Efficient and Fair MAC for Wireless Networks with Self-interference Cancellation

Nikhil Singh[†1], Dinan Gunawardena[*], Alexandre Proutiere[*], Božidar Radunović[*], Horia Vlad Balan[‡1], Peter Key[*]
[*]Microsoft Research, UK
[†]University of Illinois at Urbana Champaign
[‡]University of Southern California

***Abstract*—Recent advances in PHY layer design demonstrated efficient self-interference cancellation and full-duplex in a single band. Building a MAC that exploits self-interference cancellation is a challenging task. Links can be scheduled concurrently, but only if they either (i) don't interfere or (ii) allow for self-interference cancellation. Two issues arise: Firstly, it is difficult to construct a schedule that fully exploits the potentials for self-interference cancellation for arbitrary traffic patterns. Secondly, designing an efficient and fair distributed MAC is a daunting task; the issues become even more pronounced when scheduling under the constraints. We propose ContraFlow, a novel MAC that exploits the benefits of self-interference cancellation and increases spatial reuse. We use full-duplex to eliminate hidden terminals, and we rectify decentralized coordination inefficiencies among nodes, thereby improving fairness. Using measurements and simulations we illustrate the performance gains achieved when ContraFlow is used and we obtain both a throughput increase over current systems, as well as a significant improvement in fairness.**

I. Introduction

Self-interference cancellation [1] has been proposed as a means of increasing network throughput and achieving full-duplex communication using a *single* band in low-power wireless networks, such as Zigbee, with 0dBm TX power. In a follow-up work, Choi et al [2] have further improved the performance of the self-interference cancellation to fully remove the noise, again in low-power setting.

On a single wireless link, with traffic in both directions, these techniques have a potential to double the throughput. However, it is unclear how to extend the techniques to an arbitrary network. The first challenge is how to exploit self-interference cancellation when traffic is not symmetric - a full-duplex link does not suffice. We need to build a MAC that will be able to create more complex scheduling patterns, adapted to the traffic.

The second challenge is how to build a distributed MAC to schedule these patterns in a fair and efficient way. This is a difficult problem even in half-duplex networks. Current deployed wireless MACs, such as the Distributed Coordination Function (DCF) of 802.11, have numerous efficiency and fairness issues caused by hidden node and exposed terminal problems. In a network with self-interference cancellation, concurrent links have to be scheduled such that they either do not interfere or allow self-interference cancellation. This further exacerbate the problem of the efficient scheduling. All these challenges are discussed in details in Section II.

In this paper, we introduce *ContraFlow*, a novel medium-access control (MAC) protocol for low-power wireless networks that uses a single channel and exploits self-interference cancellation. ContraFlow is a distributed MAC that is able to exploit self-interference cancellation whenever an opportunity to do so exists. It uses self-interference cancellation to mitigate hidden terminal problem and adaptive scheduling to maintain fairness. To our knowledge, ContraFlow is the first MAC for full-duplex, single-channel wireless networks.

ContraFlow consists of two parts:

- **Dual links access control:** We introduce the concept of a *dual-link* to exploit spatial reuse and solve the contention resolution problem of full-duplex networks (Section II). We define a novel medium access and ARQ (Automatic Repeat-reQuest) procedure that schedules both symmetric and asymmetric dual-links with very low collision probability (explained in Section V).
- **Distributed scheduler:** We show that combining full-duplex and standard DCF creates significant efficiency and fairness problems. As a solution, we propose a novel distributed scheduler that provides efficient and fair scheduling of dual links (explained in Section V).

At the MAC layers, many solutions have been tested to eliminate hidden and exposed terminals, see e.g. [3] and references therein. One of the first is FAMA protocol [4]. The use of busy tone has been proposed in [5] to combat hidden terminals, but this requires a second signaling channel (which we don't need here). Another interesting solution is to map other interfering node and exploit the maps at the MAC layer [6]. We stand out compared to the previous solutions as we use full duplex combat hidden and exposed terminals but also to increase the efficiency of the network. Also, use of interference cancellation to improve efficiency in a WiFi setting has been recently evaluated in [7], but no MAC was proposed.

We evaluate the Contraflow MAC protocol on several topologies. We first measure the performance of the interference cancellation on a single link in hardware (Section III). We then use the results of these measurements in a simulation to evaluate the performance of the proposed MAC (Section VI).

[1] The work was done while the authors interned with Microsoft Research.

In our experiments, ContraFlows gains both in terms of system throughput and fairness.

II. Motivation and Concepts

We start by defining a *dual-link*, and then describe contention resolution, efficiency and fairness problems, which we address in the rest of the paper.

A. Dual-link

The basic idea of self-interference cancellation is as follows: when a node, say A, senses the channel idle and when its back-off counter reaches zero, it starts transmitting a packet to a node, say B. As soon as B is able to decode the PHY/MAC header of the packet (indicating who is transmitting), it immediately starts transmitting either a packet or a busy tone, depending on whether or not B has packets to send to one of its neighbors (including A). In all cases, B transmits while A transmits. B performs self-interference cancellation to decode the packet sent by A. Similarly A uses self-interference cancellation either to know that B sends a busy tone or to decode the packet sent by B. If A does not detect any transmission from B, it becomes aware that its transmission to B was not successful.

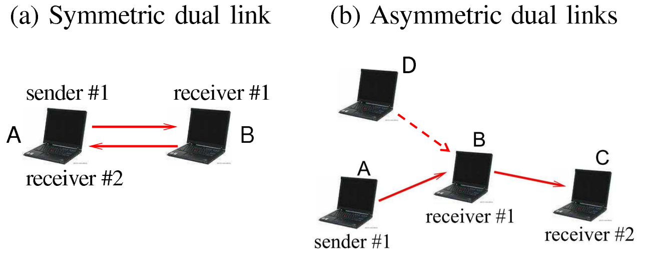


Fig. 1. Enabling dual links: using self-interference cancellation at nodes A and B, we can activate two links simultaneously, i.e., dual links - (a) a symmetric dual link (A, B, A), (b) two potential assymetric dual links: (A, B, C) and (D, B, C).

Since B is able to send a packet while receiving, successful simultaneous transmission on two links that would interfere without interference cancellation is made possible. We refer to such pairs of links as *dual-links*. Dual-link is an ordered tuple (A, B, C), where node B receives from node A and transmits to node C at the same time. Examples of symmetric and asymmetric dual-links are shown in Figure 1. Using dual-links, we increase the spatial reuse by enabling more transmissions over the same area. With dual links and optimal scheduling, we alway in increase network throughput, and can double the feasible throughput in the best case.

The use of dual-link also eliminates hidden terminals. Since the receiver B sends a signal while receiving, it prevents any other interfering node in its neighborhood from starting transmitting, ensuring that the packet sent by A is received successfully. This can be seen as a perfect RTS/CTS procedure with no overhead. Hence ContraFlow provides an efficient way of canceling the impact of hidden terminals, see Figure 2.

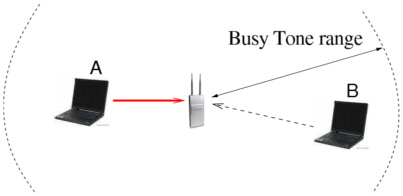


Fig. 2. Handling hidden terminals: The transmission from A to the access point is protected from the hidden node B by the packet/busy tone sent by the access point while receiving the packet from A.

B. Challenges with Dual-links

The performance improvement of using full-duplex scheduling depends on the actual traffic patterns in the network. It is extremely important to be able to exploit both symmetric and asymmetric dual links to achieve the maximum performance improvement. A symmetric dual link (Figure 1 (a)) will increase throughput only if there is traffic from both A to B and B to A. An asymmetric dual link provides many more scheduling opportunities. In the example from Figure 1 (b), it would allow us to schedule either A-B, D-B or C-B in parallel with B-C.

Contention resolution: The main challenge with asymmetric dual links is how to resolve possible contention. Consider again Figure 1(b) and suppose node B starts transmitting to node C. Once node A detects this transmission, it can then potentially start transmitting to node B in parallel. However, since B is already transmitting, existing carrier-sense collision avoidance does not work any more, and node A has no way of knowing if node D has also tried to transmit to B. We solve this problem by forming dual links (both symmetric and asymmetric) in a collision-free pattern, as described in Section IV.

Dual-link access could be implemented using the existing 802.11 DCF, but the inherent fairness issues arising in networks using CSMA would then remain. In fact these issues are even more pronounced when scheduling complex patterns as asymmetric dual links (quantified in Section VI).

Efficiency: A classical CSMA/CA algorithm, applied on top of self-interference cancellation, will give equal access priority to all nodes, hence there is a substantial probability that a non-efficient dual link will be scheduled and the spatial reuse will be low. Consider again the example from Figure 1 (b). Suppose that both nodes A and D have a packet to send to B, and suppose B has a packet for C. If node B acquires a channel, node A and D will not be able to transmit because they cannot resolve the contention (or otherwise they will collide), hence the efficiency drops. On the other hand, if node A acquires a channel, then node D will refrain from transmitting due to the carrier sensed, and node B will successfully transmit concurrently with A.

Fairness: The *fairness problem* is a classical problem in 802.11 or 802.15-based networks, caused by the carrier sensing mechanisms and the basic principle of the DCF. The problem is best illustrated in a 3-link network, without hidden terminals (see Figure 3). Link 1 (A, B) and 3 (E, F) do not

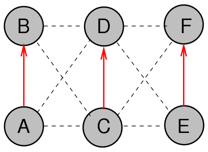


Fig. 3. A 3-link network with fairness issues. A dashed line between two nodes mean that they sense each other.

interfere with each other, whereas they interfere with and are interfered by link 2 (C, D). When link 1 (resp. link 3) is active, link 2 senses the medium busy and does not attempt to use it, whereas link 3 (resp. link 1) may start transmitting. As a result, link 2 may see the medium busy for a very long period, corresponding to a succession of transmissions on links 1 and 3. Link 2 finally observes an idle medium when all links are inactive. But now because link 2 competes with more links than links 1 and 3, it will experience a higher collision probability, and hence because of the DCF, will have a smaller transmission probability than the other links. Again, due to complex scheduling patterns, this problem is even more pronounced with dual-links.

III. Self-interference cancellation in practice

Self-interference cancellation for low-power networks has recently been proposed and evaluated in [1] and further improved in [2]. Our goal in this section is to evaluate the performance of the interference cancellation in practice and use these values to design and evaluate a MAC for full-duplex wireless networks.

These scheme from [2] is attractive because it promises to cancel the entire self-interference. However, it turns out that in an indoor environment it is very sensitive to the actual antenna location (e.g. does not work well when antennas are close to a wall). Instead, we evaluate the performance of the more robust scheme from [1] to obtain a more realistic data rates. We note that in the scenarios where [2] fully works, the performance of Contra-Flow will be even better.

In order to get realistic physical layer performance evaluation, we perform measurements on our software-defined radio test-bed which implements full-duplex communication using analog interference cancellation [8] (for a full description of the test-bed please see [1]). We give the link loss measurements from experiments using interference cancellation scheme and dual links. Both symmetric $((A, B, A))$ and asymmetric $((A, B, C))$ dual link scenarios were used, as depicted in Figure 1, and labeled ABA and ABC. The results are given in Figure 4. Here R_1 and R_2 referred to received packet success rates (goodputs) at A and B. The results can be compared to the success rates obtained on the isolated links (A, B) and $(B, A$ or $C)$ presented by the points on the x-axis and y-axis, respectively. We also illustrate the performance of dual links obtained when using perfect time-sharing (scheduling) between links (A, B) and $(B, A$ or $C)$. This performance is illustrated by the diagonal dashed line.

First note that the clustered points lie above the straight line joining the performance when just isolated nodes send.

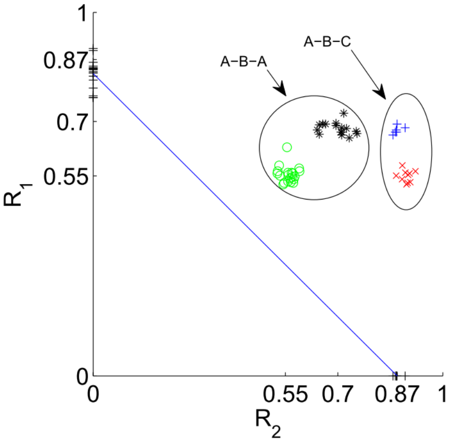


Fig. 4. (a) An example of successful transmissions on an asymmetric dual-link. (b) A collision can easily be detected by node A if the secondary transmission is absent. (c) Average performance of dual links in presence of data transmissions for two different positions of nodes

In other words, although we loose more packets when using self-interference cancellation, a consequence of not being able to perfectly cancel interference, the combined throughput is greater than the optimal possible without interference cancellation (when transmits and receives are scheduled separately). Note also that in the asymmetric scenario, link (B, C) has similar performance to the same link in isolation (since C does not transmit any signal).

IV. Handling dual links

We now detail the MAC-layer mechanisms involved in ContraFlow. The key ingredients are (a) deliberately limiting the choice of secondary senders to primary receivers to avoid collisions, (b) the use of busy tones when data packet is short, or not available, to avoid hidden terminal problems for secondary senders, (c) use a weighted list to select secondary receivers to achieve fairness. We first explain how we avoid collisions, and define our terminology.

A. Avoiding Collisions

We start by describing how ContraFlow initiates and handles dual-links. In the following we refer to the node initiating a link or a dual link as the *primary sender*, and to the node receiving the packet sent by the primary sender as a *primary receiver*. Due to self-interference cancellation, another transmission can potentially occur in parallel with the primary transmission. We call it a *secondary transmission* and the nodes involved in it the *secondary sender* and the *secondary receiver*.

To address the contention problem, described in Section II, we mandate that *only the primary receiver is allowed to initiate the secondary transmission* and become the secondary sender. Since the primary receiver makes a decision on the secondary transmission internally, there is no need for contention resolution.

For example, only node A can initiate the dual link (A, B, C) from Figure 1(b). Node A is then the primary sender, node B is the primary receiver and the secondary sender, and node C is the secondary receiver. A symmetric dual link (such as dual-link (A, B) from Figure 1(a)) can be initiated by either of the nodes.

The proposed scheme may seem overly restrictive. For example, if node B from Figure 1(b) starts a transmission, the only dual link that can be formed is the symmetric dual link (B, C). If C has no packet to send to B, we may miss an opportunity to schedule nodes A or D in parallel with B. However, as we will see in Section V, this problem is handled by the scheduling algorithm. In what follows we describe the dual link access control in detail.

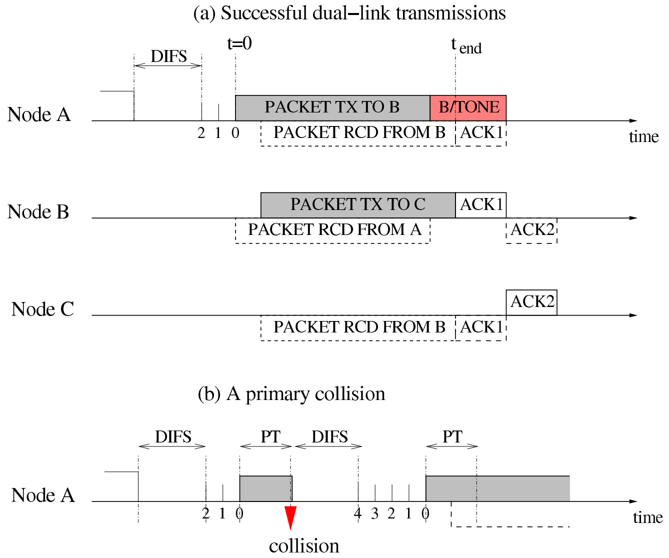
B. Packet and ACK transmissions on dual-links

Here is a chronological description of how nodes initiate and handle transmissions. Figure 5 illustrates these basic mechanisms. Note that each node implements a carrier sensing mechanism (CSMA/CA) as specified in the 802.11 or 802.15 standards. At the end of a period where the channel has been busy, a node will start decrementing its back-off counter after sensing the channel idle for a period of duration `DIFS`. Then the back-off counter is decremented in each time slot, should it be sensed idle.

Primary transmission. Assume that at time 0, node A has its back-off counter equal to zero. A then becomes a primary sender, and it starts transmitting to node B a packet whose header indicates that it is a primary transmission. While starting the transmission, A also starts a *primary* timer expiring at `PT`.

Secondary transmission. As soon as B is able to decode the MAC header of the packet sent by A, it becomes the primary receiver and immediately decides to transmit a packet to a secondary receiver, say C, chosen from a list $\mathcal{S}_A$ of nodes (note that C may be different than A). The way nodes create and maintain these lists is described in Section IV-C. If no packet is available, B may decide to send a busy tone, if needed to protect from a hidden terminal. Next,
(i) If at time `PT`, A could not sense a signal sent by B, it immediately stops transmitting and declares a *primary collision*. A then updates its back-off algorithm parameter accordingly (see the next section).
(ii) Otherwise, A proceeds with the transmission.

MAC acknowledgments. Let t_A (resp. t_B) be the time at which the transmission of the packet sent by A (resp. B) ends. The size of the packet sent by B is always chosen such that its transmission ends before $t_A + \epsilon$, where ϵ is the small off-set representing the time difference between the epochs at which A and B start transmitting. We justify this choice below. At time $t_{end} = \max(t_A, t_B)$, B acknowledges the packet sent by A. This ack transmission lasts for `ACK1`. During the time interval $[t_B, t_{end}]$, node B sends a busy tone, while during $[t_A, t_{end}+$`ACK1`$]$, A also makes the medium busy by sending a busy tone. Node C acknowledges the packet sent by B as soon as it senses the channel idle, i.e., at time $t_{end} +$ `ACK1`. Just as in 802.11 standards, if a transmitter does not receive the MAC ack before expiration of a timer, it declares a collision, and updates the parameters of its back-off algorithm accordingly. A transmission failure from B to C is referred to as a *secondary collision*.


(a) Successful dual–link transmissions

(b) A primary collision

Fig. 5. (a) An example of successful transmissions on an asymmetric dual-link. (b) A primary collision experienced by node A.

By the use of busy tones, we ensure the secondary transmission ends no later than the primary transmission. The rationale for this is that the secondary transmission is not as well protected from hidden terminals as the primary transmission (since in general the secondary receiver does not send a signal while receiving). The packet sent by the primary receiver could be much smaller than the packet it is receiving, e.g., it could be a TCP acknowledgment. We illustrate this scenario in Figure 6.

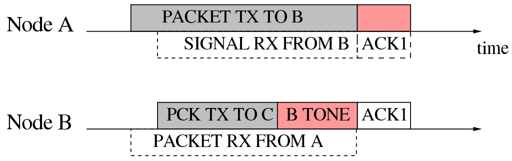


Fig. 6. A short packet is protected with a busy tone

It is important that the primary sender and receiver transmit a busy tone during the first MAC ack and before the end of the primary transmission, respectively, so as to occupy the medium to protect the transmissions of both MAC acks. Otherwise, the medium could be sensed idle for a duration greater than `DIFS` by other interfering nodes that could then start transmitting.

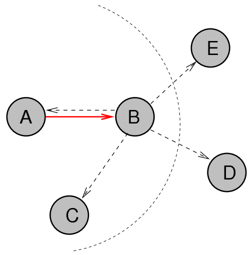
Implementation issues: One of the main concerns when implementing the proposed scheme is the processing delay in PHY/MAC. A primary receiver needs to decode the header from the primary transmission, choose a destination/packet for the secondary transmission accordingly, fetch it and start transmitting, and all this with very low overhead in order to fully exploit the potential for concurrent transmission. To address this problem, we have fully implemented this functionality in FPGA in our test-bed. Once a packet header is received by a primary receiver, it is able to transmit the secondary packet in less than 1μs, a delay which is acceptable even for the most advanced PHY layers.

C. Selection of the secondary receiver

When a primary receiver selects a secondary receiver from its neighbors, it has to account for the fact that the primary sender could interfere the reception at these nodes. The objective for a primary receiver is to do this selection so as to minimize the secondary collision rate.

Secondary collisions can be caused by either (1) the interference structure of the network, i.e., the primary sender interferes at the secondary receiver; or (2) the level of congestion of the network, i.e., a node in the neighborhood of the secondary receiver not sensing the activity of the primary sender and receiver, starts transmitting during the secondary transmissions (the hidden terminal problem for the secondary transmission). Ideally we would like to distinguish these two kinds of events.

To limit secondary collisions, when a primary receiver receives a packet from a primary sender, say A, it chooses the secondary receiver in a weighted list $\mathcal{S}_A$, where the weight of each possible secondary receiver, say C, represents the proportion of successfully secondary transmissions in the past using dual-link (A, B, C). In practice, the weight is computed on the basis of the x (=10) previous such transmissions. This weight is used to choose the secondary receiver as specified in the MAC scheduling algorithm, described in the next section (a node with a higher weight is more likely to be selected). In the simulations presented later, we observed that this simple way of building the weighted lists was sufficient to efficiently discover the interference structure of the network. In the case where the network topology is fixed and where fading is not highly varying, the weighted lists do not evolve in time. In other cases, these lists adapt to the topology and fading changes. We present an example of such list in Figure 7.



$\mathcal{S}_A$	weight
A	1.00
C	0.00
D	0.90
E	0.80

Fig. 7. An example a weighted list of secondary receivers at node B.

D. Spatial reuse and the exposed terminals

We conclude this section with a remark on spatial reuse. Our MAC-layer mechanisms completely eliminate the hidden terminal problem for the primary transmission. In the case of symmetric dual-links, we eliminate the hidden terminal problem for the secondary transmission. But we also mitigate the impact of the exposed terminal problem. Consider for example the system of Figure 8. The carrier sensing mechanism forbids both links (A, B) and (C, D) from being active simultaneously, hence reducing spatial reuse. Now for example, if the dual link (A, B, A) is activated, two links in the vicinity of node A are active simultaneously, which brings the spatial reuse around node A at the same level as that we would obtain activating link (C, D) (without the exposed terminal problem).

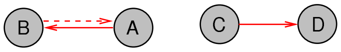


Fig. 8. Dual links mitigate the impact of exposed terminals.

V. Fair and efficient distributed scheduler

In this section we describe a distributed scheduling algorithm that tackles the efficiency and fairness problems discussed in Section II-B.

A. Distributed Scheduling Algorithm

In designing the distributed scheduling algorithm we want to keep the basic principle of CSMA and avoid using any additional signalling and message passing. The goal of the MAC is to achieve proportional fairness [9], a well accepted trade-off between efficiency and fairness. The main idea is to let nodes tune their access probability as a function of the past proportions of time their own out-going links have been active. It can be seen as an extension of those proposed in [10] to the dual-link model.

However for such protocols to work well, it is crucial to first eliminate hidden terminals: if two nodes are hidden from each other, their transmissions would collide and fail, and consequently they would increase their transmission probabilities which in turn exacerbates the hidden terminal problem there. Hence, algorithms can be implemented thanks to the fact that ContraFlow eliminates hidden terminals.

Our protocol has two components: the access scheme that defines how nodes attempt to use the channel, and the dual-link component that specifies how dual links are formed. As in most of the existing MACs (WiFi, Zigbee), we assume slotted time. For each node n, there is a set of out-going links $\mathcal{O}_n$. For each of these links, node n maintains a pressure indicator $\mathtt{p}_l$, and updates it at the beginning of each slot according to:

$$\mathtt{p}_l[t+1] = \mathtt{p}_l[t] + \epsilon \times \left(I(\mathtt{p}_l[t]) - D(\mathtt{p}_l[t], S_l[t])\right), \qquad (1)$$

where $S_l[t]$ represents the service received on link l during slot t, and where $I(\cdot)$ and $D(\cdot, \cdot)$ are positive functions. ϵ is a small parameter. I and D are such that the value of $\mathtt{p}_l$ is upper bounded by $p_{\max}$. Node n runs a back-off algorithm whose contention window at slot t is a random variable uniformly taken from the interval $[0, \mathtt{CW}_n[t]-1]$ where $\mathtt{CW}_n[t] = 1/P_n[t]$. $P_n[t]$ (upper bounded by P_{max}), is the access probability of node n related to the pressure indicators of all out-going links from n:

$$\mathtt{P}_n[t] = \max_{l \in \mathcal{O}_n} \mathtt{p}_l[t]/L_l[t], \qquad (2)$$

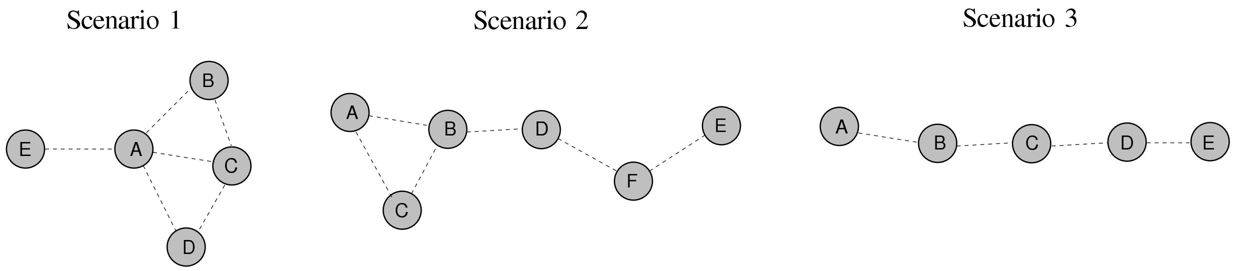


Fig. 9. Illustrative network scenarios.

where $L_l[t]$ would be the duration (in slots) of the transmission of the packet at the Head-of-Line (HoL) in the buffer of link l.

Note that $S_l[t]$ is related to the evolution of the transmission probabilities $\mathtt{P}_n$, and hence (1)-(2) may be interpreted as the equations of the stochastic approximation algorithm. This algorithm is not classic as the updates depend on stochastic processes $S_l[t]$ whose evolution are driven by the parameters themselves. Here we choose:

$$I(x) = \frac{xV}{\log x}, \quad D(x,s) = xs.$$

V is a parameter (equal to 1 in our experiments). Note that the algorithm clearly increases (resp. decreases) the transmission probability of links that are not served (resp. that are served), which will improve fairness. Finally, to choose the second link composing the dual-links, we pick the node maximizing $\mathtt{weight}_C \times \mathtt{p}_m[t]$.

We cannot prove that the proposed protocol with dual links converges to the proportionally fair rate allocation (as it was originally done for single links in [10], [11]). In fact, this is probably not the case: in order to guarantee that the optimal dual link is scheduled with high probability, we would need to add additional signalling between links comprising the dual link. We deliberately sacrifice optimality for simplicity. However, as shown experimentally in the next section, the algorithm indeed converges and significantly improves both efficiency and fairness fairness.

B. Handling collisions

In the fair MAC protocols we just presented, when transmitters do not access the channel, their pressure indicator increases, which in turn increase their transmission probability. In particular, nodes increase their transmission probability upon collisions, which can be problematic. That is why it is important to carefully choose the algorithm parameters.

The crucial parameter to control the collision probability is the minimum value of the contention window, or equivalently the maximum value $p_{\max}$ of the pressure indicator of links. Choosing a high value would limit the collision rate, but at the expense of efficiency. Ideally, to get negligible collision rates, we could choose a very large minimum contention window, but to compensate the efficiency loss, a very large channel holding time too. Here the channel holding time corresponds to the duration of a packet transmission, and it can not be arbitrarily increased (unless we can perform packet aggregation).

In the implementation, we have chosen a minimum contention window equal to 32, which basically implies that the collision probability remains always less than 1/32. To further reduce the impact of collision, we propose to halve the pressure indicator of a link when it suffers from a primary collision. We do not see the need of decreasing the pressure indicator in the case of secondary collisions, because as explained earlier, the primary objective of our work is to protect primary transmissions - secondary transmissions come as a bonus.

VI. Numerical results

To illustrate the performance gains achieved with ContraFlow, we present here results obtained through simulations. We implemented an event-driven simulator that captures all aspects of the PHY (with self-interference cancellation) and MAC layers described in the previous sections and we used it to evaluate different network scenarios.

A. Simulation framework

1) Network topology: We have chosen to simulate networks of limited sizes but that illustrate well the efficiency and fairness issues of such distributed systems, and the improvements ContraFlow provide. Some of the topologies considered are presented in Figure 9 (dashed lines depicts the interference graphs). In addition we have considered the network presented in Figure 3 known for its fairness issues.

2) PHY and MAC layers: The PHY and MAC simulation parameters are given in in the table below, where in addition, the bandwidth was 1Mbs and the SNR model takes into account the fading model, transmit power and noise, as measured in the hardware. The pressure indicators are initialized to 10 in the simulations, however any value less than p_{max} will work as well. For updating the pressure indicators and calculating the access probability we divide each time slot into 32 minislots.

Parameter	**Value**
P_{max}	0.03125
CW_{min}	32
L_{max}	25 slots (800 mini-slots)
Ack	1 slot
SNR reception	10 dB
$p_{max} = P_{max} * L_{max}$	25
$SIFS$	1 slot
$DIFS$	2 slot

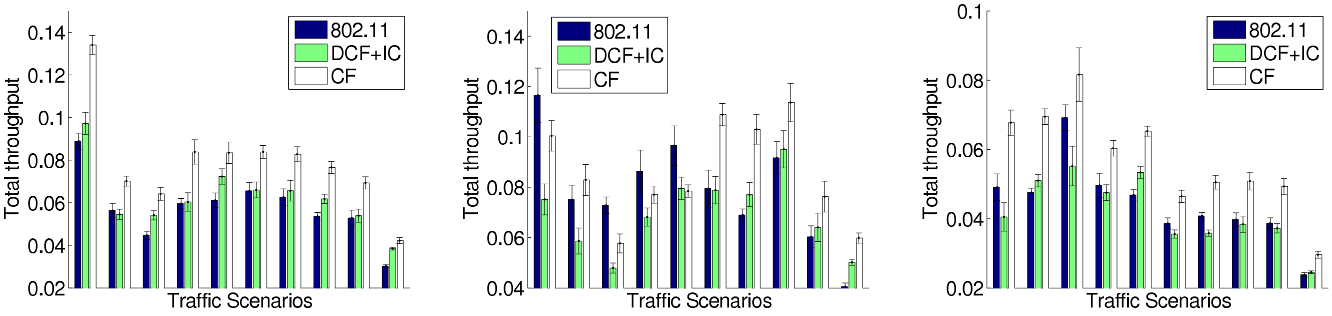


Fig. 10. Total throughput of network topologies 1 (left), 2 (middle) and 3 (right). X axis denotes different traffic matrices and Y axis is the total throughput.

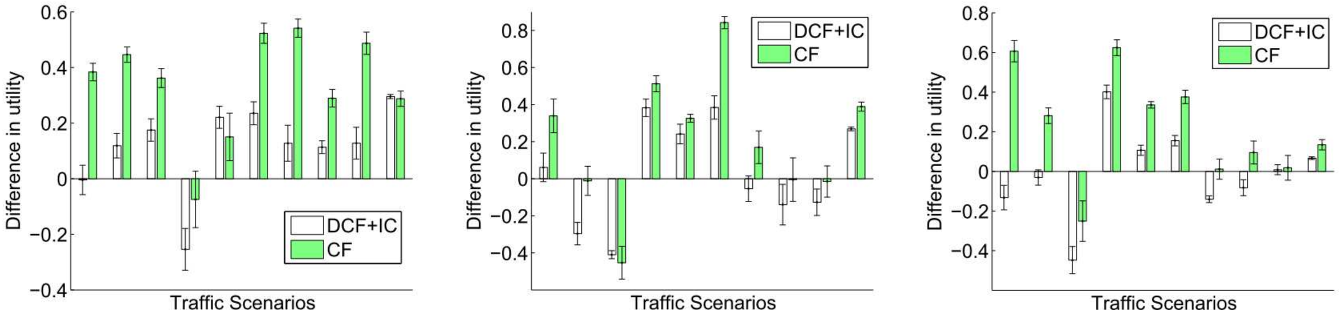


Fig. 11. Proportional fariness of network topologies 1 (left), 2 (middle) and 3 (right). X axis denotes different traffic matrices and Y axis is the absolute improvement in proportional fairness (difference in proportional fairness between DCF+IC and 802.11 and ContraFlow and 802.11, respectively).

We compare the performance obtained with 3 different systems. First we have simulated standard 802.11 protocol (e.g. with CSMA and DCF, without self-interference cancellation). Next, we have fully implemented ContraFlow (short CF), including self-interference cancellation and the new MAC scheduling algorithm. Finally, in order to compare the performance of our novel scheduler with the plain DCF, we implement a CSMA/DCF on the top of self-interference cancellation (short DCF+IC), as described next.

In DCF+IC each node implements the standard DCF to contend to become a primary transmitter. Having started to receive, the primary receiver needs to selects a secondary receiver from its weighted list $\mathcal{S}_A$ (where A denotes the primary sender; see Section IV-C). This is done using a threshold-based selection algorithm: the primary receiver selects the secondary receiver whose weight is higher that a given threshold $\mathtt{ST}$, and whose buffer is the largest (breaking tie uniformly). To give a chance to secondary receivers whose weights are below the threshold to increase their weights, every Y transmissions, the primary receiver randomly selects one of these receivers. That way, the weighted list can adapt to topology changes. The choice of Y depends on an estimate on how fast the network topology changes.

When the threshold is relatively low, it is very likely that the selected secondary receiver is not interfered by the primary sender. Indeed, assume for example that $\mathtt{ST} = 0.2$. If a node from $\mathcal{S}_A$ has a weight below the threshold, it must have experienced more than 4 successive transmission failures. It means that with high probability either it is interfered by the primary sender, or it suffers from a hidden terminal effect of another node (note that experiencing 4 direct successive collisions is very unlikely with DCF in absence of hidden terminals).

3) Traffic assumptions and performance metrics: For each topology, we have generated several traffic scenarios. For each scenario, at most one flow in each direction on each link is created. We evaluated the performance on many randomly generated traffic patterns, as well as on a few of them deliberately chosen to illustrate the issues we address with ContraFlow. All the sources are infinitely backlogged (we get similar results for TCP-like traffic but we omit them due to lack of space). For each topology, each system, and each traffic scenario, we have repeated the simulation 20 times. As for the performance metrics, we compare the total throughput and the utility (taking Proportional Fairness as the reference, i.e., the utility is the sum of the log of flow rates). Note that for example a difference of 2 in the utility in a network with 4 flows would approximately represents an average throughput gain of 60 % per flow (this gain would decrease to 28 % with 8 flows).

B. Results

1) Illustrative traffic scenarios: We now explain the benefits of ContraFlow on the network of Figure 3, known for its fairness issue (c.f. [12]). The throughputs observed on the 3 links are as follows: for simple 802.11 systems $(10.7 - 0.4 - 10.7)$ and with ContraFlow $(7.9 - 2.6 - 7.9)$. ContraFlow brings fairness at a good level (close to that of Proportional Fairness).

Another illustrative example is Scenario 1 in Figure 9, with flows $D \rightarrow A$, $A \rightarrow B$, $C \rightarrow A$ and $A \rightarrow E$. This is an access point scenarios where all nodes talk to the access point A in the middle, however some nodes (e.g. C) interfere with more nodes and other nodes (e.g. E) experience more hidden terminals. The rates of the 4 flows in the classic 802.11 systems are $(0.07 - 0.08 - 0.02 - 0.03)$ whereas ContraFlow achieves $(0.07 - 0.08 - 0.04 - 0.04)$. It gives higher rates to node C, who competes with three neighbors, and node E who competes with three hidden terminals.

2) Random traffic scenarios: We next evaluate the total throughput (Figure 10) and proportional fairness (Figure 11) for several random traffic metrics on the described topologies. We see that in many scenarios, ContraFlow has a higher total throughput, up to 30% to 50% over the plain DCF, but also over the DCF+IC. The improvement in throughput is due to the full-duplex nature of the transmission but also due to an efficient scheduling of dual links (as we can see when comparing to DCF+IC in Figure 10 and Figure 12).

	# dual links / # single links			
	DCF + IC		ContraFlow	
Topology 1	0.42	(0.08)	0.96	(0.2)
Topology 2	0.35	(0.05)	0.43	(0.08)
Topology 3	0.33	(0.05)	0.53	(0.1)

Fig. 12. The ratio of the number of dual links divided by the number of single links scheduled during a simulation run, averaged over all traffic scenarios (with 95% confidence intervals in brackets).

One might expect that the throughput improvement should be higher, up to 100%, as we can potentially transmit twice as many packets in one slot. There are three main reasons why this does not occur. Firstly, the packet loss rate increases in presence of self-interference cancellation (which is not ideal; see Section III). Secondly, different traffic patterns do not allow full exploatation of full duplex. Thirdly, the MAC design objective is to achieve proportional fairness, not to maximize the sum of rates. However, in spite of these, the improvement in average rate with ContraFlow is substantial. We also see that in the large majority of cases the utility is maximized when ContraFlow is used. We see that DCF+IC is not sufficient to improve the fairness in the system.

VII. Acknowledgements

Greg O'Shea for pchute/ukpapi, Steve Hodges and James Scott for helping us with hardware setup.

References

[1] Bozidar Radunovic, Dinan Gunawardena, Peter Key, Alexandre Proutiere, Nikhil Singh, Vlad Balan, and Gerald Dejean. Rethinking indoor wireless mesh design: Low power, low frequency, full-duplex. In *Fifth IEEE Workshop on Wireless Mesh Networks*, 2010.

[2] Jung Il Choi, Mayank Jain, Kannan Srinivasan, Philip Levis, and Sachin Katti. Achieving single channel, full duplex wireless communication. In *MobiCom '10: Proceedings of The 16th Annual International Conference on Mobile Computing and Networking*, 2010.

[3] Li Bin Jiang and Soung Chang Liew. Improving throughput and fairness by reducing exposed and hidden nodes in 802.11 networks. *IEEE Trans. on Mobile Computing*, 7:34–49, Jan. 2008.

[4] C. Fullmer and J. J. Garcia-Luna-Aceves. Floor acquisition multiple access (FAMA) for packet-radio networks. In *SIGCOMM*, pages 262–273, 1995.

[5] Z. J. Haas and Jing Deng. Busy tone multiple access (dbtma): A multiple access control scheme for ad hoc networks. *IEEE Trans. on Communications*, 50:975–985, June 2002.

[6] Mythili Vutukuru, Kyle Jamieson, and Hari Balakrishnan. Harnessing exposed terminals in wireless networks. In *Usenix NSDI*, 2008.

[7] D. Halperin, T. Anderson, and D. Wetherall. Taking the sting out of carrier sense: interference cancellation for wireless lans. In *MobiCom '08: Proceedings of the 14th ACM international conference on Mobile computing and networking*, pages 339–350, New York, NY, USA, 2008. ACM.

[8] Anand Raghavan, Edward Gebara, Emmanouil M. Tentzeris, and Joy Laskar. Analysis and design of an interference canceller for collocated radios. *IEEE Trans. on Microwave Theory and Techniques*, 53:3498–3509, Nov. 2005.

[9] F. Kelly, A. Maulloo, and D. Tan. Rate control in communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49:237–252, 1998.

[10] L. Jiang and J. Walrand. A distributed CSMA algorithm for throughput and utility maximization in wireless networks. In *Allerton conf. on control and comm.*, 2008.

[11] J. Liu, Y. Yi, A. Proutiere, M. Chiang, and V. Poor. Convergence and trade-off in utility-optimal CSMA. *(http://research.microsoft.com/pubs/70634/tr-2008-128.pdf)*, 2008.

[12] M. Garetto, T. Salonidis, and E. Knightly. Modeling per-flow throughput and capturing starvation in CSMA multi-hop wireless networks. In *IEEE Infocom*, 2006.