BBN: Throughput Scaling in Dense Enterprise WLANs with Blind Beamforming and Nulling

Wenjie Zhou†, Tarun Bansal†, Prasun Sinha and Kannan Srinivasan
Department of Computer Science and Engineering, The Ohio State University
2015 Neil Avenue, Columbus, OH 43210
{zhouwe, bansal, prasun, kannan}@cse.ohio-state.edu
†Co-primary Authors

ABSTRACT

Today's Enterprise Wireless LANs are comprised of densely deployed access points. This paper proposes BBN, an interference nulling scheme that leverages the high density of access points to enable multiple mobile devices to transmit simultaneously to multiple access points (APs), all within a single collision domain. BBN also leverages the capability of the APs to communicate with each other on the wired backbone to migrate most of the decoding complexity to the APs, while keeping the design at the mobile clients simple. Finally, we leverage the static nature of the access points to make BBN more practical in networks where the mobility of clients inhibit the use of traditional interference alignment schemes. We implement a prototype of BBN on USRP testbed showing its feasibility. The experiment results show that BBN provides a throughput gain of 1.48× over omniscient TDMA. Results from our trace-driven simulations show that BBN obtains a throughput of up to 5.6× over omniscient TDMA.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication, Network communications

Keywords

Enterprise Networks; Blind Beamforming and Nulling;

1. INTRODUCTION

The recent explosive growth in the number of mobile devices and the data generated by these devices has led to a decrease in the channel resources available to each individual device. Network administrators have tried to tackle this problem by densely deploying access points so that users can almost always find a closeby AP with good signal strength. However, dense deployment of APs does not scale well with the throughput demands. In the existing network protocols [18, 24], when one mobile client is transmitting uplink packets to an access point, the nearby clients have to remain silent to avoid interference to the ongoing transmission.

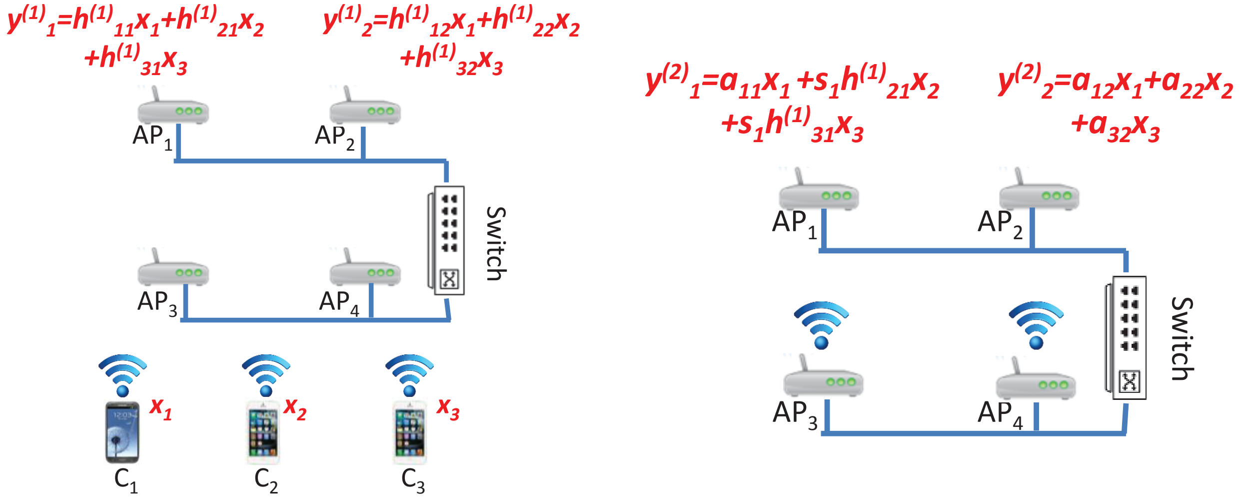
Recently, multiple algorithms have been proposed that help in scaling the throughput with number of wireless devices. **Interference Alignment (IA)** [7] is one of such techniques that requires clients to participate in a schedule with exponential number of slots. However, mobile clients are really mobile. They may not stay at the same place for a long time. **Multi-User MIMO (MU-MIMO) [9]** enables scaling of throughput with number of devices, but it requires APs to exchange samples over the backbone. Although, the wired backbone in Enterprise Wireless LANs (EWLANs) is underutilized [11, 6], exchanging samples requires significantly higher bandwidth compared to exchanging packets which cannot be supported by current wired networks [11, 12]. **Joint beamforming** based algorithms such as [15, 21] work only for the downlink traffic. To perform joint beamforming, these algorithms require all transmitters to share the contents of all packets to be transmitted. However, mobile devices are not connected through a wired backbone, and are unable to share the packets amongst each others.

This paper proposes BBN, the first implementation of Blind Beamforming and Nulling scheme that enables multiple nearby access points to concurrently receive uplink packets from multiple mobile clients, all within a single collision domain without overwhelming the backbone. BBN does not increase energy consumption on the clients compared to 802.11 and executes exactly over two time slots. BBN leverages three properties that are unique to EWLANs: (i) *Dense deployment of APs (See Fig. 3 and [18])*; (ii) *Capability of these APs to exchange packets with each other over the underutilized wired backbone*; and, (iii) *Immobility of APs resulting in relatively stationary channels* (See Fig. 2). When an AP is receiving uplink data, existing algorithms [18] including IEEE 802.11 WiFi, suppress nearby APs to transmit or receive data. In contrast, BBN makes use of the energy-rich access points to assist their clients (mobile devices) in decoding their packets at their respective access points. In BBN, the clients only participate in the first slot and the access points participate for the clients in the second slot.

Consider the example enterprise WLAN shown in Fig. 1(a) where all the APs and the three clients are in a single collision domain. Assume that the three users want to upload one packet each to the backbone. An omniscient TDMA scheduling algorithm with global knowledge would require three time slots to complete this upload. In BBN, in the first slot as shown in Fig. 1(a), all users will transmit at

MobiCom'14, September 7-11, 2014, Maui, Hawaii, USA.

http://dx.doi.org/10.1145/2639108.2639113 .



(a) First slot. x_1, x_2 and x_3 are the three packets transmitted by C_1, C_2 and C_3, respectively. $y_i^{(1)}$ are the received samples at AP_i in the first slot. $h_{ij}^{(1)}$ is the channel from client i to AP_j during time slot 1.

(b) Second slot. A subset of APs (AP_3 and AP_4) transmit in the second slot while the rest of the APs receive. $y_i^{(2)}$ are the received samples at AP_i in the second slot. a_{ij} are the combined channel coefficients of client i at AP_j after the transmissions of the second slot. s_i is the scaling coefficient at AP_i compared with first slot.

Figure 1: Illustration of BBN over a topology of 3 clients and 4 APs. All devices belong to the same collision domain and can hear each other. More details about the expressions are discussed in Section 2.

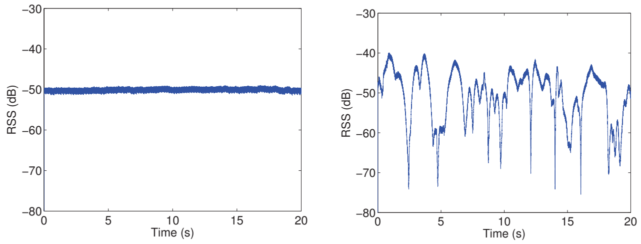
the same time. All the 4 APs will receive a combination of three transmitted packets. In the second slot, AP_3 and AP_4 will retransmit the received signals by first precoding [13] them such that the following condition is satisfied as shown in Fig. 1(b): At AP_1, samples corresponding to x_2 and x_3 in the second slot align with the samples corresponding to x_2 and x_3 in the first slot. Decoding happens in multiple steps as follows:

1. At the end of the second slot, AP_1 scales the samples received by AP_1 in the second slot and subtracts them from the samples received in the first slot. This scaling is done such that samples corresponding to x_2 and x_3 are *nulled*. Afterwards, it is left with only the samples corresponding to x_1. AP_1 decodes the samples to obtain the packet transmitted by C_1. Next, it transmits the decoded packet over the backbone to AP_2.
2. AP_2 recreates the samples corresponding to x_1 and subtracts them from the samples received in the first slot and the second slot.
3. After subtraction, AP_2 is left with two equations (one from each slot), and two variables (x_2 and x_3). AP_2 solves the two equations to obtain x_2 and x_3.
4. Afterwards, AP_1 and AP_2 forward x_1, x_2 and x_3 towards their destinations.

BBN enables the three transmitters with single antenna to upload three packets in two slots, improving the throughput by 50% compared to omniscient TDMA. In Section 2, we show that in networks with high enough density of APs, BBN enables N mobile clients to transmit N uplink packets in exactly two slots resulting in unbounded throughput. Also, note that BBN requires the APs to exchange only the decoded packets instead of the raw samples. Compared with our previous work RobinHood [5], which works only in a single collision domain, BBN supports multiple collision domain and is more robust to decoding failures.

The focus of BBN is to increase throughput of the uplink traffic for clients with a single antenna. This is in contrast with [21, 15] that focus on downlink traffic. Recently, uplink traffic [12, 6] has been growing at a fast rate due to the emergence of a wide-range of applications, such as cloud computing, video conferencing, online gaming, VoIP, and traffic generated from mobile devices (e.g., location information or sensor readings). BBN makes extensive use of the wired backbone. Besides transmitting the decoded packets, the channel state information, which are required to do nulling in the second slot, are also exchanged over the backbone. Since BBN migrates most of the complexity from the mobile devices to the APs, it allows BBN to work even when the channel from clients to APs is rapidly changing due to client mobility. BBN works as long as the APs are time-synchronized with each other and it places very few requirements on the clients. This paper makes the following contributions:

1. We propose a blind beamforming and nulling scheme, BBN, that scales uplink throughput with the number of access points. BBN also works over multiple collision domains.
2. This paper shows the first implementation of blind beamforming and nulling on USRP radios. Experiments performed on our testbed show that BBN achieves 1.48× throughput compared to omniscient TDMA.
3. Trace-driven simulation results show that in a large Enterprise WLAN, BBN can leverage the density of



(a) Channel between a pair of APs

(b) Channel between a mobile client and an AP

Figure 2: Received Signal Strength (RSS) in an office environment. The channel between APs is relatively stationary compared to channel between AP and mobile client.

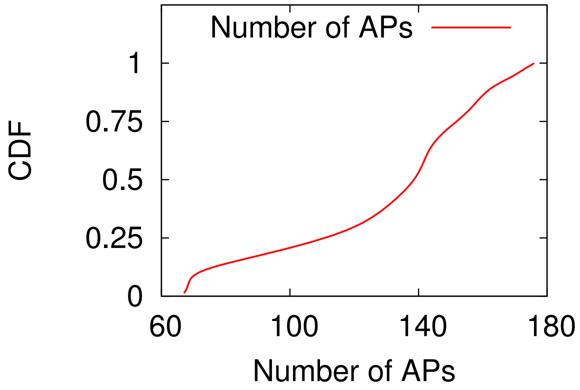


Figure 3: CDF of number of APs observed across different locations. The data was collected at multiple places including a hospital, a large university library and an apartment complex.

the access points. In EWLANs with high density of APs, BBN provides a throughput of 5.6× compared to omniscient TDMA and 52.4× compared to IEEE 802.11.

2. ILLUSTRATION

Before discussing BBN in detail, we define a few notations. All of the clients and APs in BBN are assumed to have only one antenna. The network consists of clients C_1, C_2 and C_3 and four APs from AP_1 to AP_4 that are connected through a wired backbone. Let $h_{ij}^{(1)}$ be the channel coefficient between C_i and AP_j in slot 1. In the second slot, a subset of APs are selected to transmit. For this example, this set consists of AP_3 and AP_4. Let $h_{kj}^{(2)}$ be the channel coefficient between AP_k and AP_j in slot 2. In this section, we assume that all the wireless devices are in single collision domain (*i.e.*, they can all hear each other). In Section 5, we extend BBN to networks with multiple collision domains. Let x_i be the packet sent by C_i in slot 1. In the following discussion, we ignore the presence of noise since it is not possible to null the noise. However, we do take noise into account in our analysis (See Section 4.4) and then later in our simulations (Section 7). Let $y_{ik}^{(t)}$ be the component of x_i received by AP_k in slot t. We have:

$$y_{ik}^{(1)} = h_{ik}^{(1)} x_i \tag{1}$$

Let v_k be the precoding vector for AP_k in the second slot and M be the total number of APs (In this example, we have $M = 4$). Let $y_{ij}^{(2)}$ be the component of x_i received by AP_j in slot 2. We have:

$$y_{ij}^{(2)} = \sum_{k=3}^{M} h_{kj}^{(2)} v_k y_{ik}^{(1)} = \sum_{k=3}^{M} h_{kj}^{(2)} v_k h_{ik}^{(1)} x_i \tag{2}$$

We want to ensure that components of x_2 and x_3 at AP_1 are a linear combination of their components in the first slot. Let s_i be the scaling coefficient at AP_i. Thus,

$$y_{21}^{(2)} = \sum_{k=3}^{M} h_{k1}^{(2)} v_k h_{2k}^{(1)} x_2 = s_1 y_{21}^{(1)} = s_1 h_{21}^{(1)} x_2 \tag{3}$$

$$y_{31}^{(2)} = \sum_{k=3}^{M} h_{k1}^{(2)} v_k h_{3k}^{(1)} x_3 = s_1 y_{31}^{(1)} = s_1 h_{31}^{(1)} x_3 \tag{4}$$

Simplifying these equations, we get

$$\sum_{k=3}^{M} h_{k1}^{(2)} v_k h_{2k}^{(1)} - s_1 h_{21}^{(1)} = 0 \tag{5}$$

$$\sum_{k=3}^{M} h_{k1}^{(2)} v_k h_{3k}^{(1)} - s_1 h_{31}^{(1)} = 0 \tag{6}$$

Since, the right sides of Eqs. 5 and 6 are all 0, instead of 2, at least 3 variables are required to obtain non-zero solutions. One of these variables is the scaling coefficient (s_1). Thus, a total of 2 transmitting APs are required to supply these variables. Further, two receiving APs are also required such that the first AP decodes x_1 while the second AP decodes x_2 and x_3. Thus, in total $M = 2 + 2 = 4$ APs are required to support 3 clients as in Fig. 1.

In BBN, for the network shown in Fig. 1, at the end of slot 1, AP 3 and AP 4 solve Eqs. 5 and 6 to obtain precoding vectors which are then used during slot 2 (See Eq. 2). This computation may take time (due to communication among APs over the backbone). In general wireless networks, this creates inaccuracies since the channel between APs and the mobile clients may change from the time the channel state information (CSI) was measured to the time when the APs retransmit the data in the second slot. Thus, the precoding vectors that were computed based on old CSI may not be suitable for the channel's current state. This may lead to inaccurate beamforming and nulling. However, in BBN, the mobile clients do not participate in the second slot. Only the APs transmit and receive data in the second slot. Due to the immobile nature of the APs, the channel (or CSI) between APs changes very slowly (See Fig. 2(a)). Thus, the CSI computed among APs is valid for longer duration compared to CSI between mobile clients and APs. *By requiring only the APs to transmit in the second slot, BBN ensures higher accuracy of joint beamforming and joint nulling.*

Number of APs required: In general, if there are N clients in the network, then BBN needs to align $(N-1)$ packets at the first AP, $(N-2)$ packets at the second AP and so on. Thus, a total of at least $(N-1)+(N-2)+\cdots+2 = \frac{N^2-N-2}{2}$ variables are required to satisfy all the constraints. However, to obtain a non-zero solution, we need to include one extra AP, *i.e.* a total of $\frac{N^2-N}{2}$ APs. However, $N-2$ of the variables are supplied by the scaling coefficients at the receiving APs. Thus, a total of $\frac{N^2-N}{2}-(N-2)=\frac{N^2-3N+4}{2}$ transmitting APs are required. Finally, $N-1$ receiving APs are also required in slot 2, where the first $N-2$ receiving APs decode one unique packet while the last AP decodes 2 packets. Therefore, with $\frac{N^2-3N+4}{2}+N-1=\frac{N^2-N+2}{2}$ APs, BBN can leverage this high density of APs to decode N uplink packets in exactly two slots. Further, in contrast to [5], BBN requires N fewer APs.

3. CHALLENGES

Note that when the APs (*i.e.*, AP_3 and AP_4) in slot 2 transmit, they have to align the samples of x_2 and x_3 at AP_1. To achieve this, they precode the signals that they received in the first slot and transmit. However, in contrast to the existing solutions [21], in BBN, the transmitting APs are not aware of what they are transmitting (since they are unable to decode the samples received in the first slot). We call this *Blind Beamforming and Nulling.* Although the idea behind BBN is simple, there are multiple challenges that need to be handled to make it practical.

1. **Oblivious to the contents of the transmitted signal:** The APs transmitting in slot 2 are not aware of the contents of the signals transmitted in slot 2. Despite this, they need to cancel out (or align) the different contents of the signal at different receiving APs.

2. **Synchronization**: In order for the APs transmitting in slot 2 to align their signals at the receiving APs, these transmitting APs are required to be synchronized at the sample level. This requirement is similar to the requirements of the other existing algorithms that focus on downlink traffic [21, 15, 20]. Observe that BBN does not impose synchronization requirement on the mobile clients.

3. **Multi-collision domain:** The previous discussion assumes that all clients and all APs can hear each other directly. However, this may not be true for large scale EWLANs. Thus, we need a mechanism to extend BBN to such networks.

4. **Inconsistency in the AP density:** To decode N packets, BBN requires $\frac{N^2-N+2}{2}$ access points nearby. However, the actual number of APs present may be higher or lower than this number. If the number of available APs is higher, then BBN can make use of all of them. On the other hand, if the number of available APs is smaller, than a mechanism is required to select a subset of the clients.

5. **Robustness:** Unlike downlink [21], where each client individually decodes its own packet, in BBN, decoding happens in a cascading fashion. Decoding of a packet depends on the successful decoding of the previous packets. Clearly, in such a design, failure in decoding of one packet, makes all future decodings unsuccessful. We need a new mechanism to increase the robustness of the decoding.

How we handle these challenges are explained in the next two sections.

4. PHYSICAL LAYER DESIGN

In this section, we explain the physical layer working of BBN using three different phases. First, we explain how multiple clients transmit simultaneously to the APs and how the channel state information between clients and APs is estimated. Then, we show how the APs conduct blind-beamforming and nulling without knowing the contents of the transmitted signals. Finally, the decoding process is explained. In BBN, the clients participate in only the first phase while the APs participate in all the three phases.

4.1 Phase I: Client transmission

As explained in Section 2, the transmissions in BBN are divided into two slots. In the first slot, the clients transmit concurrently to the APs. Besides the received combined samples from the clients to APs, the channel state information (CSI) between all the clients and APs is also computed in this phase. To obtain the CSIs, each client sends an access code (or unique PN sequences [17] assigned to each client) that is free of interference.

The transmission timeline of Phase I is shown in Fig. 4. First, the APs broadcast an *approve* message. This message contains the IDs of the clients that are allowed to transmit in this slot (For more details on how the APs select the subset of clients, refer to Section 5 that describes the MAC design of BBN). The relative order of the IDs determines the time when a client should transmit its access code. Since the clients are not synchronized, the transmission of access codes may partially overlap with each other due to wireless propagation delay. To avoid this overlap, a small time gap, called inter-access-code-space (IACS), is inserted between the transmissions. Finally, after the transmission of access codes, the clients transmit their packets simultaneously. All the APs compute the CSI from different clients using the interference-free access codes and also store the received samples corresponding to the data packets. In our experiments and simulations, we set the duration of IACS to $2\mu s$, which is enough to compensate for the propagation delay if the maximum distance between the clients and APs is no more than 300 meters.

To conduct blind-beamforming, besides the CSIs between the clients and APs, the CSIs between the transmitting APs and receiving APs are also required. As shown in Fig. 4, all of the APs broadcast their access codes one after the other. When one AP broadcasts, all other APs can estimate the CSI from that AP. The estimated CSIs along with the CSIs between clients and APs are forwarded to a *group-head AP* through the wired backbone network. The head AP, uses these CSIs to compute the best sets of transmitting APs, the set of receiving APs, the decoding order, and the precoding vector to be used by each of the transmitting AP. This information is then sent back by the group-head AP to every AP in the group. In Fig. 5, AP_3 and AP_4 are selected as the transmitting APs.

This computation at the group-head AP and the distribution of result back to APs may take some time due to delays over the wired backbone. To ensure that all APs have received the computed results back from the group-head AP, BBN requires all APs to wait for Backbone-Inter-Frame-Space (BIFS) duration.

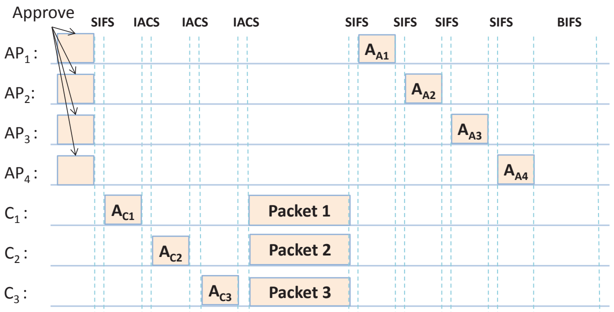


Figure 4: Phase I time-line: A_{Ci} and A_{Aj} represent the access codes for C_i and AP_j, respectively.

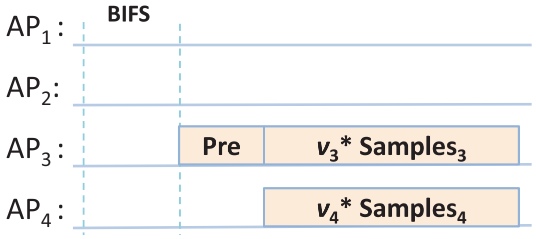


Figure 5: Phase II time-line: v_i denotes the precoding vector of AP_i.

4.2 Phase II: Blind-beamforming

After waiting for BIFS time, all APs multiply the samples received in the first slot with their precoding vectors and retransmit them (See Fig. 5). The value of BIFS can be selected on the basis of the speed of the Ethernet and the expected delays involved. To avoid wastage of wireless channel during BIFS, APs in BBN participate in another set of communication (*e.g.*, downlink traffic) while waiting to hear back from the group-head AP. Observe that since the APs are relatively stationary, the precoding vectors computed by group-head AP are valid for a long duration as described in Section 2. Further, due to the relatively stationary channel among APs, we do not need to frequently measure the channel among APs which further reduces the overhead incurred during Phase I. The short packet **Pre** sent by AP_3 is a sequence known to all of the APs. The purpose of sending this sequence is two fold: 1.) it can be viewed as a preamble for the receiving APs to detect the correct start point of the retransmission; 2.) it can be used to estimate the sampling offset between the transmitting APs and receiving APs using the same techniques as in packet subtraction [6, 10].

4.3 Phase III: Decoding Packets

In BBN, the packets are decoded in a sequential order. The first AP decodes one packet and sends it to the next AP which, upon receiving the packet, recreates the received samples, and then subtracts those samples from the received samples. The remaining samples are decoded to obtain the second packet. This process is continued until all packets have been decoded. Performing a successful subtraction requires estimating various offsets such as frequency offset, sampling offset, and phase offset. Once the offsets have been estimated, the AP needs to recreate the received samples. This sequential decoding and packet subtraction have been well-studied in the literature [6, 10]. We refer the reader to the existing literature.

Another practical issue to note is that there is sampling offset between the transmitting APs and the receiving APs in the second slot. This offset makes it difficult to align the components of x_2 and x_3 received by AP_1 in the second slot with the corresponding components received in the first slot. To that end, the packet **Pre** as explained in the previous section can be used to estimate the offset.

4.4 Computing the Packet Decoding Order

In the previous discussion (Sec. 2), we assumed that x_1 is decoded first, followed by x_2 and x_3. We also assumed that joint precoding leaves no residual noise. However, in practice, joint precoding and packet subtraction are not perfect and leave some residual noise. Thus, in this section, we compute the optimal order in which packets should be decoded such that the decoding accuracy is maximized in the presence of the residual noise. To determine the optimal decoding order, we need to compute the expected received signal strength (RSS) of each packet (say x_i) at each AP (say AP_j). The exact value of RSS depends on the precoding vectors which in turn depend on the rest of the matching. This makes the problem combinatorial in nature.

We compute the expected RSS of client i at AP_j using a heuristic. In the second slot, let AP_N to AP_M be the set of transmitting APs and AP_1 to AP_{N-1} be the set of receiving APs. Observe that in the second slot, AP_j receives components of x_i that have been retransmitted by all APs in the range AP_N to AP_M. Thus, components of x_i arrive at AP_j through $M-N+1$ different paths. Each of these $M-N+1$ paths start at client i, pass through some transmitting AP (say AP_k) and end at AP_j. Further, each of these paths consist of two links: First from C_i to AP_k and, second from AP_k to AP_j. We say that RSS_{ij} is expected to be high only if there is at least one path on which x_i has high signal strength on both the links. If P_0 is the transmission power level, then, we can estimate the RSS of x_i at AP_j as follows:

$$RSS_{ij} \approx P_0 \times \max_{k=N\ldots M}\left(\min\left(||h^{(1)}_{ik}||^2, ||h^{(2)}_{kj}||^2\right)\right) \qquad (7)$$

Consider client C_i that transmits packet x_i at data-rate R_i. Let AP_j be the receiving AP that decodes x_i. If τ_i is the minimum SNR required to decode x_i where τ_i depends on the physical layer data rate, then the residual noise that can be tolerated at AP_j during the decoding is given by [6]: $\frac{RSS_{ij}}{\tau_i}$. Using this, BBN computes the maximum residual noise that each packet can tolerate. Let us say AP_j decodes the i^{th} packet in the decoding sequence.

Observe that in BBN the packets are decoded sequentially. So, if a packet is not decoded correctly, then all other packets that depend on it can't be decoded either. So, in order to improve the decoding probability of all the packets, the decoding order is chosen by arranging the packets in non-increasing order of the maximum residual noise that they can tolerate.

5. MAC DESIGN

In this section, we first explain how BBN works in large scale networks. Next, we explain how BBN leverages the variation in the density of access points to improve the throughput of the uplink traffic. Finally, we explain how BBN coexists with ongoing downlink traffic in the network.

5.1 Multi-Collision Domain

The previous sections describe how BBN works in a single collision domain. To work in a practical multi-collision domain, BBN needs to solve multiple challenges:

1. In a multi-collision domain network, an AP may not be able to hear all other APs. This makes it difficult to synchronize them since one of the best performning algorithms with high synchronization accuracy [20] works only within a single collision domain.
2. The traffic distribution may be different across different parts of the network. For example, some parts of network may experience higher downlink traffic compared to others.
3. The MAC algorithm should ensure fairness across different clients.
4. Previous discussion of BBN requires that all cooperating APs and all clients are able to hear each other. Satisfying this requirement is challenging since frequent mobility of clients requires frequent re-computations.

BBN as described in Section 2 requires that: (i) **All cooperating APs should be able to hear each other**; and, (ii) **All APs should be able to hear all clients**. So, one naive way of extending BBN to multi-collision networks would be to arrange both the APs and clients in groups such that within each group all APs and all clients can hear each other. However, this naive approach would require frequent re-computation of groups due to client mobility.

To ensure that BBN works with mobile networks without requiring frequent re-computations, we divide the EWLAN into cliques of APs while only satisfying the first requirement. Satisfying that requirement implies decomposing the graph into as few cliques of APs as possible. Since, decomposing graphs into fewest cliques is an NP-Hard problem, BBN uses a greedy polynomial-time algorithm to compute such cliques. Our polynomial-time algorithm repeatedly finds a maximal clique among all APs. Then, it removes the vertices (and the edges incident on them) that are part of the maximal clique. The algorithm then runs on the remaining graph to find the maximal clique. This process is repeated until every AP is a part of some clique. All the APs that are in the same maximal clique, form a single *group*. This decomposition algorithm can be run by a central server similar to [24, 15]. *Ensuring that all APs in the same group can hear each other allows BBN to leverage the existing synchronization algorithms (such as SourceSync [20]) to synchronize all the APs that are part of the same group.*

Observe that since the APs are immobile, once the membership of different groups has been computed, it can be used for long periods of time. It is possible that an AP may not be able to hear a client that belongs to the same group. Thus, grouping based on APs only satisfies the first requirement specified above while the second requirement may be violated. We handle this in Subsection 5.2.1.

Computing neighbor relationship among groups: To prevent interference from neighboring groups and to keep groups independent, BBN ensures that at any time if the APs belonging to group G are communicating, then the APs belonging to neighboring groups should not communicate. Two groups (say G_i and G_j) are said to be neighbors of each other if (i) There exists a wireless device (an AP or a client) in G_i that is in the interference range of a wireless device in G_j; or, (ii) There exists a wireless device (an AP or a client) in G_j that is in the interference range of a wireless device in G_i. To decouple the dependence of neighbor-relation from the location of mobile clients, BBN takes a conservative approach such that G_i and G_j are called neighbors even if there could potentially exist a client that can be in the transmission range of some AP in G_i while being in the interference range of some AP in G_j. *By decoupling the neighbor relationship from the location of mobile clients, BBN significantly reduces the overhead that may otherwise arise due to frequent re-computations.*

Scheduling different groups: To ensure that two neighboring groups are not transmitting simultaneously, BBN uses a central server [21, 15, 24] that manages the interference among neighboring groups. Since the schedule length in BBN is always two slots across all the groups, it makes it convenient for the server to schedule the active groups. In BBN, at any time t, the server computes the set of groups that will communicate for the next two slots (t and $t+1$). This set is computed using maximum independent set techniques such that two groups are allowed to transmit/receive simultaneously only if they do not interfere with each other. However, due to unexpected delays on the wired backbone, the latency from the central server to the APs may result in APs unnecessarily waiting for the control messages from the server while the wireless channel is idle. To avoid this waiting, the server in BBN proactively computes the schedule and transmits it to the APs over the backbone.

Client-AP association: In BBN, clients do not permanently associate with any specific AP or a group. The clients simply wait for the *poll* packet from any neighboring AP and transmit uplink data as soon as they receive the corresponding *approve* packet as shown in Fig. 6. The *poll* packet is a special control packet that is broadcast by the APs, requesting the clients to reply if they have any uplink packet. *By keeping the clients stateless, BBN reduces the control messages exchanged between APs and clients.*

ACK transmission: In BBN, the APs decode the packets during Phase 3. After decoding, the APs send ACK over the wireless to the clients as shown in Fig. 6.

Downlink traffic: Uplink transmissions in BBN can coexist with downlink traffic. Each group in BBN can either perform downlink transmissions or uplink transmissions, independently of the other groups. For downlink communication, existing algorithms [21, 15, 24] can be used. The central server used in BBN can also be used for managing downlink interference as in the existing algorithms [15, 24].

5.2 Computing the set of transmitting clients

In a system with N clients and $\frac{N^2-N+2}{2}$ APs, BBN guarantees that each client can transmit 1 packet every two slots. Within a single group, it is possible that the number of APs may not be high enough to support all the clients. In that case, the group-head AP selects a subset of clients that would transmit in the first time slot. To ensure fairness among clients, BBN uses a weighted credit based system [15] such that the credit of a client is high if it has not been scheduled for a long period of time. Thus, the clients with the highest credit are given priority to transmit. This is further described in Section 5.2.1.

Fig. 6 explains the complete working of BBN. Initially, the APs in a group (if allowed by the central server) poll the network for uplink traffic. To make sure that all of the clients are able to receive the poll message, the APs broadcast the same poll message simultaneously. This is followed by a contention period in which different clients transmit short packets conveying their credit balance to contend for the uplink transmission. At the same time, all APs in the group hear for such packets and forward them to the "group-head AP". The group-head AP upon receiving the forwarded packets from all APs computes the set of clients that are allowed to transmit their data packets. This information is conveyed by the group-head AP back to the other APs. Next, all the APs simultaneously broadcast the *Approve* message that contains the list of clients that are allowed to transmit in this slot. Finally, the approved clients transmit their data packets which are decoded by the APs in three phases as described in Section 4. Broadcasting simultaneously requires the APs to be synchronized with each other. However, since all APs can hear each other, we achieve this synchronization using SourceSync as discussed before. This information exchange among APs may take some time due to non-zero latency over the wired backbone. During this time, the APs may utilize the wireless channel for other transmissions.

On the other hand, it is also possible that the number of clients are low while there are more APs available (*e.g.*, in highly dense networks such as in Fig. 3). In that case, BBN can leverage the extra APs to further improve the robustness of decoding as discussed in Section 5.3.

5.2.1 Approve *algorithm*

In each group, a single AP is elected as the *group-head AP* that executes the *Approve* algorithm to compute the set of clients that are allowed to transmit. *Approve* (Algorithm 1) greedily computes the schedule. In each iteration, it adds the client with the highest credit value to the schedule (Line 8), thereby improving fairness. For such a client, it picks the best AP (say AP_j) that has not yet been paired with some other client (Lines 11-15). Next, Approve tries to add this client-AP pair to the schedule S and checks if S is still satisfiable (Lines 16-18). This check is done by Algorithm *Satisfiable*. If this pair makes S unsatisfiable (Lines 19-21), then the pair is removed from S. Also, C_i is marked as ineligible since it cannot be paired with any AP. This process is repeated until no more client-AP pairs can be added to S (Lines 9-10).

Algorithm *Satisfiable* determines if a given schedule is satisfiable or not. When doing this computation, *Satisfiable* takes into account the set of clients that each AP can hear. Without loss of generality, let S be the schedule such that $S = \{(C_i, AP_i) : AP_i$ is the receiving AP for packet x_i and x_i is the i^{th} packet to be decoded$\}$. *Satisfiable* should return true if for every client-AP pair, say (C_i, AP_i), it can find a subset of $i-1$ unique APs in the same group that can align x_i at the receiving APs (AP_1 to AP_{i-1}). In other words, for every client-AP pair, say (C_i, AP_i), *Satisfiable* needs to find $i-1$ other APs that are in the transmission range of C_i. This computation can be done by reducing this problem to a Max Flow problem. We refer the reader to the technical report [1] for detailed discussion.

Algorithm 1: *Approve*: Computes the set of clients that will be approved in this slot

```
1  Input: For every eligible packet $P_i$, its transmitter $C_i$.
   Also, information on which AP can hear which client.
2  Output: (i) Set of clients that will be approved in this slot.
   (ii) The matching from the approved clients to the APs
   indicating which AP decodes which packet. (iii) The
   decoding order.

   // Set eligibility of all clients to true
3  $E_i \leftarrow true\ \forall i : 1 \le i \le N$
4  $\mathcal{A} \leftarrow$ All APs in the current group
   // $S$ is an ordered schedule that tells us which AP
      decodes which packet.  The client-AP pairs are
      arranged in the order in which they are decoded.
5  $S \leftarrow \{\}$
6  while $true$ do
7      $CSet \leftarrow \{C_x : E_x = true$ and $C_x \notin S\}$
8      $C_i \leftarrow C_i \in CSet$ and $C_i$ has the highest credit balance
9      if $C_i = null$ then
10         return $S$
11     $Set \leftarrow \{(C_i, AP_j) : AP_j \notin S\}$
12     $(C_i, AP_j) \leftarrow (C_i, AP_j) \in Set$ and $RSS_{ij}$ is maximum
13     if $AP_j = null$ then
14         $E_i \leftarrow false$
15         continue
16     $S \leftarrow S \cup \{(C_i, AP_j)\}$
17     Compute the decoding order in $S$ based on the residual
       noise tolerance.
18     $isSatisfiable \leftarrow Satisfiable(S, \mathcal{A})$
19     if $isSatisfiable = false$ then
20         $E_i \leftarrow false$
21         $S \leftarrow S \backslash \{(C_i, AP_j)\}$
22 return S
```

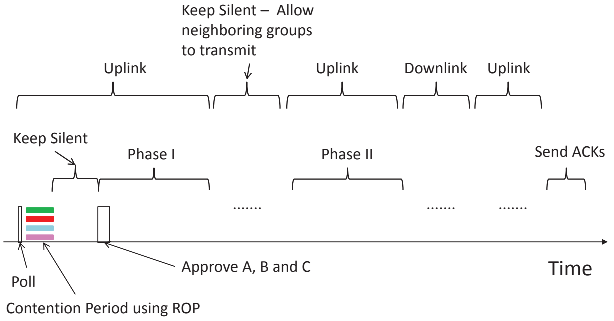


Figure 6: Timeline of data transmission in a large network. The data sent by clients during contention phase are transmitted using the Rapid OFDM Polling (ROP) [30] scheme to decrease overhead. All nodes keep silent after ROP and wait for the group-head AP to calculate and distribute the *Approve* message through the backbone network. Phase III is executed in the background over the wired backbone allowing wireless channel to be used for other purposes.

5.3 Robustness

In BBN, the first AP decodes one packet while $N-1$ packets are nulled using blind beamforming. The second AP decodes the second packet (while the other $N-2$ packets are nulled using blind beamforming) and so on. Thus, if an AP cannot decode a packet due to inaccuracies in blind beamforming or packet subtraction, then all the following packets that depend on it can also not be decoded. Therefore, to ensure that the first few packets in the decoding order can be decoded with high probability, BBN leverages the high density of APs. Specifically, BBN increases the decoding robustness of the packets if the number of APs present in a group are more than the minimum required (See Sec. 2).

Let $C_1, C_2, C_i, \cdots, C_N$ be the order in which the clients are decoded (See Section 4.4). Let the number of APs in the group be M and E be the number of extra APs that are present such that $E = M - \frac{N^2-N+2}{2}$. Recall that exactly $N-i$ packets are nulled (or aligned) at the AP that decodes the packet from client C_i. If we require one of the extra APs to independently decode the packet from C_i, then we will need another $N-i$ extra transmitting APs to ensure that packets from $C_{i+1}, C_{i+2}, \cdots, C_N$ are nulled at this extra AP. Thus, to decode the packet from C_i at two different APs, we need an additional $N-i+1$ APs (including one extra AP for receiving).

BBN increases the decoding robustness as follows: The APs in BBN find the *first* client C_i in the decoding sequence that satisfies the two requirements: (i) The packet from C_i is decoded at only one AP; and, (ii) $E \geq N-i+1$. Let C_i be the first client in the decoding sequence that satisfies the two constraints. Then, BBN ensures that the packet transmitted by C_i can be independently decoded by two different APs. BBN decreases E by $N-i+1$ since this is the number of APs required to achieve independent decoding of C_i. Finally, this process is repeated as long as possible to achieve independent decodings of some packets. Thus, in highly dense networks, BBN leverages the extra APs present in the network to further increase the decoding probability of each packet. Even if extra APs are not available, BBN can restrict the number of clients that transmit simultaneously. This frees up some APs that can be used for increasing the robustness of decoding. Currently, we leave the problem of proactively reducing the number of transmitters to increase the decoding robustness as our future work.

6. EXPERIMENTS

6.1 Setup

We evaluate BBN in a testbed with 7 USRP N210 nodes. The setup is as follows:

1. **Hardware and software setup:** Each USRP is equipped with a WBX daughterboard and operates in the 400 MHz band. All nodes are within single collision domain. At the receiver side, we use the GNURadio for signal processing. The decoding is done offline in Matlab. All of the AP nodes are synchronized with an external clock source generated by OctoClock-G [2]. In practice, SourceSync [20] can be used to synchronize the transmitting APs to a nanosecond level accuracy.

2. **OFDM and modulation setup:** We use a 512 FFT system, with 200 subcarriers used for data transmitting. The cyclic prefix length is set to 128. Unless otherwise mentioned, Binary Phase Shift Keying (BPSK) is used as the modulation scheme. The sampling rate is set to 1MHz.

Apart from implementing BBN, we also implemented **Omniscient TDMA** that utilizes a central server. This server is aware of (i) packet queue at different clients; and, (ii) the channel between all clients and all APs. Omni-TDMA schedules the three different clients in a round-robin fashion with each client transmitting to the AP to which it has the best channel.

6.2 Micro-Benchmarks

Many works have shown the effectiveness of beamforming [21, 11]. Since our blind-beamforming and nulling involves transmitting unknown samples, its effectiveness and accuracy is unclear. In this section, we evaluate the performance of blind-beamforming and nulling using the signal to interference and noise ratio (SINR). In the following experiments, 3 clients and 4 APs were deployed in our testbed as shown in Fig. 1.

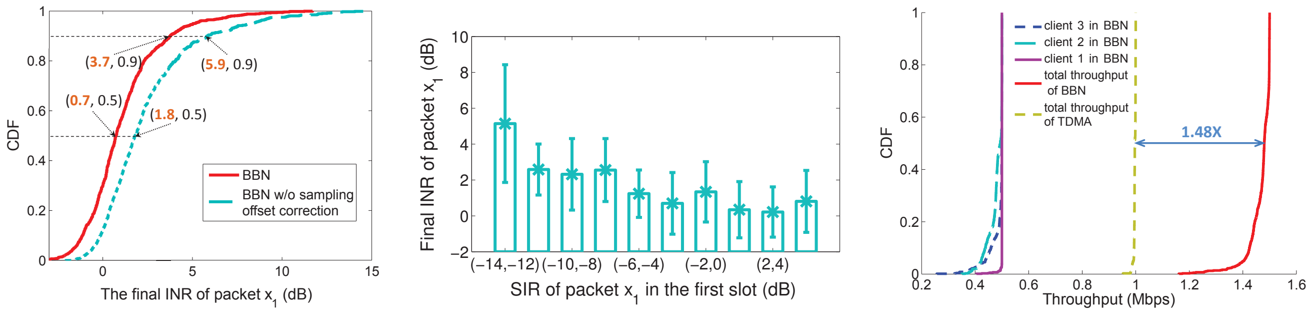
6.2.1 Blind-beamforming and Nulling Effects

First of all, we study the blind-beamforming and nulling effect as described in Section 4. Since there are a total of 12 links between all APs and clients, it is difficult to control the SNR of every link. Instead, we place the clients and APs randomly in our testbed and record the actual SNRs. We repeat the experiment 20 times for each of the 50 randomly chosen topologies. Over various topologies, the SNR between clients and APs varied from 6 dB to 35 dB. We compute the final interference to noise ratio (INR) of packet x_1 when it is decoded by AP_1. The INR distribution is shown in Fig. 7(a). The median of the INR is 0.7 dB, which is just slightly above the noise floor, and the 90^{th} percentile INR is 3.7 dB. This indicates that residual interference from blind-beamforming and nulling is relatively small and demonstrates the practicality of BBN.

The INR distribution in Fig. 7(a) shows that it could be as high as 10 dB, which is a large value compared with typical SNR values, *e.g.*, 20 dB. We look deeper into the INR results and present it in an another way in Fig. 7(b). The y-axis is the final INR of packet x_1 at AP_1. The x-axis is the range of signal to interference ratio (SIR) in dB that x_1 experiences in the first slot across all of the APs. The smaller the value on the x-axis, the higher the amount of interference to be cancelled in the second slot. This figure shows that as the SIR increases, the final INR decreases. When the first slot SIR is larger than -12 dB, the median of the that is 0.6 dB and the 90^{th} percentile is 2.7 dB (Fig. 7(b)). Based on this result, we can enable BBN when the SIR value is larger than a threshold and fall back to the default IEEE 802.11 scheme when the SIR value is small. We leave the study of computing the exact threshold value as future work.

6.2.2 Sampling Offset

As discussed in Section 4.3, there is sampling offset between the samples received by AP_1 from phase I and phase II. To study the effect of the sampling offset, we turn off the sampling offset correction in BBN and compute the residual interference to noise ratio for x_1. The result shown in Fig. 7(a) shows that without sampling offset correction, the median INR increases by 1.1 dB and the 90^{th} percentile increases by 2.2 dB. This demonstrates that the sampling offset correction done in BBN reduces the residual interference.



(a) Final INR distribution of x_1 at the first decoding AP. The median of the INR is 0.7 dB and the 90^{th} percentile INR is 3.7 dB.

(b) Final INR of the packet x_1 at the first decoding AP.

(c) Throughput of BBN compared to Omniscient TDMA.

Figure 7: Experiment results collected over USRP testbed.

6.3 Throughput

In this section, we study the throughput performance of BBN. The throughput of each client in BBN is recorded and compared with that of omniscient TDMA. A total of 20000 packets are transmitted by each client across different topologies. Fig. 7(c) shows that on an average, BBN provides a throughput gain of 1.48× compared with omniscient TDMA. The figure also shows the throughput of client 1 is higher than that of clients 2 and 3. This is because x_2 and x_3 are decoded only if x_1 is decoded. Further, even if x_1 is decoded, x_2 and x_3 may not be decoded due to the residual interference from subtraction.

7. TRACE-DRIVEN SIMULATION

This section explains the setup and the results from the trace-driven simulations.

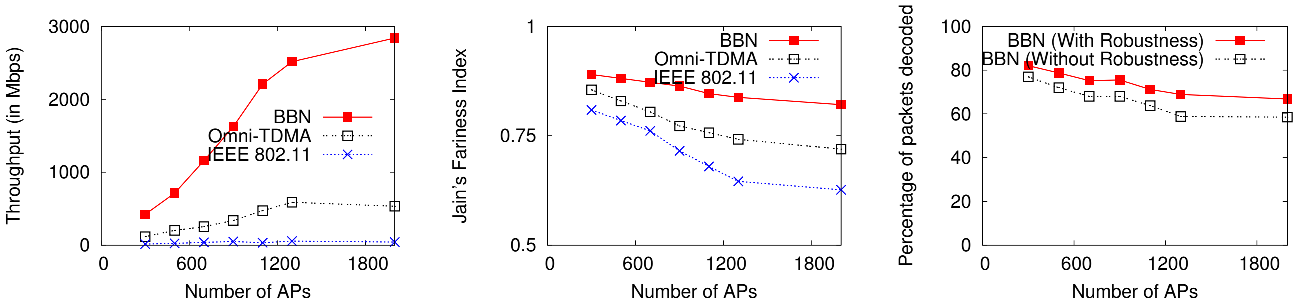
7.1 Simulation Setup

Apart from implementing BBN, we also implemented two other algorithms: (i) **Omniscient TDMA algorithm:** Described before in Section 6. However, this time similar to BBN, Omniscient TDMA also uses a credit-based system where a client has high credit value if it has not transmitted for a long time. In each slot, it schedules a maximum independent set of "client to AP" links (where weight of link = credit of the client × throughput of the client when using that link). The physical layer rate of a link is chosen by picking the highest data rate that can be decoded by the AP; and, (ii) **IEEE 802.11g (without RTS/CTS).** To evaluate the gain provided by BBN irrespective of the downlink algorithm used, only the uplink traffic from clients to APs was generated. Various traces were incorporated into the simulation: (i) **Noise due to Blind Beamforming and Nulling:** The simulator incorporated noise arising due to imperfect nulling. For this, we used the traces collected from our experiments (See Fig. 7(a)). (ii) **Noise due to subtraction:** When an AP subtracts a packet, it has to recreate its samples and correct for various offsets such as sampling offset and frequency offset. An imperfect correction leads to imperfect subtraction resulting in residual noise. The simulator incorporated this residual noise using the traces collected by us in experiments. (iii) **Path Loss between clients and APs:** Incorporated from the traces [25]. (iv) **Path Loss between APs:** Incorporated from the traces [25]. For simplicity, the packets were assumed to be of constant length (1500 bytes). In practice, one may use packet aggregation to avoid the overhead induced by small sized packets. In this section, we study the behavior of BBN in a large EWLAN that spans over multiple collision domains (*e.g.*, the campus of a university). Our simulator first randomly deploys 1000 clients in a field of size 500m × 500m. APs are also deployed randomly and the number of APs is varied. In this setup, different devices may belong to different groups as described in Section 5. The overhead of different protocols was taken into account during the simulation. For BBN, this includes the overhead of all the control packets in both wired and wireless domains. As discussed in previous sections, when the APs are exchanging control packets over the backbone for the purpose of computing the set of approved clients or for computing the precoding vectors, at that time the wireless channel can be utilized for other transmissions. In the simulation also, during these periods, the central server allows the neighboring groups to use the wireless channel for their transmissions. Also, APs in BBN used extra APs to further increase the decoding robustness as described in Section 5.3. Finally, clients in BBN and IEEE 802.11 used the Auto Rate Fallback (ARF) algorithm to determine the physical layer data rate.

7.2 Results

Next, we describe the results from our trace-driven simulations.

1. **Total Throughput across all clients:** Throughput increases for all algorithms as they leverage the increase in the physical layer data rate (See Fig. 8(a)). For 802.11, the increase is not substantial since a large number of collisions (due to hidden terminals) reduces the number of successful transmissions. With increase in number of APs, the throughput in BBN increases because of two reasons: (i) Higher AP density implies more APs are present in each group, resulting in higher throughput since more clients can be supported at the same time; and, (ii) Higher data rate at clients due to



(a) Total Throughput (b) Jain's Fairness Index (c) Percentage of packets decoded

Figure 8: Trace-Driven Simulation Results for Multi-Collision Domain

higher AP density. As the density of the APs increase, throughput in BBN increases substantially compared to TDMA. When the number of APs is 2000, each client is in the range of an average of 76 APs. At that density, BBN throughput is 5.6× compared to TDMA, and 52.4× compared to IEEE 802.11. This is lower than the expected gain since in BBN, clients use ARF to adjust their physical layer data rate while clients in Omniscient TDMA transmit at the best possible data rate.

2. **Fairness:** Fig. 8(b) shows the variation in Jain's fairness index with variation in number of APs. IEEE 802.11 has very low fairness since a client may get starved if it is in the range of multiple APs. The fairness index of BBN is higher than other algorithms since BBN allows all clients to transmit. BBN has higher fairness than TDMA since BBN performs precoding over transmissions from all clients. Thus, even the clients that are far away from all APs may experience high throughput due to beamforming from multiple helper APs.

3. **Decoding probability:** As the density of the network becomes higher, the length of the decoding chain increases. Thus, a decoding failure on one packet implies a decoding failure on all the other packets that depend on it. Fig. 8(c) shows the percentage of packets decoded successfully decreases with increase in density. However, still the throughput in BBN increases (See Fig. 8(a)) since higher density enables multiple clients in APs to transmit successfully. Further, with high density of APs, BBN can use robustness techniques discussed in Section 5.3 to increase the decoding probability. Fig. 8(c) also shows the decoding probability when BBN does not use robustness techniques described before. With the increase in number of APs, there is a higher chance that BBN can leverage those APs to improve robustness. Thus, with increasing density, the improvement provided by robustness further increase.

8. DISCUSSION

In this section, we discuss some further modifications that make BBN more practical.

Reducing overhead of channel estimation: To compute the precoding vectors, the APs in BBN require the knowledge of channel between all clients and APs as well as the channel between all APs. The problem of computing the channel from clients to APs has been well studied in the context of MIMO networks [11, 29]. To compute the channel values, we plan to use PN sequences to estimate the channel from multiple transmitters simultaneously [16].

Overhead on the backbone: In BBN when decoding N uplink packets, the APs need to exchange $\frac{(N-1)\times(N-2)}{2}$ data packets. This is in contrast with [21] that requires exchange of $(N-1)\times(N)$ data packets when performing joint beamforming for the downlink traffic. In a MU-MIMO system where APs exchange sample information through the backbone, the overhead on the backbone is almost $23N$ packets if the data rate is 24 Mbps (16 QAM, 1/2 coding rate) and the 802.11a standard is used. BBN introduces less overhead on the backbone compared to MU-MIMO if the number of clients is no more than **24**, while achieving the same throughput gain of $N/2$. In addition, APs in BBN also need to exchange relatively smaller control packets related to channel state information, scheduling etc. The number of participating clients can be adjusted based on how much overhead can be tolerated on the wired backbone.

APs with multiple antennas: If the APs are equipped with multiple antennas, BBN can leverage them to reduce the number of required APs. Specifically, if each AP is equipped with K antennas, then to receive N uplink packets simultaneously, BBN would require only $\frac{N'^2-N'+2}{2K}$ APs where $N'=N-(K-1)$, a reduction by a factor of more than K. On the other hand, with M APs each with K antennas, BBN can support around $\sqrt{2MK}+K$ clients instead of $\sqrt{2M}$ clients with one antenna per AP.

Packet Aggregation: In practice, packet sizes can be different and clients may pick different data rates, resulting in different uplink transmission durations. However, the transmissions in BBN are scheduled slot by slot. This slot based scheduling may result in a waste of channel resources since the transmission of some clients may not take the whole slot duration. To solve this problem, we use the same technique, packet aggregation, as in 802.11n standard. More specifically, we aggregate and split packets based on the data rates to create virtual packets that take exactly one slot to transmit.

9. RELATED WORK

Although BBN builds on several prior work, it differs from them in various ways.

Backbone usage: The idea of using the wired backbone to increase wireless throughput is not new. In MegaMIMO[21], multiple APs cooperatively precode the transmissions such that each client receives only the packets intended for it while the other transmissions are canceled out. However, MegaMIMO requires that transmitters exchange packets among themselves and thus, it works only for the downlink transmissions. On the other hand, BBN improves the throughput for the uplink traffic. Also, in contrast to MegaMIMO and OpenRF [15], transmitters in BBN jointly perform nulling without knowing the actual contents of the packets.

A recently proposed protocol Symphony [6] also focuses on uplink traffic. However, in contrast to BBN, Symphony improves the network throughput only when the APs are in different collision domains. In Epicenter [12], authors propose that APs should exchange coarse representations of symbols to decode corrupted bits. Similarly, authors in [28] also propose that APs exchange bits or raw samples on the backbone to facilitate packet decoding. In all these algorithms, the APs cooperate to decode the same packet whereas in BBN, APs encourage transmitters to collide and then cooperate to decode multiple packets simultaneously without exchanging the raw samples.

The idea of using the backbone for improving the uplink throughput was also proposed in our earlier work called RobinHood [5]. There are several differences between RobinHood and BBN. First, RobinHood was designed for a single collision domain, whereas BBN works for multiple collision domains. Second, BBN includes several techniques to increase the decoding probability. And third, it requires N fewer APs as compared to RobinHood since BBN utilizes the samples from the first slot at the receiving APs.

Interference Alignment: Previously, researchers (see [13] and references therein) have used interference alignment to improve the capacity of wireless networks. However, unlike BBN, they either require APs to exchange samples over the backbone [4], work only for the downlink traffic [26], assume presence of significant number of clients [19], require multiple antennas at transmitters or receivers [11], require the antennas to be physically moved [3] to a certain point, require the channel to change from one slot to another [7], precode over exponential number of time slots [7], or provide limited throughput gain [3], or do not scale with number of APs [11]. These assumptions are not practical in mobile networks since if the client is stationary, the channel may not change [27] from one packet to another. In contrast to the previous works, BBN works even if the channel stays stationary. The concept of interactive interference alignment [8] is similar to our work. However, it requires three slots to finish the transmissions. Also it requires the clients to be synchronized and transmit in two of the slots. TurboRate [23] introduces a client rate adaptation scheme for multiuser MIMO networks. Each client learns the direction of its signal received at the AP and uses SNR after projection on the direction of other clients to pick the best uplink data rate. The idea of SNR projection could help us design a better data rate selection algorithm for BBN.

Wireless Relays: Researchers [22, 14] have also looked at the problem of using special relay nodes to assist in high speed communication between specific pairs of source and destination nodes. In contrast, the focus of BBN is to leverage the high density of APs and the wired backbone to carefully select the set of destination APs, determine which AP decodes which packet, and to use the wired backbone to migrate all the complexity away from the clients. Further, with previous works, it is possible that the destination AP is unable to decode a packet due to low SNR. However, in BBN, APs leverage the high density of APs to increase robustness (See Sec. 5.3).

10. CONCLUSIONS AND FUTURE WORK

In the previous sections, we discussed BBN, a blind beamforming and nulling scheme that leverages the high density of access points to enable multiple mobile devices to transmit simultaneously. Feasibility of BBN was verified on a USRP testbed. Measurements show that BBN achieves a throughput gain of 1.48× over omniscient TDMA. Using trace-driven simulations, we showed that in dense wireless LANs, BBN provides a throughput of up to 5.6× compared to omniscient TDMA. Currently, traffic to BBN may interfere significantly with the packet exchange in BBN. One way to avoid this is for wireless devices in BBN to send RTS/CTS at the beginning so as to block the wireless channel for a certain duration.

Acknowledgements

This material is based upon work partially supported by the National Science Foundation under Grants CNS-1161490 and CNS-1302620.

11. REFERENCES

[1] BBN. Tech. rep. `http://sites.google.com/site/anontechrep765/`.

[2] OctoClock-G, accessed Jan. 2014. `https://www.ettus.com/product/details/OctoClock-G`.

[3] Adib, F., Kumar, S., Aryan, O., Gollakota, S., and Katabi, D. Interference Alignment by Motion. In *Proc. of ACM MobiCom 2013*.

[4] Annapureddy, V. S., El Gamal, A., and Veeravalli, V. V. Degrees of Freedom of Interference Channels with CoMP Transmission and Reception. *IEEE Transactions on Information Theory 58*, 9 (2012), 5740–5760.

[5] Bansal, T., and *et al*. RobinHood: Sharing the Happiness in a Wireless Jungle. In *Proc. of ACM HotMobile 2014*.

[6] Bansal, T., and *et al*. Symphony: Cooperative Packet Recovery over the Wired Backbone in Enterprise WLANs. In *ACM MobiCom 2013*.

[7] Cadambe, V. R., and Jafar, S. A. Interference Alignment and the Degrees of Freedom for the K User Interference Channel. *IEEE Transactions on Information Theory* (2007).

[8] Geng, Q., Kannan, S., and Viswanath, P. Interactive Interference Alignment. *CoRR* (2013).

[9] Gesbert, D., Kountouris, M., Heath, R., Chae, C.-B., and Salzer, T. Shifting the MIMO Paradigm. *Signal Processing Magazine, IEEE 24*, 5 (Sept 2007), 36–46.

[10] GOLLAKOTA, S., AND KATABI, D. Zigzag Decoding: Combating Hidden Terminals in Wireless Networks. In *ACM SIGCOMM 2008*.

[11] GOLLAKOTA, S., PERLI, S. D., AND KATABI, D. Interference Alignment and Cancellation. In *Proc. of ACM SIGCOMM 2009*.

[12] GOWDA, M., SEN, S., ROY CHOUDHURY, R., AND S., L. Cooperative Packet Recovery in Enterprise WLANs. In *IEEE INFOCOM 2013*.

[13] JAFAR, S. A. *Interference Alignment: A New Look at Signal Dimensions in a Communication Network*. Now Publishers, 2011.

[14] KUHN, M., BERGER, S., HAMMERSTROM, I., AND WITTNEBEN, A. Power Line Enhanced Cooperative Wireless Communications. *IEEE JSAC 24*, 7 (2006), 1401–1410.

[15] KUMAR, S., CIFUENTES, D., GOLLAKOTA, S., AND KATABI, D. Bringing Cross-Layer MIMO to Today's Wireless LANs. In *Proc. of ACM SIGCOMM 2013*.

[16] LI, T., AND *et al*. CRMA: Collision-Resistant Multiple Access. In *Proc. of ACM MobiCom 2011*.

[17] MAGISTRETTI, E., GUREWITZ, O., AND KNIGHTLY, E. 802.11 ec: Collision Avoidance Without Control Messages. In *Proc. of ACM MobiCom* (2012).

[18] MURTY, R., PADHYE, J., CHANDRA, R., WOLMAN, A., AND ZILL, B. Designing High Performance Enterprise Wi-Fi Networks. In *Proc. of USENIX NSDI 2008*.

[19] NAZER, B., AND *et al*. Ergodic Interference Alignment. In *Proc. of IEEE ISIT 2009*.

[20] RAHUL, H., HASSANIEH, H., AND KATABI, D. SourceSync: A Distributed Wireless Architecture for Exploiting Sender Diversity. In *Proc. of ACM SIGCOMM 2010*.

[21] RAHUL, H., KUMAR, S., AND KATABI, D. MegaMIMO: Scaling Wireless Capacity with User Demand. In *Proc. of ACM SIGCOMM 2012*.

[22] RANKOV, B., AND WITTNEBEN, A. Spectral Efficient Protocols for Half-Duplex Fading Relay Channels. *IEEE Journal on Selected Areas in Communications 25*, 2 (2007), 379–389.

[23] SHEN, W.-L., TUNG, Y.-C., LEE, K.-C., LIN, K. C.-J., GOLLAKOTA, S., KATABI, D., AND CHEN, M.-S. Rate Adaptation for 802.11 Multiuser Mimo Networks. In *Proc. of ACM Mobicom 2012*, pp. 29–40.

[24] SHRIVASTAVA, V., AND *et al*. CENTAUR: Realizing the Full Potential of Centralized WLANs Through a Hybrid Data Path. In *Proc. of ACM MobiCom 2009*.

[25] STANFORD INFORMATION NETWORKING GROUP (SING). SING Datasets. `http://sing.stanford.edu/srikank/datasets.html`.

[26] SUH, C., HO, M., AND TSE, D. N. Downlink Interference Alignment. *IEEE Transactions on Communications 59*, 9 (2011), 2616–2626.

[27] VUTUKURU, M., BALAKRISHNAN, H., AND JAMIESON, K. Cross-Layer Wireless Bit Rate Adaptation. In *Proc. of ACM SIGCOMM* (2009).

[28] WOO, G. R., KHERADPOUR, P., SHEN, D., AND KATABI, D. Beyond the Bits: Cooperative Packet Recovery Using Physical Layer Information. In *Proc. of ACM MobiCom 2007*.

[29] XIE, X., ZHANG, X., AND SUNDARESAN, K. Adaptive Feedback Compression for MIMO Networks. In *Proc. of ACM MobiCom 2013*.

[30] ZHOU, W., LI, D., SRINIVASAN, K., AND SINHA, P. DOMINO: Relative Scheduling in Enterprise Wireless LANs. In *Proc. of ACM CoNEXT 2013*.