Efficiency and Fairness in Distributed Wireless Networks Through Self-interference Cancellation and Scheduling

Božidar Radunović, Dinan Gunawardena, Alexandre Proutiere,
Nikhil Singh, Vlad Balan, Peter Key

Microsoft Research,Cambridge, UK

Technical Report
MSR-TR-2009-27

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
http://www.research.microsoft.com

Abstract

Handling interference is one of the major challenges in the design of multi-user distributed wireless systems. In current systems, interference is managed through carrier sensing mechanisms such as CSMA/CA and through MAC algorithms based on random back-off. However, the asymmetry in channel sensing inevitably causes degraded throughput and fairness issues, such as those caused by hidden terminal problems. We propose ContraFlow, a solution based on self-interference cancellation and innovative scheduling mechanisms that increases spatial reuse, eliminates hidden terminals, and rectifies decentralized coordination inefficiencies among nodes, thereby improving fairness. Self-interference cancellation is a technique allowing a node to cancel its own transmitted signal and hence to successfully receive data while transmitting on the same channel. We demonstrate the feasibility of such techniques in a low power WPAN setting, using Lyrtech software-defined radios. Self-interference cancellation repairs carrier sensing, making it possible to successfully eliminate hidden terminal problems, even when using current multi-user MAC protocols; but it also provides the opportunity to design new distributed MAC scheduling algorithms that increase the spatial reuse and solve most of the fairness problems associated with current algorithms. We use simulations to 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.

1. INTRODUCTION

Handling interference is one of the major challenges in the design of multi-user wireless systems. Traditionally, to combat interference, transmissions on interfering links are made *orthogonal* in the time, frequency or code domain. Such coordination is relatively simple to enforce in centralized systems, such as 2G-3G cellular networks, but becomes notoriously problematic in wireless networks whose control mechanisms have to be distributed, such as for 802.11 and 802.15-based systems and related ad-hoc networks, mesh networks, WLANs and WPANs.

Such systems are interference limited, and use carrier sensing mechanisms (CSMA) and random back-off MAC algorithms (such as the Distributed Coordination Function (DCF) of 802.11) to attempt to separate transmissions on interfering links in time. However, the current carrier sensing mechanisms are too simple to capture the actual structure of interference in the network. Collisions occur due to *hidden* terminals, and spatial reuse is degraded due to *exposed* terminals. Hidden terminals are nodes whose transmissions interfere, but which are not prevented by the carrier sensing mechanism from transmitting simultaneously. Exposed terminals are nodes that do not interfere, but are unnecessarily made simultaneously inactive by the carrier sensing mechanisms. Various virtual carrier sensing partial solutions have been proposed, such as the 802.11 use of RTS/CTS, a signaling procedure designed to resolve the hidden terminal problem. However, the over-head it generates significantly reduces the system throughput, and hence it is rarely deployed in practice. Lack of fairness is another deficiency of current sensing mechanisms and MAC algorithms: for instance, when a link has more interferers than others, it senses the medium busy a higher proportion of time than its neighbors, and when it finally observes the channel idle, it competes with more transmitters, experiences more collisions and hence gets a reduced throughput.

In this paper, we propose *ContraFlow*, a solution to alleviate the throughput and fairness issues in wireless networks with distributed control. This solution is based on *Self-Interference Cancellation* (SIC), allowing a node to transmit and receive successfully and simultaneously on the same channel. In-coming and out-going links from a node usually interfere, hence simultaneous transmissions and reception at a node requires several orthogonal channels. Interference Cancellation (IC) allows both transmissions on the *same* channel: to receive a packet while transmitting, the node first decodes the relatively strong signal it is transmitting, and then uses this information to decode the received signal. We show using a software radio test-bed that this is indeed possible in a WPAN setting (see Section 2 for detail).

ContraFlow exploits this functionality 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 SIC to decode the packet sent by A. Similarly A uses SIC 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 is not successful. This scheme has two immediate advantages:

(i) Eliminating hidden terminals. First, 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 and instantaneous RTS/CTS procedure. It has the same advantages of RTS/CTS and it is much less likely to be missed. Hence ContraFlow provides an efficient way of canceling the impact of hidden terminals, see Figure 1(a).

(ii) Enabling dual links. Second, since B is actually also able to send a packet while receiving, successful transmissions on two links that would interfere without IC are made possible. We refer to such pairs of links as *dual-links*, and provide examples of symmetric and asymmetric dual-links in Figure 1(b). With dual-links, we can theoretically double the feasible throughput of the network.

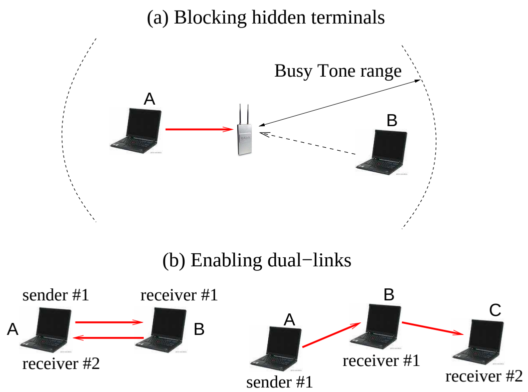


Figure 1: (a) Handling hidden terminals: The transmission from A to the access point is protected from the hidden node B by the busy tone/packet sent by the access point while receiving the packet from A. (b) Enabling dual links: using self-interference cancellation at nodes A and B, we can activate two links simultaneously, i.e., dual links - (left) a symmetric dual link (A, B, A), (left) an asymmetric dual link (A, B, C).

Self-interference cancellation could be implemented without modifying the scheduling MAC protocols, but the inherent fairness issues arising in network using CSMA would then not be solved. To tackle these issues, ContraFlow implements new scheduling MAC protocols, where nodes (or links) that are interfered with by more links than others get more opportunities than others to start transmitting when they sense the channel idle. This balances the fact that these nodes sense the channel idle less often than others. With these protocols, a node that does not manage to access the channel tends to decrease its contention window (increase its channel access rate), and this is necessary if we seek to improve fairness. It is important to note that this principle cannot be applied if the hidden terminal problem is not solved (with the hidden terminal problem, the transmissions of two hidden nodes typically collide, and hence these nodes would not access the channel successfully and decrease their contention window, which in turn further exacerbates the problem). Hence to ensure fairness, it is crucial to first address hidden terminals, which ContraFlow does.

IC is not a new concept, as it has been used in the 70's for example by Cover to analyze the information-theoretical capacity of the broadcast channel [1]. IC is a multi-user detection technique allowing a receiver to successfully decode several signals with different strengths: it sequentially decodes the signals in decreasing order of strengths, where the previously decoded signals is used to detect the next signal. It has always been considered as one of the most promising multi-user detection techniques, but has remained relatively challenging to implement in practice, see [2, 3]. Two major implementation issues are that receiver complexity badly scales with the number of signals to decode, and that IC requires precise estimates of the channels between the transmitters and the receiver and of other factors impacting the received signal. Our MAC guarantees that there will be at most two received signals at one time. We wish to cancel the interference of only one signal, one is almost known– since the transmitter and receiver are collocated, which simplifies the cancellation problem.

We implement and validate the concepts of ContraFlow in a Wireless Personal Area Network (WPAN) testbed based on configurable radio hardware (namely Lyrtech software-defined radios). The IC implementation is made in a FPGA, which is easily transferable in ASICs, does not require memory access, off-line processing, or storing waveforms. We propose two possible IC schemes, the *digital* and the *analog* IC schemes. The former can be implemented in real time in our testbed, and manages to reduce self-interference by more than 30 dB when interference is generated by a busy tone, and 20 dB when it corresponds to a packet transmission. The latter has to be implemented off-line due to the specific architecture of Lyrtech architecture, but performs much better the digital IC; it reduces interference by 40 dB or 30 dB in the case of busy tone or packet transmission, respectively. Both IC schemes allow ContraFlow to deliver the expected performance improvements.

Finally we simulate network performance using the channel measurements from the testbed. As our results suggest, ContraFlows first removes most of the hidden terminal problems and then provides significant gains both in terms of system throughput and fairness.

The paper is organized as follows: In the next section, we describe and discuss the two proposed IC schemes and their implementations on Lyrtech radio boards. In Section 3, we present the modifications we made at the MAC layer to handle dual links enabled in ContraFlow. Section 4 is devoted to the new random back-off MAC scheduling protocols with improved fairness. We present numerical evaluation in Section 5, and the related work in Section 6.

2. SELF-INTERFERENCE CANCELLATION IN PRACTICE

In this section, we first outline the well-known concept of interference cancellation, and explain the technical challenges faced when implementing it. We then describe how we built a system using Lyrtech software-defined radio cards, and give empirical results showing that we can efficiently cancel interference in practice in our WPAN testbed. Our baseline design implements 802.11b PHY, although it can easily be extended to other designs.

2.1 From theory to practice

Interference cancellation is simple to explain theoretically: if two received signals $r_1(t)$ and $r_2(t)$, with respective powers P_1 and P_2, interfere, the receiver could potentially decode both, provided that one is stronger than the other, say $P_1 > P_2$. The receiver first estimates the signal $r_1(t)$ treating the second signal as noise. A good estimation can be performed if the corresponding SINR $= P_1/(P_2 + N)$ is large (where N denotes the power of thermal noise). The receiver then subtracts the estimated signal $\hat{r}_1(t)$ from the received signal, thus obtaining $r_2(t) + [r_1(t) - \hat{r}_1(t)] + n(t)$. If the error in estimating $r_1(t)$ is small enough, the receiver successfully cancels the interference of the first signal and hence can decode the second one.

In practice, the received signal $r_1(t)$ is a complicated functional of the signal $s_1(t)$ sent by the source, resulting from the perturbations introduced by electronic circuitry, the antennas, and radio fading. This functional has a non-linear component $E(s_1(t), t)$ arising from the electronic circuitry that is difficult to estimate. It also has a linear component from multi-path propagation. As the result the received signal has the following form:

$$r_1(t) = \sum_{i \text{ path}} h_i \times s_1(t - d_i) + E(s_1(t), t),$$

$$\text{with } s_1(t) = \kappa \sin((\omega + \delta\omega(t))t + \phi(t + \delta t))$$

where $\phi \in \{\pi/4, 3\pi/4, 5\pi/4, 7\pi/4\}$ for QPSK, and where the sum over i models multipath fading, and h_i and d_i are the channel gain and delay along the i-th path. The above formulas illustrate the challenges in practical implementations of interference cancellation. To obtain a precise estimate $\hat{r}_1(t)$ of $r_1(t)$, one needs: (i) to deal with channel uncertainty using the linear model, i.e., to estimate the various multipath gains and delays h_i, d_i, ignoring $E(s, t)$ for simplicity; (ii) to deal with the carrier phase uncertainty, i.e., to estimate the unknown carrier shift $\delta\omega(t)$; (iii) to estimate the symbol sent, here represented by the function ϕ; (iv) to deal with the symbol timing uncertainty, i.e., to estimate the time shift δt.

Carriers are slowly changing sine waves and hence easy to estimate and correct. Moreover, any carrier phase estimation error is not catastrophic. However a symbol time estimation error translates into a misplaced phase- shift (a complete change in the signal's amplitude). This results in a large impulsive residual noise. For a similar reason the phase shifts and resultant symbol changes also emphasize the reconstruction errors caused by the channel and the circuit uncertainty. When sending a busy-tone, which is a repeated sequence of bits (e.g. '0' bits) we send a plain, unmodulated carrier. Noise cancellation is easier since the channel reconstruction error is smaller.

Note that for our special case of self-cancellation, we *know* the symbols themselves (the ϕs) and the symbol timing uncertainty (the δt), since we are both transmitting and receiving on the same device.

Interference cancellation schemes perform well in asymmetric scenarios, where one signal is stronger than the other, but where the asymmetry is not too large . Indeed, the residual interference $r_1(t) - \hat{r}_1(t)$ typically grows with the received power P_1 of the signal. For example, in [3], the interference cancellation implementation can decode successfully two interfering signals when their power ratio remains roughly between 3 and 30 dB. We wish to cancel interference in very asymmetric scenarios, since we seek to cancel self-interference. Our task is simplified by the fact that the co-location of transmitter and receiver, hence some of the components of $r_1(t)$ are either known (e.g. the symbol and its time shift) or easier to estimate (e.g. the path gains and delays, the carrier phase shift). As a consequence, we are able to design interference cancellation schemes that perform well even in very asymmetric scenarios.

2.2 The Lyrtech platform

To assess the feasibility of self-interference cancellation, we use a configurable radio hardware, namely a Lyrtech Small-Form Factor Software Defined Radio platform [4]. It consists of a radio module with ADC/DAC conversion, an FPGA board and a DSP board. The architecture of the platform is illustrated in Figure 2.

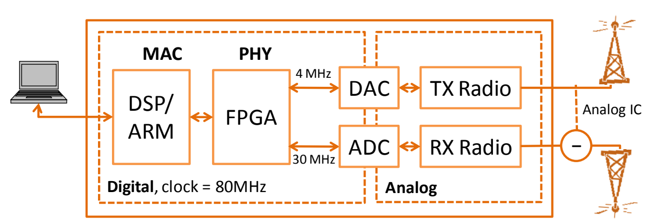


Figure 2: The Lyrtech platform

The radio board operates on TV band frequencies. It comprises separate transmission and reception circuits, each having its own antenna. The two antennas are approximately 10 cm apart. The antenna separation weakens the received self-signal to the extent that it does not saturate the 14-bit AD converter and can be further processed and canceled digitally (although this is not an issue in analog IC).

The radio module operates at 20 MHz bandwidth. It uses a fixed transmission power roughly equal to 0.5mW, which corresponds to the most common WPAN technologies, such as UWB, Zigbee and Bluetooth.

We implemented an 802.11b-compatible physical layer using a single rate of 1 Mbps. It uses QPSK modulation and CDMA spreading. Both the transmitter and the receiver paths of the physical layer are fully implemented in FPGA. The transmitter path receives a MAC frame of a packet from the DSP, spreads it, modulates it, and transmits it. The received signal at the output of ADC is a modulated 30 MHz carrier. The receiver path synchronizes to the 30 MHz carrier (using a Costas loop, see [5] for details), estimates the symbols and the symbol timing, despreads them, detects the packet preamble and delivers the decoded packet to the DSP. It also provides the carrier sensing functionality. The interference cancellation schemes presented below are also implemented in the FPGA; whereas MAC layer mechanisms are performed in the DSP. The latter interfaces with FPGA, and it implements the outline of the dual-link protocol presented in Section 3.

2.3 Interference cancellation schemes

We now present two self-interference cancellation schemes, one digital and other analog. We implement the former in real time in our platform. The Lyrtech board cannot be modified to provide a real time implementation for the latter scheme, hence we implemented an off-line algorithm to evaluate its performance.

2.3.1 Digital cancellation

This interference cancellation scheme is a digital implementation of a scheme described above, i.e., at a given node (board), we estimate the part of the received signal corresponding to that sent by the same node $\hat{r}_1(t)$, and subtracts it from the received signal to be left with the signal plus noise coming from other nodes. This scheme is illustrated in Figure 3.

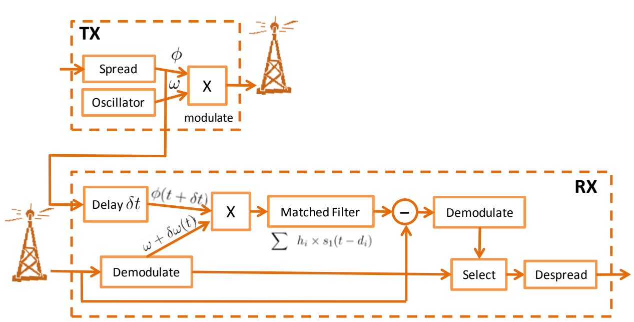


Figure 3: Digital cancellation.

Interference is created locally by the node willing to cancel interference, which simplifies the interference cancellation scheme and improves its performance as explained below.

(i) *Channel estimation.* We observe that due to proximity of the receiving and transmitting antenna, the impact of fading is almost time-invariant, which allows us to estimate this impact off-line. Standard techniques, see e.g. [6], are then sufficient to obtain very accurate estimates of the channel.
(ii) *Carrier phase uncertainty.* We estimate this uncertainty using standard Phase-Locked-Loop (PLL) techniques (c.f. [5]).
(iii)-(iv) *Symbol and symbol timing uncertainty.* We eliminate this type of uncertainty. We observe that the delay between the time when a symbol is transmitted and the time at which the same symbol is received on the other antenna is fixed. Since we know the transmitted symbols and their transmission times at the transmitter circuit in the FPGA, we convey this information to the receiver circuit.
(v) *Non-linear circuitry effect.* This effect is more difficult to estimate (although we observe that it is time-invariant). Hence we ignore it, and treat it as noise, which is the main source of the possible inefficiencies of the digital cancellation scheme.

We illustrate the performance of the digital cancellation scheme in Figure 4. Blue points are SNR levels measured at different distances between a transmitter and a receiver. The SNRs are given relative to the thermal noise and are measured in FPGA. The dashed line is the level of the self-interference, received over the 10cm link from the transmit to the receive antenna. The thick red arrows denote the performance of the two IC techniques. In the case of digital IC we eliminate 22dB of noise when canceling a packet (the upper arrow) and 32 dB of noise when canceling a busy tone (the upper and the lower arrow). As an illustration, we are able to receive a signal at a distance of 3m with an SNR = 8dB after the digital IC.

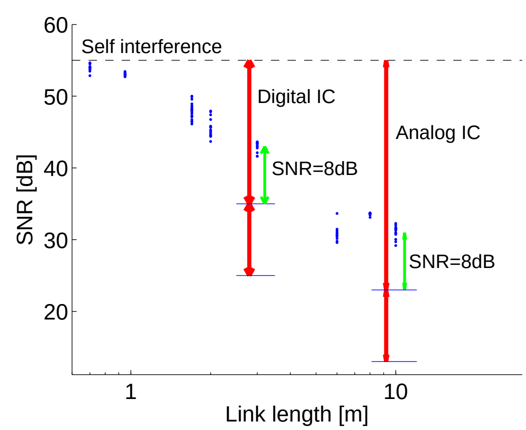


Figure 4: Performance of the cancellation schemes.

2.3.2 *Analog cancellation*

The main source of noise in the digital cancellation scheme stems from the possible inefficiencies of the various estimations that have to be performed, dominated by the non-linear circuitry effect. Also, a strong self-interfering signal can saturate the DAC and no further digital processing would be possible. Finally, different digital IC have to be implemented for different PHYs. In the proposed analog cancellation scheme, we cancel this effect by producing two copies of the transmitted signal to the receiver: one through the air and the other through a wire, where both passed through the transmitter and the receiver circuitry. As a consequence, both signals will be exposed to the same transfer functions of all the circuitry. The only difference between the signals is caused by signal propagation through the air, however this effect is small, since the antennas are physically close, and can be estimated since the transfer function is predominately linear and slowly changing.

We cannot modify our boards to implement such a scheme in real time. Instead, we performed the following experiment. We first transmitted a series of packets through the air from one antenna to the other on the same board. We the repeated the experiment but sending through a wire (coax). We then compared the received signals. Note that there is a clock drift in the radio which varies in time, and hence differs for different transmissions. We selected transmissions with similar clock drifts and compared them. The results are presented in Figure 4. The thick red arrow on the right denotes the performance of the Analog IC. We eliminate 32dB of noise when canceling a packet (the upper arrow) and 40 dB of noise when canceling a busy tone (the upper and the lower arrow).

2.4 Performance on dual-links: Digital cancellation

We now give results from experiments using digital 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. In Figure 5 (left), we compare the performances of an isolated link (A, B) when B does or does not send a busy tone. In the former case, B implements the digital cancellation scheme. The performance metric is the packet success rate, and each point represents the average success rate over 500 packets of size 1000 bytes. Here A and B are at distance 1.5 m, but we observed that the performance does not really vary as the distance does not exceed more than 2m. In Figure 5 (right), we evaluate the packet success rates for dual symmetric (ABA) and asymmetric (ABC) dual links. 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. 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).

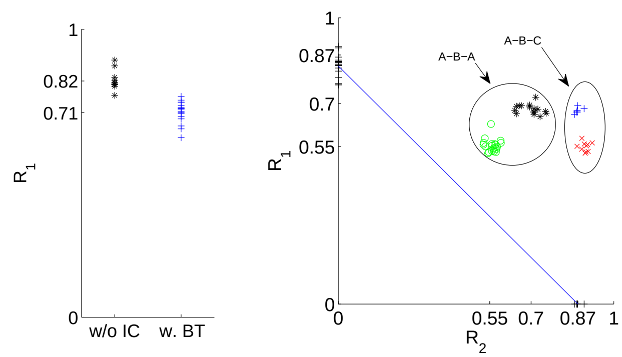


Figure 5: Average performance of dual links.

2.5 Discussion

We conclude the section with a few remarks.

Towards analog cancellation. Although we have presented both digital and analog self-interference scheme, it should be clear that in future, only analog cancellation needs to be considered, for performance reasons and because it is (conceptually) easier to implement. Our SDR platform did not allow such an implementation in real time, and instead we implemented the digital cancellation scheme that achieves lower, but yet good, performance as shown in §2.4.

Performance issues: By allowing concurrent, full-duplex transmission, we have increased the number of transmissions but also the overall noise in the system. We measured benefits for the 802.11b PHY. It remains as a future work to study what are the limits of the analog IC and its implications for PHY designs.

Power and distance scaling: An increase in the transmission power will not impact the performance of a system with self interference cancellation. It will raise both the level of the received signal and the residual interference after the IC, but the SNR will remain the same.

Consequently, in order to increase the dimensions of a network with IC, raising the transmission power will not help; one has to further improve the interference cancellation techniques instead.

3. HANDLING DUAL LINKS

We now detail the MAC-layer mechanisms involved in ContraFlow. We first explain how we handle dual-links using interference cancellation. Then, we show how interference cancellation can be exploited in the design of random back-off MAC algorithms ensuring both efficiency and fairness.

3.1 Packet and ACK transmissions on dual-links

We start by describing how ContraFlow initiate and handle dual-links. In the following we refer to the node initiating a link or a dual link as the *primary sender*, to the node receiving the packet sent by the primary sender as a *primary receiver*, and in the case of dual-links, the node receiving the packet sent by the primary receiver as the *secondary receiver*. For example, in the system presented in Figure 1.b (left), node A is the primary sender and the secondary receiver, and B is the primary receiver.

Here is a chronological description of how nodes initiate and handle transmissions. Figure 6 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 either transmit a busy tone or 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 §3.2.

(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 $[t_A, t_{end} + \texttt{ACK1}]$, 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 $t_{end} + \texttt{ACK1}$. 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*.

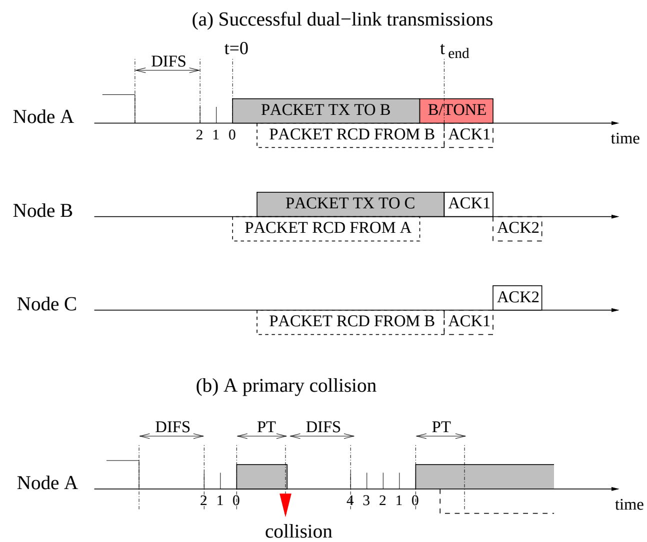
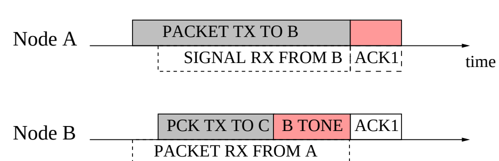


Figure 6: (a) An example of successful transmissions on an asymmetric dual-link. (b) A primary collision experienced by node A.

We impose that the secondary transmission ends almost 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 (because the secondary receiver does not send a signal while receiving, except in the case of symmetric dual link, i.e., the primary sender is also the secondary receiver). 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 below.



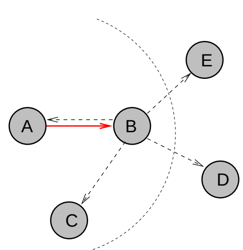
It is important that the primary sender and receiver transmit a busy tone during the first MAC ack and before the end of the primary transmission, respectively, so as to occupy the medium to protect the transmissions of both MAC acks. Otherwise, the medium could be sensed idle for a duration greater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

3.2 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.

There are two causes for secondary collisions: they can be due either (1) to the interference structure of the network, i.e., the primary sender interferes at the secondary receiver; or (2) to 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 numerical experiments 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

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

3.3 Impact of PHY-layer design

The way interference cancellation is implemented in our platform impacts the structure of a transmitted packet. The latter depends on whether indirect or direct interference cancellation is used.

Indirect cancellation at the primary receiver. In this case, we have specific synchronization issues. When node B starts receiving a packet from node A, it becomes a secondary transmitter. It first needs to parse the header of the packet to find out who the transmitter is. The transmission of the PHY preamble and the MAC header by the primary transmitter takes δ_1, roughly equal to 300 μs in our implementation. Then node B needs to select a node from $\mathcal{S}_A$ and starts transmitting. This is done at the MAC layer which is executed at the DSP, and once a packet is selected, it has to be transferred to the FPGA and to the radio board, which takes δ_2 ranging from 100 μs to 200μs in our platform. Once node B is transmitting, it needs to cancel its own signal. As described in Section 2, it needs to lock on the phase of its signal in order to reconstruct it, which takes δ_3 up to 50 μs. To allow node B to perform these operations, the primary sender A sends, after the PHY preamble and MAC header, a busy tone for a duration $\delta_4 \geq \delta_2 + \delta_3$. Finally, after starting transmitting, node B is no longer synchronized on the signal sent by A and it has to lock on it again (lock on the phase, symbol time and orientation) which is why after the busy tone, A resends the preamble. All these procedures are illustrated in Figure 8(a). Note that the overhead due to this synchronization issue is δ_4 plus the duration of the preamble, i.e., around 450 μs in our implementation. This roughly represents 5% of the entire packet transmission for a 1000 bytes payload at 1Mbps. Also note that these synchronization issues do not exist at the primary sender.

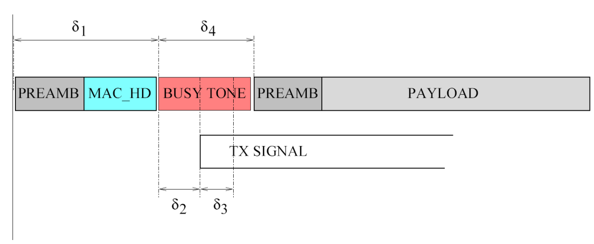


Figure 8: A structure of a packet transmitted on a dual link. The figure depicts a secondary transmitter receiving a packet, and starting a secondary transmission in the case of indirect cancellation.

Direct cancellation. In this case, there is no synchronization issues as discussed in Section 2, and the primary sender can send the packet payload right after the MAC header.

3.4 Discussion

We conclude this section by two remarks on the MAC-layer design as presented above.

Dual links are always initiated by the primary sender. As shown above, a dual link (A, B, C) is initiated by A, the primary sender. This may not always be the best choice, for example if B has no packets to send. However, this is the only possible choice. Indeed if B initiates the dual link, after it starts transmitting, the channel is seen busy by node A. Then the latter has no simple mean of deciding, using carrier sensing, if it can send to B without causing a collision.

Spectral reuse and exposed terminals. 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 9. 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).

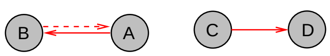


Figure 9: Dual links mitigate the impact of exposed terminals.

4. RANDOM MAC ALGORITHMS

In this section, we propose several random MAC scheduling algorithms suited to exploit the interference cancellation feature of ContraFlow. We first start by explaining how to adapt the current MAC algorithms (e.g. those of the 802.11 or 802.15 standards) to ContraFlow architecture. Then we present new random back-off MAC algorithms that aim at solving the inherent fairness issues of protocols that build over CSMA/CA.

4.1 Using DCF with ContraFlow

ContraFlow could be used without modifying the MAC protocols of 802.11 standard, i.e., the DCF. The latter specifies how each node attempts to use the channel when it has packet to transmit. More specifically, each primary sender accesses the medium using DCF: (i) when a node whose buffers are initially empty receives a packet to transmit, it will do so after observing a medium idle period of duration DIFS; (ii) when its buffers are not empty, after a successful primary transmission, it randomly picks a back-off counter uniformly between 0 and $\texttt{CW}_{\min} - 1$, and after a primary collision, it picks a back-off counter uniformly between 0 and $2^{\min(7,m+1)}\texttt{CW}_{\min} - 1$, where it chose this counter between 0 and $2^m\texttt{CW}_{\min} - 1$ for the previous un-successful transmission.

It remains to specify how the primary receiver selects a secondary receiver from its weighted list $\mathcal{S}_A$ where A denotes the primary sender. We propose a *threshold-based selection* algorithm. A primary receiver selects the secondary receiver whose weight is higher that a given threshold 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 $\texttt{ST} = 0.2$. 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).

4.2 Fair random algorithms for ContraFlow

The lack of fairness observed in 802.11 or 802.15-based system is caused by the carrier sensing mechanisms and to basic principle of the DCF. The problem is classically illustrated in a 3-link network, without hidden terminals (see Figure 10).

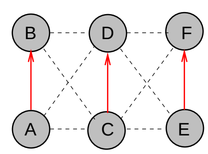


Figure 10: A 3-link network with fairness issues. A dashed line between two nodes mean that they sense each other.

Link 1 (A, B) and 3 (E, F) do not interfere with each other, whereas they interfere and interfered with 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.

As illustrated, fairness problems are caused by the bad interaction of the carrier sensing mechanisms and of the MAC algorithm (DCF). Keeping CSMA, the only way of ensuring fairness is to increase the transmission probability of nodes that do not get access to the channel, and decreasing that of nodes being served. 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.

We propose a random access MAC protocols that does not involve any message passing and 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 [7] to the dual-link model. 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. In the following description we assume slotted time in order to simplify the presentation.

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:

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

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 $[0, 2\mathtt{CW}_n[t]-1]$ where $\mathtt{CW}_n[t] = 1/P_n[t]$. $P_n[t]$ is the access probability of node n related to the pressure indicators of all out-going links from n:

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

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. It can be established [7, 8] that without the dual link feature of ContraFlow, without hidden terminals, and with packets of fixed and identical sizes, the algorithm tends to a *proportionally fair* sharing of the resources [9] when ϵ is small enough, i.e., it maximizes the sum of the log of the long-term throughput of the various links, which can be regarded as a good fairness criterion. The analysis of the convergence properties of the algorithm combined with ContraFlow is beyond the scope of this paper. However, as shown experimentally in the next section, the algorithm indeed converges and significantly improves fairness.

Finally, to choose the second link composing the dual-links, we pick the node maximizing $\mathtt{weight}_C \times \mathtt{p}_m[t]$.

4.3 Handling collisions

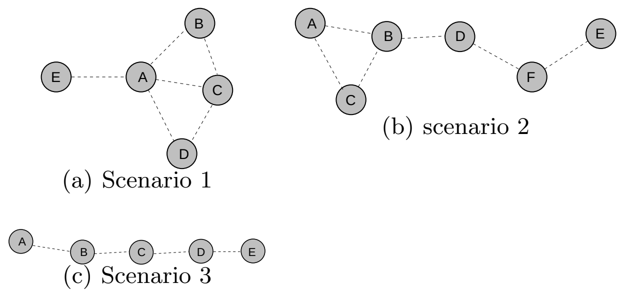
In the fair MAC protocols we just presented, when transmitters do not access the channel, their pressure indicator increases, which in turn increase their transmission probability. In particular, nodes increase their transmission probability upon collisions, which can be problematic. That is why it is 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 16, which basically implies that the collision probability remains always less than 1/16. 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.

5. 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 digital IC) and



(a) Scenario 1

(b) scenario 2

(c) Scenario 3

Figure 11: Network scenarios.

Parameter	Value
P_max	132
CW_{min}	16
L_{max}	25 slots
Ack	1 slots
SNR reception	10 dB
$\rho_{max} = P_{max} * L_{max}$	18.75
q_{max}	18.75
$SIFS$	1 slot
$DIFS$	2 slot

Table 1: Simulation parameters

MAC layers described in the previous sections and we used it to evaluate different network scenarios. Next we explain the simulation setup, and then provide the results.

5.1 Simulation framework

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 11 (dashed lines depicts the interference graphs). In addition we have considered the network presented in Figure 10 known for its fairness issues.

PHY and MAC layers. We compare the performance obtained with 3 different systems: (i) first we have simulated standard 802.11 protocol (e.g. with CSMA and DCF, without SIC); (ii) then we have complemented the previous system with SIC; (iii) Finally, we have fully implemented ContraFlow, including SIC and the new MAC scheduling algorithm.

The simulation parameters are given in Table 1, where in addition, the bandwidth was 1Mbs and the SNR model takes into account the fading model, transmit power and noise.

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 have randomly generated traffic patterns, except for a few of them deliberately chosen to illustrate the issues we address with ContraFlow. We assume 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).

5.2 Results

5.2.1 *Illustrative traffic scenarios*

We now explain the benefits of ContraFlow on the network of Figure 10, known for its fairness issue (c.f. [10]). To further exacerbate this issue, we artificially increase the packet TX duration to 300 slots (only in this example), and use UDP traffic. 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 11, 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.

5.2.2 *Random traffic scenarios*

We next evaluate the total throughput (Figure 12) and proportional fairness (Figure 13) for several random traffic metrics on the described topologies. We see that in many scenarios, ContraFlow has higher total throughput due to full-duplex nature. In some scenarios this is not the case because of relatively high loss rate with Direct IC. We also see that in the large majority of cases the utility is maximized when CotraFlow is used. IC with DCF is not sufficient to improve the fairness in the system.

6. RELATED WORK

There has recently been an important research effort to improve the throughput and fairness of CSMA-based distributed wireless systems, by proposing solutions at the PHY, MAC, or even higher layers.

The use of busy tone has been proposed in [11] to combat hidden terminals, but this requires a second sig-

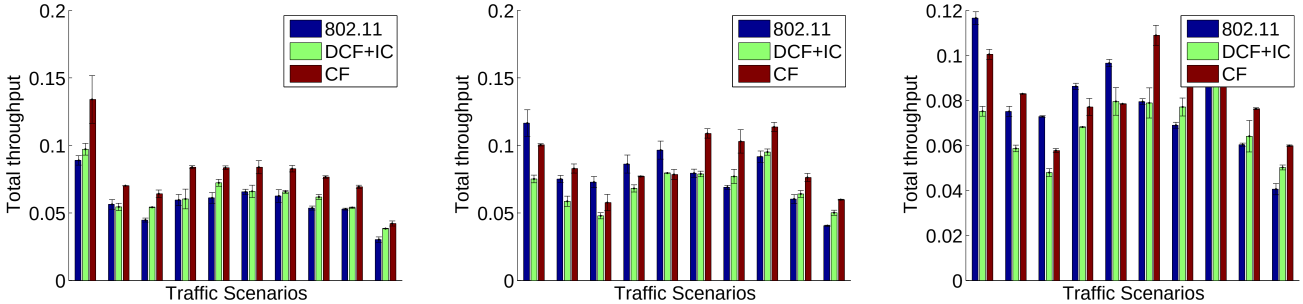


Figure 12: Total throughput of network topologies 1 (left), 2 (middle) and 3 (right). X axis denotes different traffic matrices and Y axis is the total throughput.

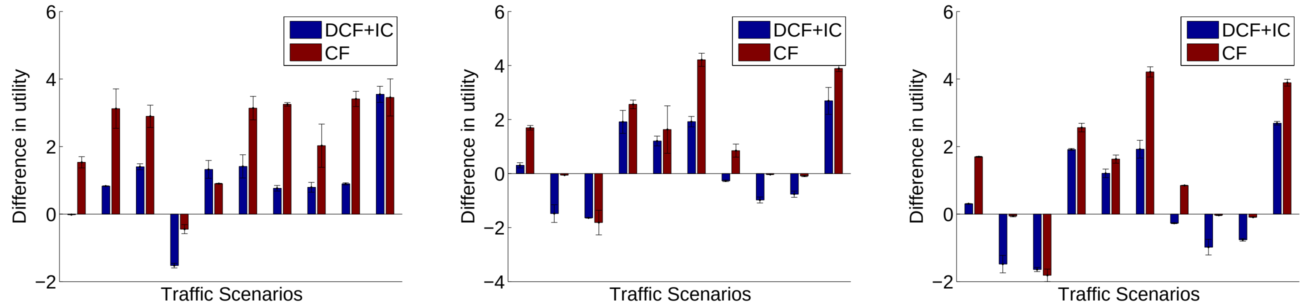


Figure 13: Proportional fariness of network topologies 1 (left), 2 (middle) and 3 (right). X axis denotes different traffic matrices and Y axis is the absolute improvement in proportional fairness (difference in proportional fairness between DCF+IC and 802.11 and ContraFlow and 802.11, respectively).

naling channel (which we don't need here). The use of directional antennas could help improving throughput as shown in [12]; the use of MIMO techniques help as well. It is worth noting that our nodes have two antennas, and that in theory, our proposed scheme outperforms the capacity of a 2x2 MIMO channel [13], just because in our case, both nodes use their maximum power, so the total used power is twice greater than that used in a 2x2 MIMO system. To improve efficiency, smart Multi-User Detection techniques have been also advocated recently, such as those presented in [14,15]. Again in theory, ContraFlow should do as good or better than these techniques. Implementing IC in distributed wireless systems has been recently proposed in [3,16]; there the authors suggest that interference from other nodes (not self-interference only) can be cancelled, but then, the design of appropriate MAC scheduling algorithms for this kind of systems remains unclear. It should be noted that in the above mentioned related solutions, the implementation is made on GNU software defined radios, and hence has to be performed off-line, whereas we have implemented ContraFlow in real time. Also note that most of the aforementioned approaches are orthogonal and hence complementary to ours. Finally, the feasibility of analog self-interference cancellation has been shown in [17], where similar interference reductions as those observed in ContraFlow are obtained.

At the MAC layers, many solutions have been tested to eliminate hidden and exposed terminals, see e.g. [18] and references therein. A recent interesting proposal is to alleviate the interference impact by learning the interference map, and taking scheduling decisions according to this map. At higher layers, network coding could also boost the system throughput, as demonstrated in [19]. Again these approaches are complementary to ours.

Fairness issues are well-known in CSMA networks, and they have been modeled analytically, see e.g. [10]. Theoretical solutions to these issues have been studied, e.g. in [20], but their practical implementations would require heavy message passing procedures as in [21], which introduces a significant overhead. It is worth mentioning the solution proposed in [22], where basically, a node interfered by more links than others would artificially creates *collisions* and increase the contention windows of its neighbors so as to get more opportunities to transmit. The overhead introduced by these collisions could be potentially important. The MAC scheduling algorithms presented here and implemented do not have this problem, and seems to constitute the most promising solution to fairness issues.

7. REFERENCES

[1] Thomas M. Cover. Broadcast channels. *IEEE Trans. Info. Theory*, pages 2–14, 1972.

[2] J.G. Andrews. Interference cancellation for cellular systems: a contemporary overview. *Wireless Communications, IEEE*, 12(2):19–29, April 2005.

[3] 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.

[4] `http://www.lyrtech.com/`.

[5] J. Hamkins (Ed), M. K. Simon (Ed.), and J. H. Yuen (Series Ed.). *Autonomous Software-Defined Radio Receivers for Deep Space Applications.* Wiley, 2006.

[6] `http://www.tools4sdr.com/`.

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

[8] 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)*.

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

[11] 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, June 2002.

[12] Anand Prabhu Subramanian and Samir R. Das. Addressing deafness and hidden terminal problem in directional antenna based wireless multi-hop networks. *ACM/Kluwer Wireless Networks (WINET 2008) Journal, Special Issue on best papers of COMSWARE 2007*, 2008.

[13] E. Teletar. Capacity of multi-antenna gaussian channels. 10:585–596, 1999.

[14] Shyamnath Gollakota and Dina Katabi. Zigzag decoding: combating hidden terminals in wireless networks. In *ACM Sigcomm*, pages 159–170, New York, NY, USA, 2008. ACM.

[15] Sachin Katti, Shyamnath Gollakota, and Dina Katabi. Embracing wireless interference: Analog network coding. In *ACM Sigcomm*, 2007.

[16] Daniel Halperin, Thomas Anderson, and David Wetherall. Interference cancellation: better receivers for a new wireless mac. In *Sixth Workshop on Hot Topics in Networks, HotNets-VI*, November 2007.

[17] 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, Nov. 2005.

[18] 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, Jan. 2008.

[19] Sachin Katti, Hariharan Rahul, Wenjun Hu, Dina Katabi, Muriel Medard, and Jon Crowcroft. Xors in the air: Practical wireless network coding. In *ACM Sigcomm*, 2006.

[20] M. J. Neely, E. Modiano, and C. Li. Fairness and optimal stochastic control for heterogeneous networks. *IEEE/ACM Transactions on Networking*, 16:369–409, April 2008.

[21] E. Modiano, D. Shah, and G. Zussman. Maximizing throughput in wireless networks via gossip. In *ACM Sigmtrics/Performance*, 2006.

[22] Ying Jian and Shigang Chen. Can csma/ca networks be made fair? In *ACM Mobicom*, 2008.