coop. comm. in wireless networks* (2009).
- [19] OGGIER, F., AND HASSIBI, B. A coding scheme for wireless networks with multiple antenna nodes and no channel informations. In *Proc. of ICASSP* (2007).
- [20] OZGUR, A., LEVEQUE, O., AND TSE, D. Hierarchical cooperation achieves optimal capacity scaling in ad-hoc networks. *Transactions on Information Theory* (2007).
- [21] RAHUL, H., HASSANIEH, H., AND KATABI, D. SourceSync: a distributed wireless architecture for exploiting sender diversity. In *Proc. of ACM SIGCOMM* (Aug. 2010).
- [22] RANDUNOV, B., AND BOUDEC, J. Y. L. Rate performance objectives of multihop wireless networks. In *Proc. of IEEE INFOCOM* (2004).
- [23] REIS, C., MAHAJAN, R., RODRIG, M., WETHERALL, D., AND ZAHORJAN, J. Measurement-based models of delivery and interference. In *Proc. of ACM SIGCOMM* (2006).
- [24] MIT Roofnet. <http://www.pdos.lcs.mit.edu/roofnet/>.
- [25] ROZNER, E., HAN, M. K., QIU, L., AND ZHANG, Y. Accurate model-driven optimization of opportunistic routing. In *Proc. of ACM SIGMETRICS* (2011).
- [26] SCHMIDL, T., AND COX, D. Robust frequency and timing synchronization for ofdm. In *IEEE Trans. Communications* (Dec. 1997).
- [27] SRINIVASAN, K., JAIN, M., CHOI, J. I., AZIM, T., KIM, E. S., LEVIS, P., AND KRISHNAMACHARI, B. The kappa factor: Inferring protocol performance using inter-link reception correlation. In *Proc. of ACM MobiCom* (2010).
- [28] TAN, K., FANG, J., ZHANG, Y., CHEN, S., SHI, L., ZHANG, J., AND ZHANG, Y. Fine grained channel access in wireless LAN. In *Proc. of ACM SIGCOMM* (Aug. 2010).
- [29] YAN, H., MALTZ, D. A., NG, T. S. E., GOGINENI, H., AND ZHANG H. Tesseract: A 4D network control plane. In *Proc. of NSDI* (2007).
- [30] YU, X., MORAES, R., SADJADPOUR, H., AND GARCIA-LUNA-ACEVES, J. Capacity of MIMO mobile wireless ad hoc networks. In *Proc. of IEEE WirelessCom* (2005).
- [31] ZHANG, Y., WANG, G., AND AMIN, M. G. Cooperative spatial multiplexing in multi-hop wireless networks. In *ICASSP* (2006).
- [32] ZTE releases first 5G architecture based on dynamic mesh. http://www.zte.com.cn/en/press_center/news/201406/t20140623_425167.htm.