Technique Report:
SIM+: A Simulator for Full Duplex Communications

ABSTRACT

Recent studies have shown that wireless full duplex communication is possible with the help of multiple self-interference cancellation techniques. Previous work mainly focuses on the design of the physical layer system, and only study the performance between two nodes or a few nodes. The true potential of this new capability is still largely unknown in a general network setup. However, there are no large scale testbeds that do full duplex, hindering protocol and network system design developments with this new capability. Simulation has historically been key to making progress in the space of protocol and network system design.

This paper presents the first network simulator that is capable of supporting full duplex communication. We extend the state-of-the-art network simulator, ns-3 and present SIM+. We enable many key features for a full duplex node within SIM+. A set of such MAC layer features for the full duplex receiver includes: sending a packet back to the transmitter, broadcasting busytone, and sending a packet to a secondary receiver. We provide some preliminary results using SIM+. Some of these results are counter-intuitive.

1. INTRODUCTION

Full duplex wireless transmission in a single channel is shown to be feasible in recent work [3, 5, 6, 2, 4]. With the combination of several self-interference cancellation techniques, a wireless node could transmit and receive at the same time without using separate channels. Given that the available wireless spectrum is limited, this new physical layer technique is a promising direction for future wireless communication.

Previous work with full duplex implementations indicated that the full duplex system could achieve almost twice the throughput of a half duplex system. However, [3, 5] only studied the throughput of their full duplex design between two nodes. Experiments with 3 nodes were conducted in [6]. These results are limited by their topologies and do not characterize the full potential of full duplex technique. The performance of full duplex in a large network still remains unexplored. This paper seeks to build a simulator that can enable the community to start understanding the true potentials and limitations of this new – full duplex – capability. Such a simulator will also foster the design and development of future full duplex protocols and network architectures.

In this paper, we present the first full duplex network simulator, SIM+, based on the widely used network simulator, ns-3. SIM+ provides general full duplex operations that could be used in the future design of full duplex protocols. SIM+ is built based on the "wifi" module in ns-3, which implements the IEEE 802.11 standard. Since 802.11 is a half duplex system, the "wifi" module is designed without any thought to support transmitting and receiving at the same time. Enabling full duplex operation requires modifications in the physical layer implementation and creating a direct connection between the upper MAC layer and the physical layer. We provision many key features of full duplex in SIM+: a full duplex receiver could send a packet back to the transmitter, send a packet to a secondary receiver, or broadcast a busytone signal to protect the packet reception. These features have been introduced in several full-duplex MAC protocols [6, 7, 8]. A simple MAC with sending a packet back to the transmitter and broadcasting busytone is introduced in [6]. If a packet for the transmitter exists, the receiver sends it back. Busytone signal is padded at the end of transmission to make sure that both of the transmissions end at the same time. In [7], the transmitters initially send a half duplex packet. Subsequent transmissions between the same transmitter and receiver are full duplex transmissions. A scheme called shared random backoff is used to align the full duplex transmissions in time. In [8], the receiver of a primary transmission can send a packet to a secondary receiver. It also uses a busy tone signal to fill the time gap between the transmissions of the transmitter and the receiver to block neighboring transmissions.

Implementing and running SIM+ reveals the following unprecedented results. When only a pair of nodes are communicating, full duplex outperforms half duplex by $\sim 2\times$. This is as expected from previous work. When compared with 2×2 MIMO, full duplex has a comparable, yet slightly better, throughput owing to ACK timing alignments. When MIMO and full duplex operate in a large network, however, MIMO has a slight advantage over full duplex. This counter result comes from a fundamental trade off between these competing technologies. MIMO transmissions block the nodes only around one end node, while full duplex blocks nodes around both the end nodes. This gives an edge to MIMO. How-

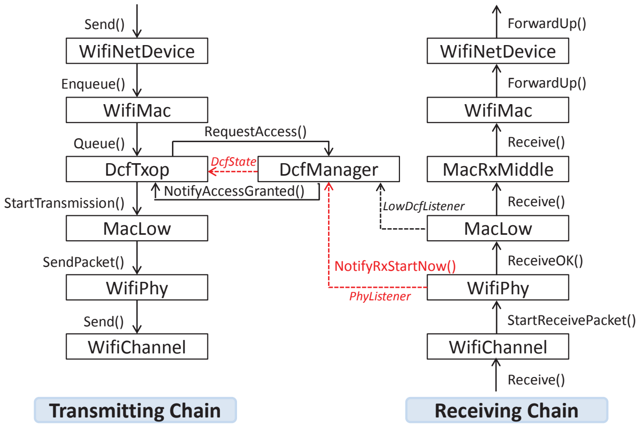


Figure 1: The transmitting and receiving chains in the "wifi" module of ns-3.

ever, MIMO transmissions are prone to hidden terminals, giving full duplex an edge. This interplay between blockage and hidden terminals is only visible when going to a larger network, as enabled by SIM+. We also present a case when all of the access points (APs) become full duplex and all the clients remain half duplex. In this case, when a full duplex AP, while receiving from a client, can transmit to a secondary client. Then it can significantly improve network performance.

Based on these results, we believe that SIM+ can help our community to understand the true potential of full duplex. Specifically, it can help us understand *when and how to best utilize full duplex.* We have made SIM+ available online [1] for the community to use.

2. DESIGN

This section introduces the Wi-Fi module in ns-3 and then discusses the implementation of SIM+. Finally, it presents the different full duplex MAC options provided by SIM+.

2.1 Wi-Fi implementation in ns-3

The "wifi" module in ns-3 implements the IEEE 802.11 standard. There are more than 50 classes to realize different Wi-Fi features. However, this section only introduces key classes related to the transmitting and receiving chains as shown in Figure 1.

Transmitting Chain: When the upper layer sends a new packet to the *WifiNetDevice*, it's forwarded to an *WifiMac* object, which is in charge of selecting the proper data rate and creating the MAC header. *WifiMac* is inherited by three different classes: *AdhocWifiMac*, *StaWifiMac*, *ApWifiMac*. The first one just implements the basic Wi-Fi mac functions, while the other two support beacon generation, probing and station association in AP based networks. The packet and the packet header created by *WifiMac* are then pushed to a packet queue in a *DcaTxop* object, which takes care of packet fragmentation and retransmission. The Distributed Coordination Function (DCF) is implemented in *DcfManager* and *DcfState*. The class *DcfManager* uses *DcfState* objects to store the contention window size and backoff counters. The method *DcfManager::RequestAccess()* is called by the *DcaTxop* object whenever the packet queue is not empty. When the backoff value counts down to 0, *DcfState* triggers *DcaTxop::NotifyAccessGranted()* to start accessing the channel. The packet is then pushed down to an *MacLow* object. The class *MacLow* takes care of RTS/CTS/DATA/ACK transmissions, while the *WifiPhy* class maintains the physical layer state and changes the state to *TX* when it starts sending a packet. The *WifiChannel* class estimates the received signal power between nodes and triggers the receiving chain at other nodes on the same channel after a propagation delay.

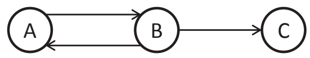
Receiving Chain: The received packet is forwarded up to *WifiPhy* from the *WifiChannel*. Depending on the current physical layer status and received signal power, the *WifiPhy* may change the physical layer state to *RX*. The packet correctness is checked at the end of reception, based on the interference level and data rate. A correctly received packet is sent to the *MacLow* object and a CTS or ACK packet may be sent based on the received packet type. Data packets are forwarded up to an *MacRxMiddle* object, which handles fragments and duplicates. Finally, the packet is forwarded up to the upper layer by *WifiMac* and *WifiNetDevice*.

2.2 Full duplex implementation in ns-3

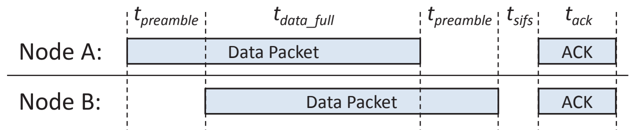
We choose to extend the "wifi" module, presented above, to build our full duplex implementation, SIM+. To enable full duplex operation, there are two key issues: 1) the physical layer should support simultaneous transmission and reception; 2) the MAC layer should take a full duplex action at the beginning of receiving a new packet.

In the "wifi" module, there is only one physical layer state machine, which changes between transmitting, receiving and idle listening, maintained using the class *WifiPhyStateHelper*. Although using one state to enable transmitting and receiving at the same time is realizable, it requires more states and transitions. For example, we need to add one state for concurrent transmitting and receiving. This state has both the properties of transmitting and receiving. It requires modifications in three different classes, *WifiPhy*, *WifiPhyStateHelper* and *WifiPhyListener*. These modifications are complex and not easy to maintain. In SIM+, we create two physical layer states in *WifiPhy*; one for receiving (*m_receiveState*) and the other for transmitting (*m_sendState*). These two states are maintained separately and we only need to modify the *WifiPhy* class.

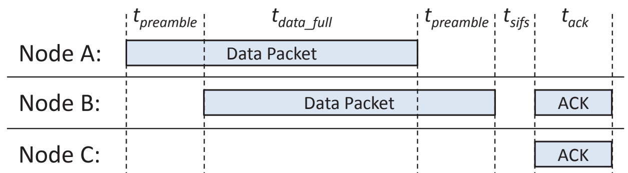
As discussed in the Section 2.1, the MAC layer packet



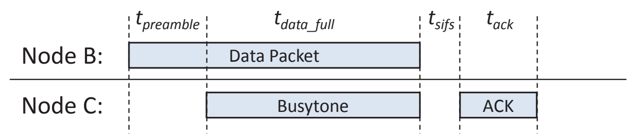
(a) Node topology. Node A and C could not hear each other, but both of them could communicate with node B. The arrows indicate traffic flow directions.



(b) Return transmission



(c) Secondary transmission



(d) Busytone transmission

Figure 2: The transmission time line of different MAC options for the full duplex receiver. $t_{preamble}$ is the time for transmitting preamble and PHY header, t_{data_full} is the time for full duplex to transmit the payload of a regular packet, t_{sifs} is the Short Interframe Space (SIFS) and t_{ack} is the time to transmit one ACK packet.

queue is maintained in *DcaTxop*. If the MAC layer decides to send a packet upon receiving a new packet, it has to dequeue a packet. This makes the class *DcaTxop* the best place to implement the reactions to currently receiving packet. In the "wifi" module, there is no direct connection between the object *WifiPhy*, where the packet receiving starts, and *DcaTxop*. However, there is a *PhyListener* object connecting the class *WifiPhy* and *DcfManager*, and a *DcfState* object connecting the class *DcfManager* and *DcaTxop*. The red lines in Figure 1 show the chain to forward the receiving packet from *WifiPhy* to *DcaTxop*. The method *DcaTxop::NotifyRxStartNow* is in charge of processing the incoming packet. Next subsection discusses different processing options.

2.3 Full duplex MAC options

The goal of SIM+ is to provide a general full duplex implementation that could be used in future protocol design. As shown in Figure 2, the receiver in a full duplex system could take different actions upon the arriving of a new packet. If a packet for the transmitter (node A) exists in the queue, the receiver (node B) could send it back to the transmitter (Figure 2(b)). Node B could also send a packet to another node (node C) if the reception at node C could tolerate the interference from node A (Figure 2(c)). To protect the reception from hidden terminals, node B could broadcast a busytone signal (Figure 2(d)). SIM+ enables these options in *DcaTxop* using attributes:

- *m_enableReturnPacket*: It enables sending a return packet if the receiver also has a packet for the transmitter in the queue.
- *m_enableForward*: It enables the receiver to send a packet to another destination while receiving. This should be used together with *m_forwardQueue* since the choices of destinations are stored there.
- *m_enableBusyTone*: It enables the sending of busytone to protect the incoming packet from hidden terminals. This option has the lowest priority. If all of the above three options are enabled, the receiver will try to find a return or secondary transmission packet first. If none of them exists, a busytone signal will be sent.
- *m_forwardQueue*: It stores the potential secondary transmission destination addresses for each primary transmitter. Each address is associated with a priority number and currently the address with the highest priority is selected as the forwarding destination.
- *m_enableCaptureEffect*: Current "wifi" module does not allow the wireless devices to relock onto a new packet with stronger signal power. However, capture effect [9] has been proposed to allow receiving a stronger transmission during the reception of weaker transmissions. This attribute, unlike the others, is maintained in the class *WifiPhy*. It improves the performance when *m_enableForward* is set to true because it allows the secondary receiver to re-capture the stronger signal from the receiver.

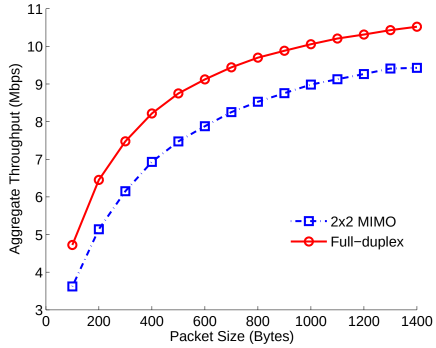
In case both *m_enableReturnPacket* and *m_enableForward* are enabled, the transmitter address should also be added into the *m_forwardQueue* with a priority value. This allows us to control the priority between sending a return packet and a secondary packet.

3. EVALUATION

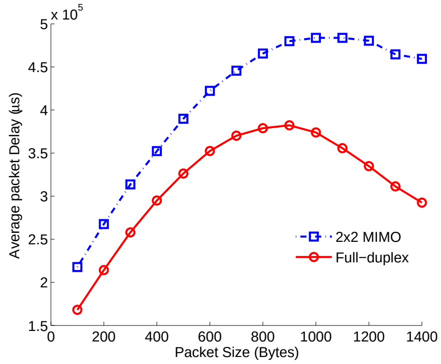
This section evaluates the performance of SIM+. First, it compares full duplex and MIMO in a 2-node scenario. Then, it explores how well full duplex could deal with hidden terminals. Finally, it studies a network with full duplex capable AP and half-duplex clients.

3.1 Full duplex v.s 2×2 MIMO

Since the full duplex systems in [5, 6] use two RF chains, the same as 2×2 MIMO, it naturally brings up a question: which one is better, full duplex or 2×2



(a) Aggregate throughput



(b) Average packet delay

Figure 3: Throughput and average delay of a 2-node network (Node A and B as shown in Figure 2(a)). The data traffic of both links $A \rightarrow B$ and $B \rightarrow A$ are UDP flows generated at 6 Mbps. The physical layer data rate is 6 Mbps for full duplex and 12 Mbps for MIMO.

MIMO? To answer this question, we create a 2-node scenario and study the throughput and delay in two cases.

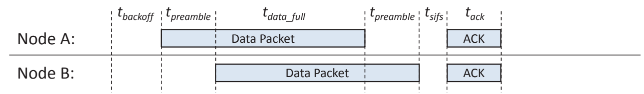
First, we set up a simple scenario with 2 nodes (A and B as shown in Figure 2(a)) sending to each other with saturated UDP traffic. The physical layer data rate is set to 6 Mbps for full duplex and 12 Mbps for 2×2 MIMO. Figure 3 shows the throughput and delay performance. Full duplex achieves 10.8% to 30.4% higher throughput than 2×2 MIMO and reduces the delay by 16.3% to 36.3%. To illustrate why full duplex has better performance, we draw the transmission time lines for full duplex and MIMO in Figure 4. The time for full duplex to transmit one packet from each node is:

$$t_{full} = t_{backoff} + 2 * t_{preamble} + t_{data\_full} + t_{sifs} + t_{ack}$$

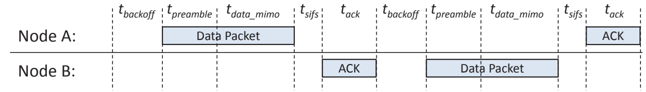
And the time for 2×2 MIMO is:

$$t_{mimo} = 2 * (t_{backoff} + t_{preamble} + t_{data\_mimo} + t_{sifs} + t_{ack})$$

Given that the data rate of MIMO is twice of full duplex, we have $t_{data\_full} = 2 * t_{data\_mimo}$. Then the time



(a) Full duplex transmission time line



(b) 2×2 MIMO transmission time line

Figure 4: The time lines of transmitting two packets in a 2-node network (Node A and B as shown in Figure 2(a)). $t_{backoff}$ is the average backoff time (72μs in 802.11g), $t_{preamble}$ is the time for transmitting preamble and PHY header, t_{data_full} and t_{data_mimo} are the time for full duplex and MIMO to transmit the payload of a regular packet, t_{sifs} is SIFS duration and t_{ack} is the time to transmit one ACK packet.

difference of transmitting two packets is:

$$t_{mimo} - t_{full} = t_{backoff} + t_{sifs} + t_{ack}$$

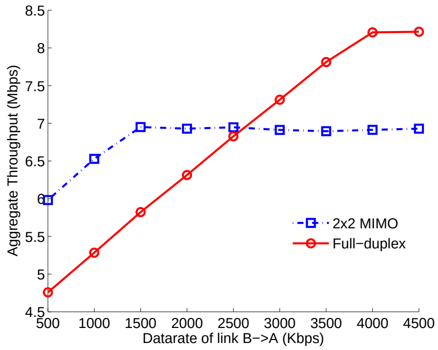
This equation shows that full duplex consumes less transmission time than MIMO, which leads to higher throughput. Note that this time difference is fixed regardless of the packet size and packet duration. So the advantages of full duplex increases when the packet size decreases.

In the experiment discussed above, both of the links are saturated, which implicitly assumes that a return packet is available all the time. In reality, this assumption may not always hold. To study the influence of traffic asymmetry, we then fix the data generation rate of link $A \rightarrow B$ to 6 Mbps, and vary the data rate of link $B \rightarrow A$. The packet size is set to 400 Bytes. The results are shown in Figure 5. Because 2×2 MIMO has higher physical data rate, it achieves higher throughput and lower delay when the data rate of link $B \rightarrow A$ is low. It then reaches the capacity limit when the data rate of link $B \rightarrow A$ is around 1500 Kbps. Full duplex, on the other hand, has higher capacity limit and achieves higher throughput when the data rate is above 2500 Kbps.

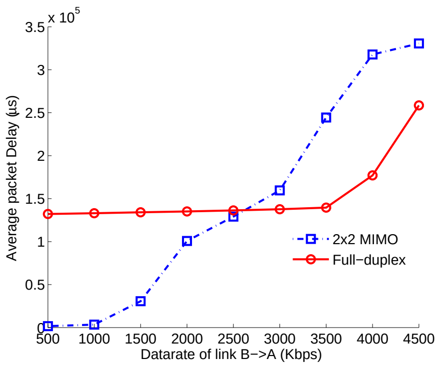
Lessons learned: These experiments show that full duplex outperforms 2×2 MIMO when the traffic is symmetric. In case of asymmetric traffic, full duplex could still gain higher throughput when there is adequate return traffic.

3.2 Hidden terminals

Hidden terminals have been an outstanding problem in Wi-Fi communication. The problem roots from the channel asymmetry at the transmitter and receiver. In full duplex system, since both the transmitter and receiver are transmitting simultaneously, the hidden ter-

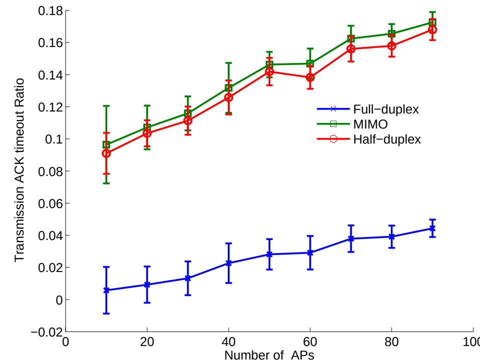


(a) Aggregate throughput

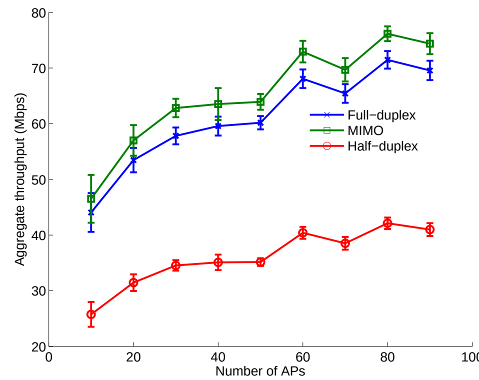


(b) Average packet delay

Figure 5: Throughput and average delay of a 2-node network (Node A and B as shown in Figure 2(a)). The data traffic of link $A \rightarrow B$ is UDP flow generated at 6 Mbps and the data traffic of link $B \rightarrow A$ varies.



(a) Transmission ACK timeout ratio of different schemes



(b) Aggregate throughput of different schemes

Figure 6: Transmission ACK timeout ratio and aggregate throughput of different schemes with varied number of APs.

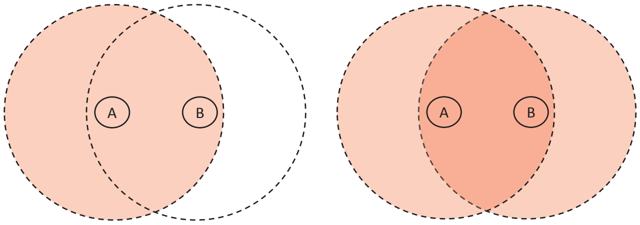
minal problem is supposed to be removed. However, since the primary receiver can not start transmitting before it decodes the MAC header of the incoming packet, there is still a chance for hidden terminal to transmit. To study the performance of full duplex against hidden terminals, we create a network with the following setup. We divide an $800 \times 800 m^2$ area into a number of equal size blocks. Then an AP is placed randomly around the center point of each block. Finally, several clients are placed around each AP. Both the uplink and downlink traffics are saturated UDP flows.

We compare the performance of full duplex with 2×2 MIMO and half-duplex. Since hidden terminal problem causes packet collisions and naturally leads to ACK timeouts at the transmitter, we use the transmission ACK timeout ratio, which is the total number of transmissions divided by the total number of ACK timeouts, and aggregate throughput as the metrics to evaluate these three schemes. Each experiment is conducted 20 times and the mean value is plotted with the standard variation as the vertical bar around it. As shown in Figure 6(a), compared with MIMO and half-duplex, full duplex experiences significantly smaller ratio of ACK timeouts. This result indicates that full duplex relieves hidden terminal problem. In Figure 6(b), the throughput of full duplex is less than that of MIMO, which contradicts with the results shown in Figure 3(a). This is because full duplex has a larger interference range for each pair of transmissions. As shown in Figure 7, only the nodes around node A can hear the transmission and are not allowed to transmit in MIMO. In full duplex, however, the nodes around both the transmitter A and receiver B will sense the channel to be busy. Given the same network area, MIMO could support more concurrent transmission pairs compared with full duplex.

Lessons learned: Although full duplex outperforms MIMO when there is only one pair of nodes, its advantage vanishes when there are more nodes in the network because of larger interference range. Note, however, that a larger interference range also increases the number of exposed terminals. A future full duplex protocol could use this fact to enable concurrent transmissions from such exposed nodes. This way full duplex can potentially improve network throughput over MIMO.

3.3 Full duplex capable AP systems

There are many wireless devices on the market to-



(a) Interference range of 2×2 MIMO. All nodes around node A are blocked from transmission.

(b) Interference range of full duplex. All nodes around node A and B are blocked from transmission.

Figure 7: Interference area of full duplex v.s. 2×2 MIMO

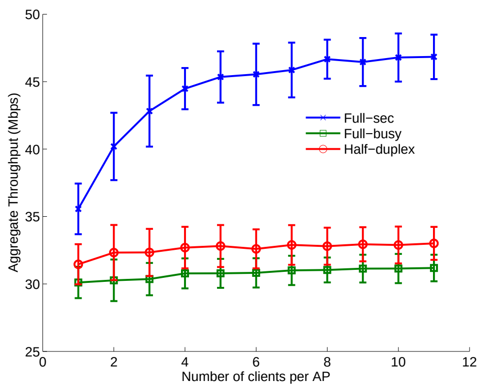


Figure 8: Aggregate throughput in a network with 20 APs. The APs are full duplex capable while the clients are half-duplex nodes.

day: such as smartphones, tablets and laptops. It is difficult to make all of these existing devices full duplex capable. However, all of these devices are connected to APs, which are static and could be maintained easier. Full duplex capable APs can send packets to another client while receiving. In this section, we study the network performance with full duplex AP and half duplex clients. We assume that all of the APs are equipped with two antennas and are full duplex capable while the clients only have one antenna and are working in half-duplex. All of the uplinks and downlinks are saturated with UDP flows. The result is shown in Figure 8. We compare the throughput of full duplex with secondary transmission only (denoted as Full-sec), full duplex with busytone only (denoted as Full-busy) and half-duplex. As the number of clients per AP increases, the opportunities for secondary transmissions also increases and so the aggregate throughput of Full-sec. The throughput of half-duplex, on the other hand, is limited by the total number of APs. Full-sec achieves 46% higher throughput than half-duplex when the number of clients per AP is 11. Full-busy keeps performing the worst because of the larger interference range compared with half-duplex.

Lessons learned: Full duplex AP is a promising direction to increase the network throughput. Busytone signal increases the interference range and it may not be the best option to turn on busytone signal all of the time. A history based scheme for turning on busytone signal is a potential future work direction.

4. CONCLUSION AND FUTURE WORK

In this paper, we present SIM+, the first full-duplex simulator with ns-3. SIM+ implements different MAC options to provide general interfaces for future development of full-duplex MAC protocols. We also study the performance of full-duplex in several scenarios. Compared with 2×2 MIMO, full-duplex performances better in a 2-node scenario while worse in a more general network. Although full-duplex could be used to deal with hidden terminal problem, it comes with the cost of fewer concurrent transmissions. Our simulation results also suggest that deploying full-duplex capable APs and allowing secondary transmissions increases network throughput. We have made SIM+ available for the community. We believe that it enables our community to understand the true potential of full duplex and to develop future network architectures based on full duplex.

Our future work focuses on two parts. First, adding self-interference simulation to the physical layer when the node is transmitting and receiving simultaneously. The current implementation of SIM+ assumes that the self-interference is canceled perfectly, which does not hold [3, 5, 6]. Second, developing effective full-duplex MAC protocols based on SIM+.

References

[1] Technique report for SIM+, July 2013. https://sites.google.com/site/techrepanon789/.

[2] E. Aryafar, M. A. Khojastepour, K. Sundaresan, S. Rangarajan, and M. Chiang. MIDU: enabling MIMO full duplex. In *Proc. of ACM MobiCom*, pages 257–268, 2012.

[3] J. I. Choi, M. Jain, K. Srinivasan, P. Levis, and S. Katti. Achieving Single Channel, Full Duplex Wireless Communication. In *Proc. of ACM MobiCom*, pages 1–12, 2010.

[4] E. M. Dinesh Bharadia and S. Katti. Full Duplex Radios. In *Proc. of ACM SIGCOMM*, 2013.

[5] M. Duarte and A. Sabharwal. Full-Duplex Wireless Communications Using Off-The-Shelf Radios: Feasibility and First Results. In *the Forty Fourth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, pages 1558 – 1562, nov. 2010.

[6] M. Jain, J. I. Choi, T. Kim, D. Bharadia, S. Seth, K. Srinivasan, P. Levis, S. Katti, and P. Sinha. Practical, Real-time, Full Duplex Wireless. In *Proc. of ACM MobiCom*, pages 301–312, 2011.

[7] A. Sahai, G. Patel, and A. Sabharwal. Pushing the limits of Full-duplex: Design and Real-time Implementation. *Technical Report TREE1104, Rice University*, 2011.

[8] N. Singh, D. Gunawardena, A. Proutiere, B. Radunovic, H. V. Balan, and P. B. Key. Efficient and Fair MAC for Wireless Networks with Self-interference Cancellation. In *International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, pages 94–101, 2011.

[9] K. Whitehouse, A. Woo, F. Jiang, J. Polastre, and D. Culler. Exploiting the Capture Effect for Collision Detection and Recovery. In *EEE EmNets Workshop*, pages 45–52, 2005.