# BiPass: Enabling End-to-End Full Duplex

Lu Chen[⋆†], Fei Wu[⋆†], Jiaqi Xu[⋆†], Kannan Srinivasan[⋆], Ness Shroff[⋆⋄]
[⋆]Department of Computer Science and Engineering, [⋄] Electrical and Computer Engineering
The Ohio State University, Columbus, OH, 43210
{chen.4728, wu.1973, xu.1629, srinivasan.115, shroff.11}@osu.edu
[†] Co-primary authors

## ABSTRACT

Full duplex techniques can potentially double the channel capacity and achieve lower delays by empowering two radios to simultaneously transmit in the same frequency band. However, full duplex is only available between two adjacent nodes within the communication range. In this paper, we present BiPass to break this limitation. With the help of full duplex capable relays, we enable simultaneous bidirectional in-band cut-through transmissions between two far apart nodes, so they can do full duplex communications as if they were within each other's transmission range. To design such a system, we analyze interference patterns and propose a loop-back interference cancellation strategy. We identify the power amplification problem at relay nodes and develop an algorithm to solve it. We also develop a routing algorithm, an opportunistic forwarding scheme, and a real-time feedback strategy to leverage this system in ad-hoc networks. To evaluate the real world performance of BiPass, we build a prototype and conduct experiments using software defined radios. We show that BiPass can achieve 1.6x median throughput gain over state-of-the-art one-way cut-through systems, and 4.09x gain over the decode-and-forward scheme. Our simulations further reveal that even when the data traffic is not bidirectional, BiPass has 1.36x throughput gain and 47% delay reduction over one-way cut-through systems in large networks.

## CCS CONCEPTS

• **Networks** → *Wireless access networks*;

## KEYWORDS

Full duplex; interference cancellation; multi-hop; relay

## 1 INTRODUCTION

Full duplex is a promising technique for enhancing future communication systems. Full duplex techniques allow radios to transmit and receive packets simultaneously in the same frequency band, which usually means higher spectrum efficiency and lower packet delays [5, 6, 8, 9, 13]. However, full duplex can only be activated between two adjacent nodes due to the fact that any radio has a limited communication range.

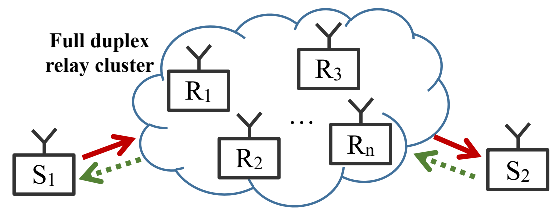



**Figure 1: Source nodes $S_1$ and $S_2$ want to exchange information. Is it possible to enable bidirectional cut-through transmissions via relays between them?**

Traditionally, relays have been employed to extend the communication range. Correspondingly, full duplex capable relays have been designed, which can forward the received signal while receiving [4, 7, 12]. Recent works [7, 17] have enabled one-way cut-through transmissions by leveraging the power of full duplex relays. Specifically, a source node can transmit packets to a far away destination node via full duplex relays almost instantly. Cut-through originates from network switch, with which a switch directly forwards a packet without decoding. However, full duplex, in its nature, is about in-band simultaneous bidirectional transmissions. This leads us to ask: ***Is it even possible to enable end-to-end full duplex via multiple full duplex relays***?

Consider the setup shown in Fig. 1. Two full duplex source nodes[1] $S_1$ and $S_2$ want to exchange data but are out of direct communication range. We want to enable simultaneous bidirectional in-band cut-through transmissions with full duplex relays between them. In other words, design a system that can enable $S_1$ and $S_2$ to do *end-to-end transmissions as if they were in direct communication range.*

This *end-to-end full duplex transmission* problem is an open problem. One could ask why state-of-the-art end-to-end single direction cut-through technique [7] ($S_1 \rightarrow S_2$) cannot naturally support bidirectional transmissions ($S_1 \leftrightarrow S_2$). The reason is that such a system is designed to cancel interference from one direction. Specifically, [7] measures the channels starting from $S_2$ backwards towards $S_1$ to measure all the causal channels and later use them to cancel such interference to enable $S_1 \rightarrow S_2$. Thus, to enable the reverse directional flow ($S_2 \rightarrow S_1$) simultaneously, [7] would need to measure the channels starting from $S_1$. These channels are different from the channels measured from $S_2$. Thus, the state-of-the-art

[1]Note that source nodes are also destination nodes in this setting. We also refer to them as end nodes in this paper.


*MobiCom '17, October 16–20, 2017, Snowbird, UT, USA*

ACM ISBN 978-1-4503-4916-1/17/10...$15.00
https://doi.org/10.1145/3117811.3117826

single direction cut-through system will not work for bi-directional end-to-end cut-through transmissions or end-to-end full duplex transmissions. More details on this aspect and power control follow in Sec. 3. This is a fundamental issue in enabling end-to-end full duplex traffic.

In this paper, we solve this open problem and present BiPass to realize the end-to-end full duplex system. There are many benefits of such a system. **Lower delay:** Compared with decode-and-forward (D&F) based solutions, our system does not need to forward packets hop by hop. Instead, packets can be exchanged as if they were close by. Compared with the state-of-the-art one-way cut-through systems [7, 17], the destination node in our system can transmit immediately without waiting for the end of the incoming stream, so it is likely to experience a much lower delay. **Throughput gain:** It has already been shown that one-way cut-through systems have throughput gain over traditional systems [7]. Similar to full duplex systems compared with half duplex systems, the throughput can potentially be further doubled in BiPass.

Despite the benefits it can potentially bring, there are multiple challenges in implementing BiPass to realize the above benefits.

**Interference management.** Due to the shared nature of the wireless medium, a transmission may be overheard by many other nodes. Since full duplex relays will forward signals that have been received, signals may keep looping among these relays. It has been shown that the interference in one-way cut-through systems is already difficult to analyze [7], and the problem is further exacerbated by the more complex interference pattern in the bidirectional setup.

**Power amplification.** Even if we can solve the interference problem, without a proper power amplification setting for the relays, the signal may gradually fade before reaching the target node, or suffer from nonlinear distortions due to over-amplification. Properly assigning power amplification coefficients is a problem that needs to be addressed.

**Utilize BiPass in networks.** In a wireless network, many relays are available to be employed to form a route. Selecting proper participating relay nodes is not an easy problem. Further, a network may have limited or even no bidirectional traffic flows. How to utilize BiPass efficiently in a network with mismatched traffic load is a question that entails further investigations.

This paper makes the following contributions:

- We propose a loop-back interference cancellation strategy and a power amplification algorithm as key solutions to address interference management and power amplification challenges above and realize the system in practice.
- We propose a routing selection algorithm, an opportunistic forwarding scheme, and a real-time feedback strategy to take full advantage of BiPass in large networks. Specifically, we relax the need for the entire path between $S_1$ and $S_2$ to be available and allow transmissions to still make progress towards their destinations. Thus, $S_1$ and $S_2$ in our framework need not be final destinations. Furthermore, the reverse flow could be end-to-end acknowledgements instead of data. This is vital for end-to-end recovery and data rate management.
- To the best of our knowledge, we are the first to propose and realize a system that enables simultaneous bidirectional in-band wireless cut-through transmissions via multiple full duplex relays. We implement our system with software defined radios, demonstrating its practicality and gains over the traditional decode-and-forward scheme, as well as the state-of-the-art one-way cut-through system.

## 2 RELATED WORK

**Full duplex and full duplex relay:** By cancelling self interference, a full duplex capable node can transmit and receive simultaneously in the same frequency band. Point-to-point full duplex systems have been well researched and successfully implemented. Specifically, [9, 13] propose antenna cancellation and analog cancellation technique respectively, [6] enables full duplex with single antenna, [3, 5, 8] show that MIMO is also possible for full duplex transmissions. Full duplex capability also holds for a relay node: it can amplify and forward the received samples almost immediately after receiving them. Full duplex relays have been successfully used in many recent works [4, 7, 12, 16].

**One-way cut-through transmissions:** Empowered by full duplex capable relays, one-way cut-through transmission has been shown to be possible. FastForward [4] enables constructive addition of signals from direct and relayed paths at the destination node, so that the received signal power is maximized. The authors in [12] propose a delay and forward system where relays also decode packets to avoid noise propagation problem. [1] uses full duplex relay at 60 GHz and applies it to virtual reality applications. However, these systems can only work for two-hop networks and are generally hard to directly extend to multi-hop cases. AirExpress [7] enables one-way cut-through transmissions via multiple relays by doing interference cancellation. The authors in [17] propose a scheduling scheme for one-way cut-through in networks. However, these systems can only support single direction transmissions, while BiPass can support bidirectional communications.

**Bidirectional transmissions:** Network coding is a strategy to improve the performance in bidirectional transmissions. There are several variations of network coding schemes including MAC-layer [2], physical-layer [18] and analog network coding [14]. The basic idea of network coding is to leverage the fact that each source node knows its own packets already, hence, a combined version of the information is enough for them to decode each other's information. In this sense, BiPass is also inspired by this idea. In fact, we think BiPass can be considered as *a generalization of the analog network coding* from one-relay networks to general multi-relay networks. In the design of [14], multi-hop is achieved by time division fashion scheduling. In contrast, BiPass is a coherent design for multi-hop networks, and has no requirement on interference range. Generally, current design of network coding techniques usually have stringent requirements such as negligible cross-hop interference, sophisticated power control, strict time/frequency synchronization, etc. In contrast, BiPass can be easily implemented in the real world without these requirements even for multi-hop communications. Also, compared with network coding, BiPass could potentially achieve larger throughput and lower delay since it has no concept of time slot, while network coding requires time division medium access. For example, 3 time slots are needed in a one-relay network with MAC-layer network coding.

## 3 THE DESIGN OF BIPASS

In this section, we discuss the design details of BiPass. All relay nodes in our system have two modes: *training mode* and *full duplex relay mode*. In the training mode, a node will transmit and receive simultaneously but without digital cancellation. It sends out a signal and estimates the channel with the received signal. In full duplex relay mode, a node will amplify and forward the signals that are being received, while performing interference cancellation using the channel estimation results from the training mode. Source nodes have the same *training mode* and *source mode*. In source mode, a source node will decode the signal after interference cancellation while transmitting its own packet.

### 3.1 Interference cancellation

To design a system that supports multi-hop bidirectional cut-through transmissions, the first question one may ask is: can we simply put several full duplex relays (doing only self interference cancellation) between source and destination? Unfortunately, this will lead to an unstable system due to an infinite signal loop problem. Conceptually, full duplex relays can be treated as special reflectors: they can amplify and retransmit whatever has been received. If no additional controls are performed, two full duplex relays may form an infinite loop: signals keep bouncing between them and signal strength may keep increasing until saturation due to the amplification made by relays. We call a system that is free from this infinite loop as a **stable system**.

*3.1.1 Sequential training.* We first define ***loop-back interference*** as: the signals (interference) received by a node that originated from itself, possibly having gone through other relays. Loop-back interference has an important feature: *a node can cancel its own loop-back interference*. This is because, similar to self interference cancellation, if a node knows what has been transmitted and the channels of the loop-back paths, it can construct the expected loop-back interference, and subtract it from the ensemble of the received signals. Intuitively, we want to cancel some of the loop-back interference to eliminate infinite loops in the system. We propose sequential training as a solution to get a stable system.

**Sequential training procedure:** Let $Ts = [R_1, R_2, ..., R_n, S_2, S_1]$ be a sequence in which nodes train for loop-back interference one after another. Once a relay node finishes training and is able to estimate its channel, it switches to full duplex relay mode to forward the signals it receives. Note that besides the self interference channel, $\boldsymbol{h}$ will also capture the channels of the relaying paths through those trained earlier than the node. Loop-back interference cancellation based on $\boldsymbol{h}$ and proper power amplification are performed while forwarding. Source nodes $S_i$ $(i = 1, 2)$ do training similarly after all relays have finished this procedure, and then change back to source mode. In the following, we refer the sequence $Ts$ as *training sequence*.

Although our solution bears similarities to the backward training process proposed in [7], it is very different in principle. One of our key finding is: *the sequential training can be performed with any sequence, in order to get a stable system*. In [7], the authors imply that the training will take place starting from the relay closest to the destination node, then move towards the source node. However, in the bidirectional case, both sides are source nodes and destination nodes. There is no concept of forward or backward direction. We believe that our views of loop-back interference differ sharply from the views on interference in [7], where the authors categorized the interference into forwarder interference, cross-hop interference, causal interference, non-causal interference etc.

In the following explanation of why this sequential training works, we assume that correct power amplification has been applied and no noise or residual interference is present in the system.

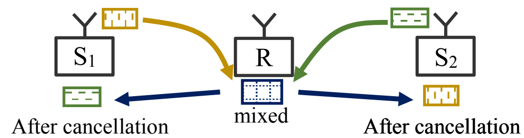



**Figure 2: A one-relay system that supports simultaneous bidirectional cut-through transmissions.**

Consider the one-relay system shown in Fig. 2. Following the training procedure, relay $R$ will first perform channel estimation. In this case, it is same as traditional self interference estimation since there are no other relays to forward the signal. After this, it changes its mode to full duplex relay. Then source nodes perform training. Without loss of generality, we assume source node $S_2$ first starts this process. For $S_2$, besides self interference, it also receives a copy from relay $R$. These signals together form loop-back interference, which can be canceled at $S_2$. After the other source node $S_1$ undertakes the same process, $S_1$ and $S_2$ start data transmissions concurrently in the same frequency band.

$R$ receives a mix of the two signals, which will then be amplified and transmitted by $R$. Thanks to the loop-back interference cancellation performed at the source nodes, $S_1$ will get only the signal from $S_2$ after cancellation and vice versa.

Now we extend the analysis to the multiple relays scenario using induction. Note that one-relay system is a stable system. Assume that there is a stable system cluster $C$ with $n$ $(n \geq 1)$ relays, as illustrated in Fig. 3. We introduce the $(n+1)$-th relay $R_{n+1}$ to the system, and perform the sequential training process after $C$, namely the training sequence $Ts = [C, R_{n+1}, S_i]$, we want to show that this system is still a stable system.

Recall that a stable system means no infinite loops inside the system. Therefore each sample input to the $n$-relay cluster $C$ will leave it within a certain time. Based on this observation, the relay cluster can be treated as a single relay. As shown in Fig. 3 with solid blue lines, for a given input directly from source nodes $S_i, i = 1, 2$, the cluster $C$ will give a certain output. This output, together with signals directly from source nodes (solid yellow line), will act as input to $R_{n+1}$. These signals get amplified and forwarded by $R_{n+1}$, which become another input to $C$. As shown with dashed green line, $C$ will give a corresponding output for this input. It must be noted that after the signal reaches $R_{n+1}$ for the first time, either from path $S_i \rightarrow R_{n+1}$ or $S_i \rightarrow C \rightarrow R_{n+1}$, it will get canceled at $R_{n+1}$ after the signal travels back again, thanks to the training and cancellation process. Because of this loop-back interference cancellation, the newly introduced relay $R_{n+1}$ does not bring any infinite loop and thus this $(n+1)$-relay system is still a stable system, which again can be treated as a single relay. By induction, we conclude that BiPass is always a stable system by doing sequential training.

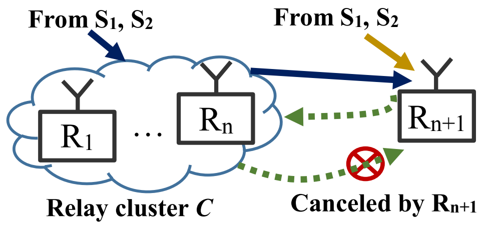



**Figure 3: Putting one more relay to a stable system still yields a stable system. All possible signal propagation paths are shown in the figure.**

Note that we made no assumptions on the physical locations of the relays in the above analysis. This leads to our key observation: *any training sequence can result in a stable system by doing sequential training.*

*3.1.2 Valid paths.* Since BiPass is a stable system that contains no loops, we can list all the possible paths from one side to another. In fact, for Fig. 3, all 6 possible paths from $S_i$ to $S_j$ ($i, j = 1, 2, i \neq j$) can be listed as: $S_i \rightarrow S_j$, $S_i \rightarrow C \rightarrow S_j$, $S_i \rightarrow R_{n+1} \rightarrow S_j$, $S_i \rightarrow C \rightarrow R_{n+1} \rightarrow S_j$, $S_i \rightarrow R_{n+1} \rightarrow C \rightarrow S_j$, and $S_i \rightarrow C \rightarrow R_{n+1} \rightarrow C \rightarrow S_j$. We call these paths *valid paths*. One can thus recursively list all valid paths in BiPass with any number of relays. Note that due to cancellation, not all combinations of nodes give valid paths. For example, $S_i \rightarrow R_{n+1} \rightarrow C \rightarrow R_{n+1} \rightarrow S_j$ is *invalid*.

**Longest valid path:** If the delay spread is larger than the cyclic prefix (CP) length, inter-symbol interference (ISI) will be observed. Hence, delay spread is worth investigating since it limits the number of relays for a given CP length (and symbol length if we keep the CP ratio the same). Consider the worst case where all nodes are in a single collision domain, the longest valid path (in terms of number of hops) will determine the delay spread since the fastest link is the direct link from one source to another.

In fact, from the above analysis, one may notice that by adding a new relay, the longest valid path changes from $S_i \rightarrow Longest(C) \rightarrow S_j$ to $S_i \rightarrow Longest(C) \rightarrow R_{n+1} \rightarrow Longest(C) \rightarrow S_j$, where $Longest(C)$ contains the longest path that goes through $C$. In other words, each time a new relay node is added to the system, a signal can go through all the possible paths again after visiting this newly added node. This means that the length of longest path will be doubled. Generally, the delay spread via $n$ relay nodes is

$$T_d = (2^n - 1)\Delta t, \quad (1)$$

where $\Delta t$ is the latency introduced at each relay node.

This indeed limits the number of relays that can be used in a route. However, in practice, the longer paths may be too weak to make a difference. For example, in the above mentioned longest path, the path loss between $R_1$ and $S_2$ may be relatively high (otherwise there is no need to choose $R_2$ and $R_3$ as additional relays), so the power delivered by this path is deemed to be relatively low. In fact, if we assume that each node has a limited interference range of 2 hops, for example, node $R_1$ may only be able to reach $R_3$ directly but not $R_4$, then we find that $T_d = O(n^2)$. Even if we consider all paths, with $\Delta t = 50\ ns$ [4] and acceptable threshold be 0.8 $\mu s$ (based on 802.11 standard CP length), we can support up to 4 relays without introducing any ISI based on Eq. 1. To employ more relays, one may use longer OFDM symbols to cover the delay spread.

## 3.2 Power Amplification

We have shown that one can get a stable system by doing sequential training. However, a full duplex relay needs to amplify the signal before forwarding in practice. Otherwise, the transmitted signal tends to be too weak for the other side to receive. We find that choosing this amplification coefficient is not an easy task.

*3.2.1 Target and challenges.* Assume that a relay receives signal $y = hx + n$, where $hx$ is the intended signal after cancellation and $n$ is the noise. It will multiply the signal by a real number $\sqrt{A}$ and transmit $b = \sqrt{A} \cdot y$. We call this $A$ as the amplification coefficient. $A$ must be chosen in a way that maximizes the throughput of the link, while subject to the following two constraints:

**Constraint 1. $|b|$ has an upper limit.** Usually there is a maximum magnitude limit on the signal a node can transmit. Each signal sample that is beyond the saturation limit suffers from clipping or other non-linear distortions, which in turn affects the decoding BER at the destination node. It has been shown that a larger clipping ratio can result in orders of increase in BER at the decoder [15]. In the following discussions, we assume the equivalent power constraint is 20 dBm.

**Constraint 2. $A$ shall be fixed once set.** One may think of a dynamic adaptation scheme to keep adjusting the value of $A$ to meet the above requirement. However, this approach can be hard to implement. To see the problem, assume a relay $R_1$ trains first followed by nodes $R_i$. If $R_1$ changes its $A$ on the go, unless we somehow inform *all* $R_i$ of this change, the cancellation performed at $R_i$ will fail since it has the same effect as the channel suddenly changes. Also, this approach may introduce large overhead and long converge time.

Next we consider two naive solutions with the network shown in Fig. 4. For ease of explanation, in the following discussions, we assume 1) the received signal power is dominated by the strongest signal strength; 2) a node can only reach one-hop away nodes and cause no interference to others. Even with these strong assumptions, how to properly assign amplification coefficients turns out to be a nontrivial problem as we will see below due to the above constrains. We will remove these assumptions in our solution presented in Sec. 3.2.3. We assume $R_1$ trains before $R_2$ in Fig. 4 and the maximum transmission power is 20 dBm. The path loss between nodes are shown in the figure.

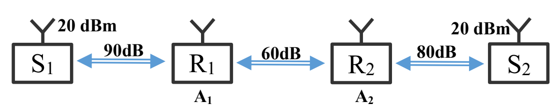



**Figure 4: A 3-hop network with path loss shown. Naive solutions will not provide optimal solutions in this toy example, as summarized in Table 1.**

**Naive solution 1: One-side dependency.** We first consider the one-side dependency scheme presented in [7]: a node should only consider its neighbor node on one side. For example, $R_1$ should only consider $S_1$ and $R_2$ should only consider $R_1$. As a result, $A_1 = 90$

**Table 1: Settings and results of possible solutions**

| Solution | $A_1$ | $A_2$ | Comments |
|---|---|---|---|
| One-side | 90 dB | 60 dB | $R_1$ Tx at 30 dBm. |
| Two-side | 60 dB | 60 dB | 20 dB SNR loss. |
| Our solution | 60 dB | 80 dB | $R_1$ trains first. |
| Our solution | 80 dB | 60 dB | $R_2$ trains first. |

dB and $A_2$ = 60 dB. While this may work for one-way cut-through transmissions, it is not a practical solution for bidirectional scenario. For example, consider the transmitting power at $R_1$ from $S_2$ to $S_1$, it can be calculated as 20 dBm - 80 dB + 60 dB - 60 dB + 90 dB = 30 dBm, which obviously violates Constraint 1.

**Naive solution 2: Two-side dependency.** To solve the problem of one-side dependency, one may think a node should consider both its neighbors and set the amplification coefficient based on the strongest link. In the toy example illustrated in Fig. 4, this scheme will lead to $A_1 = A_2 = 60$ dB. However, it turns out that this solution is a very conservative suboptimal solution, which will result in SNR loss. For example, consider the path from $S_1$ to $S_2$, the signal strength received at $S_2$ is -90 dBm, while our solution in Sec. 3.2.3 can give -70 dBm signal strength.

*3.2.2 Insight: effects of training sequence.* While maximizing throughput is our target, it is not easy to achieve directly. This is because signal strength and noise power are coupled in this system. Thus, we relax our target to maximize the signal strength received at intended node. We will maximize SNR by a route selection algorithm in Sec. 4.1 and investigate noise propagation in Sec. 5.1.

Intuitively, to get a better signal strength at the end nodes, the relays should always use their maximum transmitting power. Thus, if a node receives a total power equal to $P_{rx}$, the amplification coefficient $A$ should be chosen such that

$$A \cdot P_{rx} = P_{max}. \quad (2)$$

As a result, the task is mapped into identifying the received power at relay nodes. However, it is not easy. For a relay node, the received power is also affected by the choice of $A$ of the other nodes. Interference cancellation performed at relays further complicates the analysis. To see the problem, consider the topology shown in Fig. 4 and assume $R_1$ trains before $R_2$. Then node $R_2$ in fact intrinsically has an advantage: it will cancel the loop-back interference before amplification. This means that the path $R_2 \rightarrow R_1 \rightarrow R_2$ should not be considered when deciding the amplification coefficient of $R_2$. Instead, all valid paths from source nodes to $R_2$ that need to be taken into account are $S_1 \rightarrow R_1 \rightarrow R_2$ and $S_2 \rightarrow R_2$. By adding up the power of these two paths, we get $P_{rx2}$, and thus $A_2$. However, it must be noted that before assigning $A_2$ to $R_2$, we need to know the amplification coefficient $A_1$ at $R_1$ since it affects the received power from path $S_1 \rightarrow R_1 \rightarrow R_2$. For node $R_1$, on the other hand, it needs to consider paths including $S_1 \rightarrow R_1$, $S_2 \rightarrow R_2 \rightarrow R_1$ and $S_1 \rightarrow R_1 \rightarrow R_2 \rightarrow R_1$. Again, we need the value of $A_2$ in order to decide $A_1$. It looks like we have a dilemma: the choice of $A_1$ and $A_2$ depend on each other.

Our key insight is that *we can first assume that some nodes have already configured their amplification coefficients properly based on Eq. 2.* In this example, if we first assume $A_2$ has been set such that

**Algorithm 1:** Amplification setting algorithm

1 **Input: a)** Training sequence $Ts = \{R_1, \ldots, R_n, S_2, S_1\}$; **b)** Channel gains between any pair of participating nodes.
2 **for** $i \in [1, n]$ **do**
3 $\quad$ *Paths* ← *findPaths*(*src* = *Ts*[$i+1, i+2, \ldots$], *relay* = *Ts*[$1, \ldots, i-1$], *dest* = *Ts*[$i$]);
4 $\quad$ $P_{rx}$ ← *receivedPower*(*Paths*) ;
$\quad$ `// A vector of received power from each path`
5 $\quad$ $P_{rx}$.*extend*([$n_i, I_i$]) ;
$\quad$ `// Noise and residual interference`
6 $\quad$ $A_i \leftarrow P_{max}/\sum P_{rx}$ ;

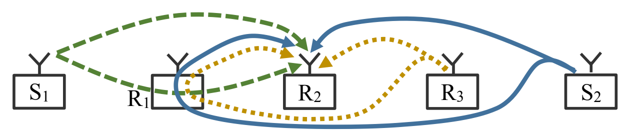



**Figure 5: A 4-hop network. The paths used to calculate $A_2$ for $R_2$ are shown. $S_1$, $S_2$ and $R_3$ are considered as source nodes and $R_1$ as relay.**

$R_2$ transmits at $P_{max}$, then for $R_1$, $R_2$ can be treated as a source node ($R_2$ will no longer relay signals). To get $A_1$, we consider paths $S_1 \rightarrow R_1$ and $R_2 \rightarrow R_1$ ($S_2$ has no path to $R_1$ since $R_2$ is no longer relay). As a result, we get $A_1$ = 60 dB. Then we can get $A_2$ = 80 dB with $A_1$ in hand, with which $R_2$ indeed transmits at $P_{max}$.

It is not hard to see that when setting $A_i$, we can only assume that nodes trained later than $R_i$ have properly configured their amplification coefficients due to interference cancellation. In other words, the setting of $A_i$ should follow the same sequence $Ts$. This implies that *the training sequence makes a difference on the choice of A*. In fact, if the training sequence is $[R_2, R_1]$ in this example, then $A_1$ = 80 dB and $A_2$ = 60 dB. A summary of power setting schemes and results are listed in Table 1. We will further investigate the effects of training sequence on system performance in Sec. 5.2.

*3.2.3 Our solution.* Based on these observations and analysis, we propose a simple and practical solution to the amplification setting problem, as shown in Alg. 1. As an illustration, we consider the 4-hop network as shown in Fig. 5 and focus on the setting of $A_2$ for $R_2$.

According to the training sequence $Ts$, we start the training process from the first relay node in $Ts$ to the last one. For each relay $R_i$, it first find all valid paths from nodes that have not yet performed training ($R_{i+1}, R_{i+2}, \ldots, R_n$) as well as the source nodes to itself $R_i$ (Line 3). For example, assume $Ts = [R_1, R_2, R_3]$ and $R_1$ has finished this process and configured $A_1$ accordingly, Fig. 5 plots all the paths for $R_2$ that need to be taken into account to set power amplification. In this case, $S_1$, $S_2$ and $R_3$ are treated as source nodes and $R_1$ as relay.

Once a node gets these paths, it can calculate the received power from each of these paths (Line 4). Note that all source nodes are assumed to have the maximum transmission power $P_{max}$. For example, the received power at $R_2$ from the path $R_3 \rightarrow R_1 \rightarrow R_2$ in Fig. 5 can be calculated as $P_{max} L_{R_3,R_1} A_1 L_{R_1,R_2}$, where $L_{R_i,R_j}$ is the path loss from node $R_i$ to node $R_j$.

A node also need to take noise and residual interference at each relay into consideration since they also contribute to the received power (Line 5).

Finally, a node set its amplification coefficient based on the total received power (Line 6). Now in Fig. 5, we have $A_1$ and $A_2$ calculated. After this, $R_3$ start to repeat this process.

*3.2.4 Overhead analysis.* **Communication overhead:** Once the training process is finished and the link is established, multiple packets can be exchanged for a long time. Thus, the training overhead is amortized. **Computation overhead:** In practice, the number of relays in a system is limited. This is mainly because the noise propagation issue as we analyze in Sec. 5.1, especially when interference cancellation is not perfect, which increases the noise floor at each node. Also, a large number of relays require very long OFDM symbols as mentioned in Sec. 3.1.2, which may not be desired in practice, especially when we want to make the system standard compliant. Besides, not all nodes are connected in reality due to the large path loss. This will reduce the number of paths on Line 3. For these reasons, computation overhead is not a primary concern in practice.

# 4 HARNESSING BIPASS IN NETWORKS

In previous sections, bidirectional cut-through transmissions are made possible via interference management and power amplification algorithms. In this section, we investigate how to harness BiPass in wireless networks.

## 4.1 Routing selection

In traditional wireless routing designs (e.g., [10]), information can only be forwarded from the source to the destination via relays hop by hop. The feasibility of cut-through breaks this basic assumption, and thus leads to an interesting open problem: *How can we select the relays such that the end-to-end throughput is maximized in a cut-through enabled wireless network?*

The problem has been challenging when only a single pair of nodes $S_1$ and $S_2$ are considered. Given a set of relays $\mathcal{R}$, there can be a large number of valid paths from $S_1$ to $S_2$ via $\mathcal{R}$, which jointly determine the end-to-end SNR. To make the problem tractable, we leverage the analytical results in Sec. 5 that, for a given relay set $\mathcal{R}$, the end-to-end SNR from $S_1$ to $S_2$ can be approximated by the strongest valid path via $\mathcal{R}$. Note that the signal strength along a given path is bottlenecked by the weakest link along that path. Then, the end-to-end SNR via a relay set $\mathcal{R}$ can be given by Eq. 3.

$$SNR(\mathcal{R}) = \max_{path \in \sigma(\mathcal{R} \bigcup \{S_1, S_2\})} \min_{(N_i, N_j) \in path} G_{N_i, N_j}, \qquad (3)$$

where $\sigma(\cdot)$ denotes the set of all permutations of $\mathcal{R}$ with $S_1$ and $S_2$ as the first and last elements respectively, $N_i \in \mathcal{R} \bigcup \{S_1, S_2\}$, and $G_{N_i,N_j}$ donates the SNR between node $N_i$ and node $N_j$.

For a single transmission pair $(S_1, S_2)$, maximizing the throughput is equivalent to maximizing the end-to-end SNR. Therefore, the **Max-SNR** routing selection for $(S_1, S_2)$ can be formulated as follows.

$$\max_{\mathcal{R}} SNR(\mathcal{R}). \qquad (4)$$

When there are many concurrent transmission pairs in the network, MaxSNR routing can be problematic. From Eqs. 3 and 4, it can be observed that many relay sets $\mathcal{R}$ could potentially maximize Eq. 4 and the number of active relays is not necessarily minimized. This leads to two drawbacks: 1) one cut-through route may incur large interference to other ongoing transmissions; 2) The increase in delay spread may cause inter-symbol interference.

To address the above issues, we propose **MaxSNR-MinHop** routing, which could achieve at most a small constant SNR gap to the Max-SNR routing but at the same time minimizes the number of active relays.

In MaxSNR-MinHop, each node $N_i$ maintains a routing table $\Gamma_{N_i}$ to all other nodes in the network. The key idea of MaxSNR-MinHop is that the routing tables are constructed for a set of discrete SNR thresholds. Each SNR thresholds ideally corresponds to the minimum SNR required for successful decoding for a given transmission rate in the physical layer. In practice, each SNR threshold could be set slight higher than the ideal SNR threshold to allow imperfect interference cancellation. As will be shown in Sec. 6, the practical SNR threshold is very close to the ideal case. For a destination $N_j$ and a given SNR threshold $SNR_{thres}$, node $N_i$ stores the next-hop forwarder, denoted by $\Gamma_{N_i}(N_j, SNR_{thres})$, which minimizes the number of hops from node $N_i$ to node $N_j$ such that the end-to-end SNR is no less than the threshold $SNR_{thres}$. To illustrate the idea precisely, let $\Psi_{N_i}(SRN_{thres})$ denote the set of neighboring nodes of $N_i$ such that the SNR between $N_i$ and $N_j$ is above $SNR_{thres}$.

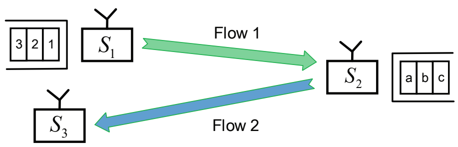
$$\Psi_{N_i}(SRN_{thres}) \triangleq \left\{ N_j | P_{max} G_{N_i,N_j} \geq SNR_{thres} \right\}.$$

Then, based on Eq. (3), the next-hop forwarder of node $N_i$ must be in $\Psi_{N_i}(SRN_{thres})$, and thus $\Gamma_{N_i}(N_j, SNR_{thres})$ can be given by

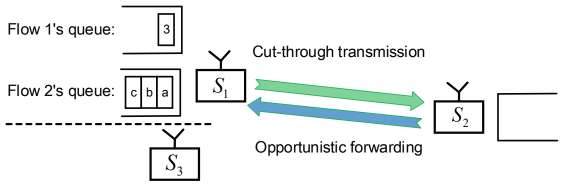
$$\Gamma_{N_i}(N_j, SNR_{thres}) = \begin{cases} Null & \text{if } \forall N_k \in \Psi_{N_i}(SRN_{thres}), \\ & \Gamma_{N_k}(N_j, SNR_{thres}) = Null. \\ \arg\min_{N_k \in \Psi_{N_i}(SRN_{thres})} & \Gamma_{N_k}(N_j, SNR_{thres}) \quad \text{otherwise.} \end{cases} \qquad (5)$$

Given the routing tables constructed (distributed construction of the routing table will be discussed), a source node $S_1$ could first check the maximum SNR threshold such that $S_2$ can be reached. Then $S_1$ starts constructing the route to $S_2$ by communicating the corresponding next-hop forwarder for the SNR threshold. The SNR threshold must also be sent to the forwarder such that the route to the destination can be constructed recursively hop by hop. Note that by increasing the number of SNR thresholds, the achieved SNR is within an arbitrarily small constant gap to the maximum SNR given by Eq. 4, the correctly constructed routing tables allows us to approximate MaxSNR routing with minimum number of hops. The key question left is how to construct the routing tables?

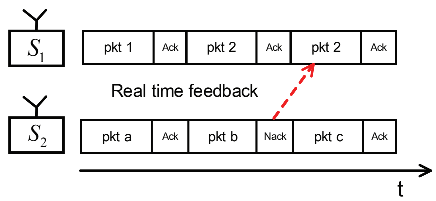
**Distributed routing table construction:** The major step in MaxSNR-MinHop is to construct the routing tables at each node in a distributed manner. Initially, each node only knows the list of neighboring nodes and their corresponding SNR thresholds. All the nodes periodically broadcast their routing tables to neighboring nodes. Once a new routing table is received from a neighbor $N_j$, the node $N_i$ updates its routing table as follows. For a destination $N_k$ and a threshold $SNR_{thres}$, if the total number of hops from $N_i$




**(a) $S_1$ want to transmit to $S_2$ while $S_2$ has data for $S_3$.**




**(b) $S_2$ opportunistically forward its packet to $S_3$.**




**(c) Real-time feedback enables more timely error controls.**

**Figure 6: Illustration of opportunistic forwarding and real-time feedback.**

to $N_j$ and from $N_j$ to $N_k$ is smaller than the known minimum hops from $N_i$ to $N_k$ for $SNR_{thres}$, in the routing table of node $N_i$, the next-hop to node $N_k$ for threshold $SNR_{thres}$ can be updated by the next-hop to node $N_j$ for threshold $SNR_{thres}$. The routing tables are exchanged among the nodes until all the routing tables converge. Although to maintain a small gap to the maximum SNR, more SNR thresholds are desirable, we note that the communication overhead of routing table construction in MaxSNR-MinHop increases linearly with the number of SNR thresholds. Fortunately, the communication overhead is not a major problem for two reasons: 1) in practice, there are only a limited number of feasible transmission rates, and thus a small number of SNR thresholds are sufficient; 2) since the routing tables need to be constructed only once, the communication overhead is amortized for the time when the network topology remains static.

**Route establishment:** The above route selection process is also a notification process: the relays will know whether it is selected to participate in the route. Similar to the designs in [7], once the route is determined, training and power setting process can start from the node closet to destination node since it has the information of all the participating relays.

## 4.2 Leveraging bidirectional transmissions

Compared with one-way cut-through, bidirectional cut-through brings most throughput and delay gains when the data traffic flows are also bidirectional. Then a natural question arises: *In a wireless network with limited or no bidirectional traffic flows, can we still achieve throughput and delay gains by leveraging bidirectional cut-through?* We answer the question affirmatively by proposing opportunistic forwarding and real-time feedback schemes, which will be illustrated through an example as shown in Fig. 6. In Fig. 6(a), there are two traffic flows in the network, where $S_1, S_2$ need to send streams of packets to $S_2$ and $S_3$, respectively.

*4.2.1 Opportunistic forwarding.* During the cut-through transmission from $S_1$ to $S_2$, $S_2$ checks its buffer, if there is any packet such that it is "closer" to the destination node from $S_1$ than that from $S_2$, $S_2$ forwards the packet to $S_1$. In the example shown in Fig. 6(b), $S_2$ forwards its packets to $S_1$ which is closer to $S_3$. Specifically in our scheme, we select our distance metric between two nodes $S_2$ and $S_3$ as the end-to-end SNR given by Eq. 4. Note that the distance from $S_2$ to $S_3$ is stored in the routing table of $S_2$, thus $S_2$ only needs to query $S_1$'s routing table about the distance from $S_1$ to $S_3$. Since bidirectional cut-through transmission is enabled between $S_1$ and $S_2$, such a query takes zero overhead. With opportunistic forwarding, bidirectional cut-through can be utilized to increase the capacity of the network without the constraint of bidirectional traffic flows. As will be shown in the simulations in Sec. 6.3.2, the throughput gain can be substantial compared with one-way cut-through.

*4.2.2 Real-time feedback.* Bidirectional cut-through enables real-time feedback for the traffic in both directions. For example, in Fig. 6(c), the packet *pkt* 2 in the transmission from $S_1$ to $S_2$ is not successfully recovered at $S_2$. With real-time feedback, $S_2$ could piggyback a NACK instantly to $S_1$ such that $S_1$ could re-send *pkt* 2 with minimum delay. In comparison, in one-way cut-through, the lost packets cannot be recovered until the dedicated feedback slot after transmitting a block of packets. To mitigate routing establishment overhead, a large block size may be used in one-way cut-through. Thus, when there are packet losses, bidirectional cut-through can achieve a significant delay gain over one-way cut-through. The delay gain is more evident in real-time applications such as video streaming or gaming, where one received packet cannot be considered useful unless all packets with smaller indices have been successfully received. Furthermore, real-time feedback can also be utilized to dynamically adjust the data rates for both directions.

# 5 SYSTEM ANALYSIS

To better understand BiPass, in this section, we investigate some problems of the system.

## 5.1 Noise propagation

Noise at relays will get amplified together with the signals. A natural concern is that the noise accumulated from different paths at the end nodes may be too high to decode packets. In this section, we look into the problem of noise propagation in BiPass.

Consider the 2-relay topology shown in Fig. 7. Since source nodes $S_1$ and $S_2$ do not introduce additional noise at other nodes when transmitting, we assume they are in receiving-only mode so that we can focus on noise propagation. We also assume that the relays have set the amplification coefficients based on Alg. 1, and are in full duplex relay mode. Note that in this setting, each relay will transmit noise with power $A_i n_0$, where $n_0$ is the noise floor and $A_i$ is its amplification coefficient. We want to know how much noise can be observed at the end nodes in this scenario.

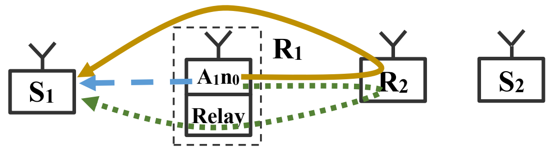



**Figure 7: All possible three paths from $R_1$ to $S_1$ are shown. They are the reverse of paths from $S_1$ to $R_1$.**

We first make the following **observation**: If channel reciprocity[2] holds between any two nodes without this system[3], then channel reciprocity holds between any two nodes when BiPass is set up and running. For example, in our system, if $S_1$ transmits at power $P_{tx}$ and the received signal power at $R_1$ is $P_{rx}$, then if $R_1$ transmits at power $P_{tx}$, $S_1$ will receive signal with power $P_{rx}$. The key insight is that all the paths from one node to the other node are exactly the reverse of all paths from the other node to this node. Since the path loss and amplifications are the same for all the path pairs, we get the above conclusion.

Recall that the amplification coefficient $A$ at relays are chosen based on the total power of received signals $P_{rx}$[4] as shown in Eq. 2. In other words, $A$ can be treated as the total signal power loss from two end nodes to this relay node. Now if a relay transmits noise with power $An_0$, the sum power received at two end nodes will be $(An_0)/A = n_0$. Since noise from different relays are uncorrelated, the above analysis can be performed for each relay. We get the following conclusion: the total additional noise power introduced at the end nodes

$$n_{add} = \sum_i n_i, \tag{6}$$

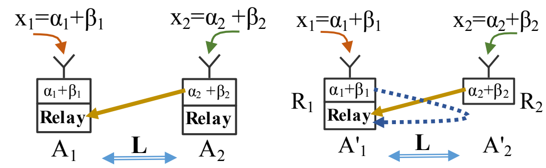
where $n_i$ is the noise floor at relay $R_i$. This conclusion implies that incorporating more relays, while it may increase signal strength, will also increase the noise floor at the end nodes.

Eq. 6 gives the total additional noise increase at both end nodes. For each relay $R_i$ , it can be shown that the noise distribution is the same as the received signal power from source nodes at that relay. For example, assume at relay $R_i$, the received signal power from $S_1$ and $S_2$ is $\alpha_i$ and $\beta_i$, respectively, then the additional noise due to $R_i$ observed at $S_1$ will be $\frac{\alpha_i}{\alpha_i+\beta_i} n_i$ and $S_2$ be $\frac{\beta_i}{\alpha_i+\beta_i} n_i$. We represent this ratio as $\phi_i := \alpha_i / \beta_i$.

## 5.2 Effects of training sequence

In Sec. 3.1.1, we conclude that any training sequence can yield a *stable* system. Later in Sec. 3.2.2 we analyze the effects of different training sequences on amplification coefficient selection of relay nodes. It seems that different training sequences may lead to different SNR of received signal at end nodes. Surprisingly, we find that training sequence has no effect on the received SNR at end nodes.

We first focus on noise propagation in a two-relay system, as shown in Fig. 8. We concluded in Sec. 5.1 that the total noise introduced at the two end nodes by each relay is equal to the noise floor $n_i$ at that node. Also, the distribution $n_i$ is same as the received signal power components ratio $\phi_i$ from source nodes. We want to show that ratio $\phi_i$ does not vary with different training sequences.




**(a) Sequence $[R_2, R_1]$.** **(b) Sequence $[R_1, R_2]$.**

**Figure 8: Received signal components at $R_1$ with different training sequences.**

Assume $x_i = \alpha_i + \beta_i$ is the external signal power (from source nodes in this case) received at relay $i$, where $\alpha_i$ and $\beta_i$ represents the signal power components from $S_1$ and $S_2$, respectively. The path loss between $R_1$ and $R_2$ is assumed to be $L$. With training sequence $[R_2, R_1]$ as shown in Fig. 8(a), assume Alg. 1 gives $A_i$ as the amplification coefficient for $R_i$; and with $[R_1, R_2]$ in Fig. 8(b), amplification coefficients are $A'_i$. Then the received signal power at $R_1$ in these two cases are

$$\begin{cases} P_{rx1} = x_1 + LA_2x_2 & (7a) \\ P'_{rx1} = x_1 + LA'_2x_2 + L^2A'_2A'_1x_1. & (7b) \end{cases}$$

Similarly, we can get the representations for $R_2$. Note that with Eq. 2, Eq. 7b can be rewritten as $P'_{rx1} = x_1 + LA'_2P'_{rx2} = x_1 + LP_{max}$.

The key insight here is that $A_i$ and $A'_i$ are related. Together with Eq. 2, we get

$$\begin{cases} P_{max} = A_2P_{rx2} = A_2(x_2 + LP_{max}) & (8a) \\ P_{max} = A'_1P'_{rx1} = A'_1(x_1 + LP_{max}) & (8b) \\ P_{max} = A'_2P'_{rx2} = A'_2(x_2 + LA'_1x_1). & (8c) \end{cases}$$

Plug the representations of $x_2$ and $x_1$ from Eq. 8a and Eq. 8b to Eq. 8c, we get $A'_2 = A_2(1 + L^2A'_2A'_1)$. Substituting $A'_2$ in the second term of Eq. 7b, we get

$$P'_{rx1} = (1 + L^2A'_2A'_1)(x_1 + LA_2x_2). \tag{9}$$

Compared with Eq. 7a, Eq. 9 shows that the received signal at $R_1$ is simply scaled by $(1 + L^2A'_2A'_1)$ and thus maintains the same signal components and power ratio $\phi$. A similar reasoning can be performed for $R_2$. We conclude that *changing the training sequence in this system does not change the noise power observed at end nodes.*

Now we consider the received signal strength at end nodes. In fact, due to the amplification taken by relays, the output of $R_1$ based on Eqs. 7a and 9 are the same since they have the same signal components and output power $P_{max}$. Similarly, the output of $R_2$ also remains the same. So *the training sequence also has no effect on signal power received at end nodes.*

Extending to multiple relays scenario is straight forward. By treating this 2-relay system as a whole, the above analysis reveals that its output is same regardless of training sequence for a given external input (from source nodes and other relays). In other words, swapping the training sequence between $R_i$ and $R_{i+1}$ in a $n$-relay system does not affect the other nodes in the system. Since swapping can be performed between any two consecutive nodes in $T_s$,

[2]Here we only consider the channel gain.
[3]Especially with the single-antenna full duplex design [6], this assumption can be easily met.
[4]We also consider the noise $n_0$ at the relays (Line 5 of Alg. 1), but we ignore noise here since in most cases $P_{rx} \gg n_0$. The conclusion in Eq. 6 thus acts as an upper bound.

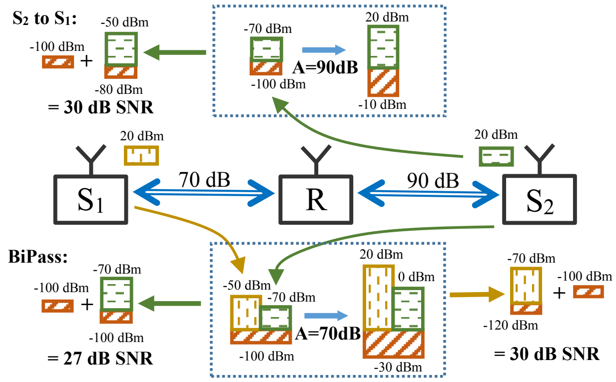



**Figure 9: A one-relay system with signal propagation process illustrated. For one-way transmission from $S_2$ to $S_1$, although a 20 dB higher $A$ is used at $R$, the SNR at $S_1$ only increases by 3 dB.**

we conclude that **the training sequence has no effect on SNR at end nodes**.

## 5.3 Are we doubling the capacity?

Another question is: compared with one-way cut-through systems, can BiPass double the channel capacity? Consider the one-relay network as shown in Fig. 9. We assume that the noise floor at all nodes are -100 dBm. For one-way cut-through, based on the one-side dependency scheme used in [7], when $S_1$ wants to transmit packets to $S_2$, we set $A = 70$ dB; for the direction $S_2$ to $S_1$, $A = 90$ dB. However, in the bidirectional case, $R$ can only give approximately 70 dB amplification (69.96 dB to be precise due to power addition) for both directions based on Alg. 1. *Does this mean we have a 20 dB SNR loss in the direction $S_2$ to $S_1$?*

In fact, in both systems, SNR is limited by the weakest link. As shown in Fig. 9, for $S_2$ to $S_1$ one-way transmission, $R$ receives a signal with -70 dBm power (30 dB SNR). Recall that in the amplification process, noise is also amplified. So, although $S_1$ receives -50 dBm signal but it also receives -80 dBm noise. The SNR at $S_1$ is thus still limited by 30 dB. Similarly, we can see $S_2$ can also have a 30 dB SNR for the direction $S_1$ to $S_2$. While in our system, we get 30 dB SNR at $S_2$ and 27 dB SNR at $S_1$ due to the increase of noise floor.

We make the following **observations**: 1) SNR is basically limited by the weakest link in the path for both one-way and bidirectional transmissions, a larger $A$ does not necessarily mean a large SNR increase. 2) Our system can not exactly double the channel capacity compared with one-way cut-through full duplex systems, but close to double. This is due to the noise accumulation problem that we analyzed in Sec. 5.1.

## 5.4 Additional Issues

**Synchronization:** Clearly, we do not need time synchronization. For frequency synchronization, previous work [7] has shown that a signal goes through a full duplex relay will remain the same frequency. Thus frequency offset is also not a problem, which usually entails careful processing [19].

**Imperfect cancellation:** It is possible that due to reasons including channel estimation errors, channel changes, etc, the cancellation at relays or end nodes are imperfect in reality. Since residual interference can be treated as noise, it will have a similar behavior as our analysis in Sec. 5.1. So, imperfect cancellation will only result in noise floor increase at end nodes.

**Data rate selection:** Feedback based scheme, the de facto rate selection scheme used today, is clearly one option for our system. Based on the packet reception rate, the source node can adjust its rate accordingly. In fact, since real-time feedback is possible in our system, a much faster reaction is expected than traditional half duplex systems. Alternatively, one can take advantage of the training process performed by the other source node to decide the data rate.

# 6 EVALUATIONS

To evaluate the performance of BiPass in the real world, we build a prototype with NI PXIe-1082 software defined radio platform with NI-5791 RF front end. Experiments are performed in a large office with 24 metallic cubicles inside. The carrier frequency is set to be 2.57 GHz to avoid external interference and the bandwidth used is 10 MHz. Our full duplex implementation follows the design in [13], a node has two antennas for dedicated transmitting and receiving. We think the design in [6], which uses one antenna for both transmitting and receiving, as well as considering nonlinear signal distortions, may further improve the performance of BiPass. The symbol size of OFDM is set to be 1024 samples (subcarriers) and CP ratio 0.2. We make it larger than 802.11 standard since our platform introduce a larger latency ($\sim 0.8\ \mu s$) when act as full duplex relays. The transmission power is set to be 0 dBm. Up to 3 relays are used in our experiments. Similar to the previous sections, we name the source nodes as $S_i$ and relay nodes as $R_i$.

## 6.1 Micro-benchmark

*6.1.1 Power amplification.* In Sec. 3.2.3, we propose an algorithm for setting the power amplification coefficient of each relay in BiPass. In this section, we investigate the performance of this algorithm, compared with the two naive solutions (one-side and two-side dependency) mentioned in Sec. 3.2.1.

Fig. 10 shows the CDF of the transmitting power used by all the relays in 3-relay systems with different power setting schemes. We find that with the one-side dependency scheme, the power at relays can easily go beyond the limit which means signal clipping happens very often, thus resulting in an increasing BER at decoder [15]. We also observe the cases where amplification is so high that it exceeds our hardware limitation, in turn results in an unstable system. We utilize 20 dBm as transmission power to represent these cases in the figure. For two-side dependency solution, the transmitted power is conservative in all experiments. We observe the cases where relays use power 17 dB less power for transmission, which means large SNR loss as we analyzed in Sec. 3.2.1. With our proposed solution in Alg. 1, the transmitting power at relays are within ± 2 dB to its maximum power which is tolerant in reality. The small variances are mainly caused by path loss estimation errors, uncalibrated analog power amplifier, etc.

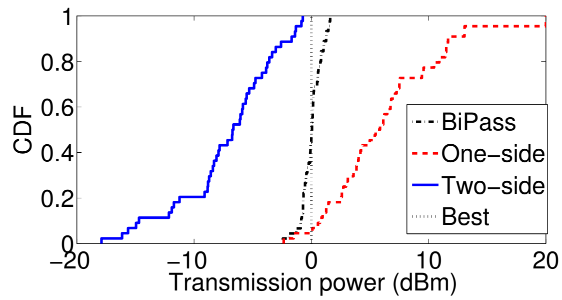



**Figure 10: Transmission power used at relay nodes in a 3-relay system with different power schemes.**

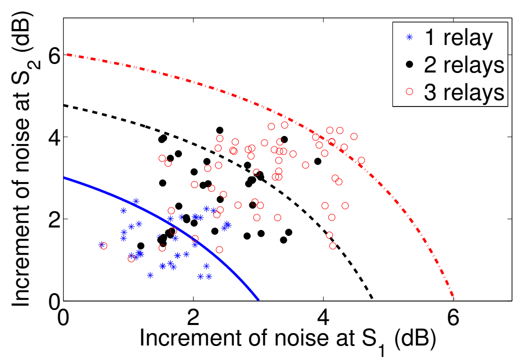



**Figure 11: Increment of noise floor at each end node. The lines represent the theoretical values.**

6.1.2 *Noise propagation.* In this part, we do experiments to record the noise floor increase observed at the end nodes. For each experiment, we set up the link using sequential training procedure and set amplification coefficients with Alg. 1, end nodes are in receiving-only mode without transmitting any signal. We record the noise floor before and after the link set up and calculate the noise increase. Fig. 11 shows this noise increase observed by each end node with 1 to 3 relays in between. We also plot the theoretical values based on Eq. 6. Note that in our implementation, each node contains two antennas for dedicated transmission and reception, which means the channel reciprocity assumption made in Sec. 5.1 is not fulfilled. Even in this case, we can observe that the noise floor at end nodes increases with the number of employed relays, and the difference between theoretical values are within several dB.

6.1.3 *Effects of the training sequence.* To evaluate the effects of the training sequence, we record the SNR observed at end nodes with different training sequences in 3-relay systems. Fig. 12 shows the signal strength and noise power at end nodes in a typical trace. Again because of our two-antenna node design, we can see signal strength and noise power differences with different training sequences. Otherwise the signal strength observed by both end nodes should be the same with the same training sequence based on our channel reciprocity observation in Sec. 5.1. And for the same token, we observe that the signal and noise power tends to change in a

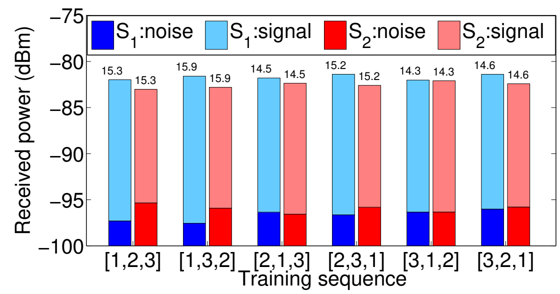



**Figure 12: Signal and noise power at end nodes with different $Ts$. SNR values are indicated on top.**

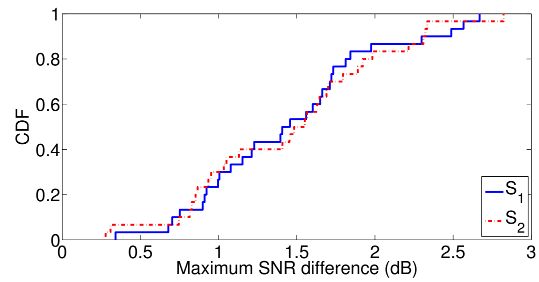



**Figure 13: Maximum SNR difference at end nodes with different $Ts$.**

correlated way. Nevertheless, the SNR difference is only within 2 dB in this experiment.

We further repeat the experiment for 30 times and record the largest difference between SNR values observed at end nodes with different training sequences. Fig. 13 shows that the difference is limited by 3 dB. This result implies that training sequence plays an insignificant role even without the channel reciprocity assumption we made in Sec. 5.2.

**Table 2: Decoding possibilities at relay nodes**

| Relay Node | $R_1$ | $R_2$ | $R_3$ |
|---|---|---|---|
| Decoding rate | 0/30 | 0/30 | 0/30 |

6.1.4 *Relay decodability.* Since relay nodes receive packets from both sides, they tend to have a much lower chance to decode packets. To validate this, we do experiments with 3-relay systems and try to decode packets at relay using successive interference cancellation. Table 2 shows that relays can not decode any packet in all experiments. However, in one-way cut-through system, the relays can 100 percent decode all the packets. This result is expected since signals at relays are mixed from two source nodes in two-way system. The decoder experiences troubles on packet detection and channel estimation. Since data rate is selected based on the target node, the interference at relays results in much higher BER because of reduced SNR. Similar to network coding based schemes, we think this feature of our system can be further explored for security applications.

## 6.2 System throughput

In this section, we compare the achievable throughput of BiPass with both half duplex *D&F* scheme and AirExpress. We run experiments in a total of 90 randomly chosen topologies with 1 to 3 relays being employed. In each experiment, we apply BiPass, AirExpress and *D&F* for comparisons. In the case of AirExpress and *D&F*, we enable data transfer of each direction separately. For *D&F*, we iterate all possible routes and choose the path that can maximize its throughput. For each scheme, we use the highest achievable physical layer data rates which can be supported by the system. In each experiments, 200 packets are transmitted. The communication overhead for training is excluded in the calculation of throughput.

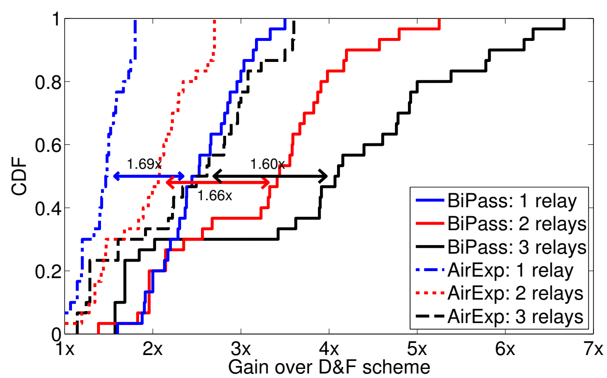



**Figure 14: Throughput gain over half duplex *D&F* scheme with BiPass and AirExpress. BiPass shows over 1.6x median gain over AirExpress.**

Fig. 14 shows the result of the throughput gain over *D&F* with AirExpress and BiPass. As we can see, both one-way and two-way systems can achieve a significant gain over *D&F*, and the gain increases with more relays. Specifically, BiPass shows median gains of 2.48x, 3.43x, and 4.09x over *D&F* with 1 to 3 relays in the system. Also, BiPass shows higher throughput over AirExpress since transmissions can happen in both directions concurrently with BiPass. We observe 1.69x, 1.66x and 1.6x median gain over AirExpress with 1 to 3 relays. The gain slightly decreases with number of relays due to the noise accumulation problem we analyzed in Sec. 5.1. Also, there is small residual interference after cancellation in our system, which further increases the noise floor at end nodes.

## 6.3 BiPass in networks

In order to understand the benefits of bidirectional cut-through in large wireless networks, we evaluate BiPass via simulations. In the simulations, 100 nodes are randomly placed in a $1000m \times 1000m$ field. Since one can expect that BiPass leads to a higher throughput gain when flows are always in both directions between a pair of nodes, we set up a challenging scenario for BiPass, where 10 traffic flows are generated between randomly picked nodes such that most of the traffic is not bidirectional. For BiPass, the power amplification is controlled by Alg. 1, while in other schemes each radio operates at its maximum power 20 dBm. 8 discrete data rates from 802.11g are used in the simulations. The residual interference due to imperfect cancellation and noise propagation are measured from the experiments and then fed into the simulator. To compare BiPass with existing techniques, we implement two benchmark schemes, i.e., the traditional *D&F* scheme with half duplex radios [10] and the one-way cut-through scheme AirExpress [7]. To avoid inter-symbol interference, the maximum number of relays for both BiPass and AirExpress is set to be 4. Throughout the simulations, the MAC layer employs a simple CSMA mechanism such that two routes/flows can be transmitted concurrently only when their mutual interference is marginal. We apply 100 random topologies in each of the following network simulations.

*6.3.1 Different routing schemes.* To study the performance of routing selection algorithms, we implement MinHop routing [10], our proposed MaxSNR routing and MaxSNR-MinHop routing algorithms. In MinHop routing, the routes are selected to minimize the expected number of hops.

Fig. 16 shows the impact of different routing algorithms on BiPass. It can be observed that the performance of MinHop is very bad for BiPass. This is because the routes selected by MinHop do not maximize the end-to-end SNR and thus is a poor metric for bidirectional cut-through. Compared with MaxSNR routing, MaxSNR-MinHop could achieve an additional 14% throughput gain. The major reason is that by minimizing the number of active relays, one transmission incurs less interference to the network and there might be more concurrent transmissions in the network.

*6.3.2 Throughput gain of BiPass.* In Fig. 15, we compare the achievable throughput of BiPass with AirExpress and the traditional *D&F* schemes. First, AirExpress and BiPass could achieve much higher throughput than *D&F*. The reason is that in *D&F*, when a node is transmitting, all its neighbor nodes cannot transmit to avoid incurring interference. As a result, the interference constraint greatly limit the achievable throughput of *D&F*. Second, BiPass achieves a median throughput gain of 1.36x over AirExpress. The throughput gain comes from two factors: 1) BiPass is able to leverage opportunistic forwarding while AirExpress could not; 2) BiPass uses a better routing selection algorithm than AirExpress. Our results suggest that even when the traffic flows are not bidirectional, BiPass still leads to substantial throughput improvement.

*6.3.3 Delay performance of BiPass.* Fig. 17 shows the delay performance of BiPass, AirExpress and *D&F*. First, the delay of AirExpress and BiPass is much lower than that of *D&F*. The reason is that in *D&F*, the delay of packets quickly accumulate hop by hop, while one cut-through transmission can cross multiple hops. Second, attributed to the opportunistic forwarding, BiPass achieves much lower delay than AirExpress. Specifically, the median delay reduction is 47%.

*6.3.4 Delay gain v.s. block size.* To mitigate the communication overhead of establishing a route, it is reasonable to assume that packets are transmitted in blocks for both AirExpress and BiPass. Fig. 18 plots the in-order delay performance in a network with a single traffic flow. In-order delay means a received packet cannot be considered useful unless all packets with smaller indices have been successfully received. The end-to-end packet loss probability is 0.1. Fig. 18 reveals that with real-time feedback enabled by BiPass, the in-order delay of BiPass is a constant independent of the block

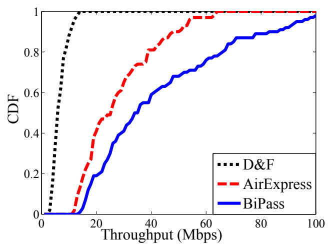



**Figure 15: Throughput gain of BiPass over AirExpress and *D&F* schemes.**

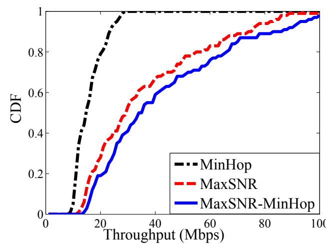



**Figure 16: Throughput of BiPass with different routing selection schemes.**

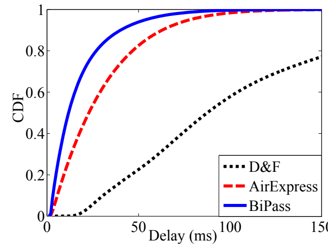



**Figure 17: Delay gain of BiPass over AirExpress and *D&F* schemes.**

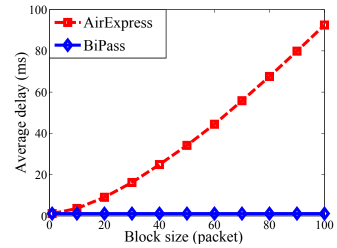



**Figure 18: In-order delay performance when packets are transmitted in blocks.**

size. Note that the delay of BiPass in Fig. 18 is much less than that in Fig. 17, this is because there is only a single flow in this new scenario, thus there are no contention latencies. On the other hand, for AirExpress, since feedback is only available after transmitting the whole block of packets, the delay increases significantly with the block size. This suggests that BiPass is much more appropriate for real-time applications such as video streaming or gaming.

## 7 DISCUSSION

In this section, we discuss some practical issues with BiPass and how we can further improve its performance.

**Integrating with MIMO:** MIMO can significantly improve the performance of wireless systems by transmitting multiple streams simultaneously. Full duplex MIMO has been shown possible in [3, 5]. By carefully choosing relays, or using relays with multiple antennas, we think BiPass can be extended to use MIMO.

**More general topologies:** Our current design works for two end nodes to exchange information. As extensions of our work, more general scenarios can be considered. For example, to support multiple nodes on each side, or to allow multiple simultaneous flows on overlapped paths. We think possible directions to realize these scenarios are using multiple antennas, multiple channel bands, etc.

**Mobile scenario:** Our current design considers only static environment. A possible extension of our work is to consider mobile scenario. Clearly, with our current design, a node would need to estimate channels frequently in this scenario, which means high overhead. It is possible that some of the nodes do not need to retrain when only one node moves, how to take advantage of this fact may help to extend BiPass to mobile scenario.

**Other forwarding schemes:** In our design of BiPass, amplify-and-forward is essentially implemented. Other forwarding schemes such as quantize-map-forward [11] may be explored to mitigate issues like noise propagation. However, how to do interference cancellation with these schemes is challenging.

**Scheduling in networks:** In our current network design, flows are CSMA based. We think how to optimally schedule flows in networks, especially considering inter-flow interference, is an open problem to solve.

## 8 CONCLUSION

In this paper, we present BiPass, the first system that supports simultaneous bidirectional cut-through transmissions in the same frequency band via multiple full duplex relays. Our end-to-end full duplex system can be treated as a generalized version of the current full duplex system, where we enable full duplex communications between two far away nodes. While our system focus on WiFi settings, the same approach can be extended to LTE, 60 GHz, etc. We believe that our design will inspire more researches and act as an enabling technique for many more exciting applications in the future. For example, [1] utilizes a full duplex relay at 60 GHZ to improve performance of virtual reality. Our system may further boost the performance by enabling multiple such relays.

## ACKNOWLEDGEMENTS

We would like to thank our reviewers for their comments and suggestions to improve our paper. This work has been partially supported by NSF grants CNS-1254032, CNS-1302620, EARS-1547306, CNS-1514260, CNS-1314538, CNS-1518829, Army Research Office W911NF-14-1-0368, and Office of Naval Research N00014-15-1-2166.

## REFERENCES

[1] Omid Abari, Dinesh Bharadia, Austin Duffield, and Dina Katabi. 2016. Cutting the cord in virtual reality. In *Proceedings of the 15th ACM Workshop on Hot Topics in Networks*. ACM, 162–168.

[2] Rudolf Ahlswede, Ning Cai, S-YR Li, and Raymond W Yeung. 2000. Network information flow. *IEEE Transactions on information theory* 46, 4 (2000), 1204–1216.

[3] Ehsan Aryafar, Mohammad Amir Khojastepour, Karthikeyan Sundaresan, Sampath Rangarajan, and Mung Chiang. 2012. MIDU: enabling MIMO full duplex. In *Proceedings of the 18th annual international conference on Mobile computing and networking*. ACM, 257–268.

[4] Dinesh Bharadia and Sachin Katti. 2014. FastForward: fast and constructive full duplex relays. In *ACM SIGCOMM Computer Communication Review*, Vol. 44. ACM, 199–210.

[5] Dinesh Bharadia and Sachin Katti. 2014. Full duplex MIMO radios. In *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*. 359–372.

[6] Dinesh Bharadia, Emily McMilin, and Sachin Katti. 2013. Full duplex radios. In *ACM SIGCOMM Computer Communication Review*, Vol. 43. ACM, 375–386.

[7] Bo Chen, Yue Qiao, Ouyang Zhang, and Kannan Srinivasan. 2015. AirExpress: Enabling Seamless In-band Wireless Multi-hop Transmission. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 566–577.

[8] Bo Chen, Vivek Yenamandra, and Kannan Srinivasan. 2015. FlexRadio: Fully Flexible Radios and Networks.. In *NSDI*. 205–218.

[9] Jung Il Choi, Mayank Jain, Kannan Srinivasan, Phil Levis, and Sachin Katti. 2010. Achieving single channel, full duplex wireless communication. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*. ACM, 1–12.

[10] Douglas SJ De Couto. 2004. *High-throughput routing for multi-hop wireless networks*. Ph.D. Dissertation. Massachusetts Institute of Technology.

[11] Melissa Duarte, Ayan Sengupta, Siddhartha Brahma, Christina Fragouli, and Suhas Diggavi. 2013. Quantize-map-forward (QMF) relaying: An experimental study. In *Proceedings of the fourteenth ACM international symposium on Mobile ad hoc networking and computing*. ACM, 227–236.

[12] Kai-Cheng Hsu, Kate Ching-Ju Lin, and Hung-Yu Wei. 2016. Full-duplex delay-and-forward relaying. In *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. ACM, 221–230.
[13] Mayank Jain, Jung Il Choi, Taemin Kim, Dinesh Bharadia, Siddharth Seth, Kannan Srinivasan, Philip Levis, Sachin Katti, and Prasun Sinha. 2011. Practical, real-time, full duplex wireless. In *Proceedings of the 17th annual international conference on Mobile computing and networking*. ACM, 301–312.
[14] Sachin Katti, Shyamnath Gollakota, and Dina Katabi. 2007. Embracing wireless interference: Analog network coding. *ACM SIGCOMM Computer Communication Review* 37, 4 (2007), 397–408.
[15] Xiaodong Li and Leonard J Cimini. 1997. Effects of clipping and filtering on the performance of OFDM. In *Vehicular Technology Conference, 1997, IEEE 47th*, Vol. 3. IEEE, 1634–1638.
[16] Yue Qiao, Ouyang Zhang, Wenjie Zhou, Kannan Srinivasan, and Anish Arora. 2016. PhyCloak: obfuscating sensing from communication signals. In *2016 USENIX Annual Technical Conference (USENIX ATC 16)*. USENIX Association.
[17] Yang Yang and Ness B Shroff. 2015. Scheduling in wireless networks with full-duplex cut-through transmission. In *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2164–2172.
[18] Shengli Zhang, Soung Chang Liew, and Patrick P Lam. 2006. Hot topic: Physical-layer network coding. In *Proceedings of the 12th annual international conference on Mobile computing and networking*. ACM, 358–365.
[19] Wenjie Zhou, Tanmoy Das, Lu Chen, Kannan Srinivasan, and Prasun Sinha. 2016. BASIC: backbone-assisted successive interference cancellation.. In *MobiCom*. 149–161.