

Known Interference Cancellation: Resolving Collisions due to Repeated Transmissions

Chuan Qin^{*}, Naveen Santhapuri[†], Souvik Sen[†], Srihari Nelakuditi^{*}

^{*} University of South Carolina [†] Duke University

Abstract—Reception of duplicate packets by a node in a wireless network is a common occurrence. Reasons for repeated transmissions range from broadcast flooding to multicast streaming to unicast forwarding. These repeated transmissions may also get involved in collisions like other original transmissions. We argue that when one of the colliding packets is previously overheard, its interference can be cancelled to decode the other packet. In other words, when a receiver overhears a packet, it becomes effectively immune to the interference caused by the packet’s subsequent transmission. We refer to this as *known interference cancellation* (KIC). In this paper, we identify the scenarios in which KIC is applicable. We then implement KIC on USRP/GnuRadio testbed to demonstrate its feasibility and conduct QualNet simulations to illustrate its potential performance gain.

I. INTRODUCTION

It is typical for a node in a wireless network to receive or overhear two copies of the same packet (barring a few fields in the header). These duplicate receptions happen in many network scenarios such as the following.

- Broadcast flooding: During flooding, a node may receive a copy of the broadcast packet from each of its neighbors.
- Multicast streaming: A node may be in the range of more than one multicast transmitter/forwarder and hence receives multiple copies of a multicast packet.
- Unicast forwarding: A node in the vicinity of both the sender and the relay receives the packet once from the sender and again when the relay forwards the packet.
- Packet retransmission: A packet or its ACK is lost between a sender and a receiver. A neighboring node overhears the original transmission and the retransmission.

We posit that repeated packet transmissions allow a receiver to recover from collisions using interference cancellation.

Interference cancellation [8] is an approach in which a receiver extracts the signal of interest (SI) from a collision by modeling the interference and canceling it to obtain the SI. In the case of *unknown* interference (UI), i.e., when a receiver has no a priori knowledge of the interfering transmission, cancellation is possible only if the UI is significantly stronger than the SI. In that case, the receiver needs to decode the UI, cancel it, and then decode the SI. On the other hand, *known* interference (KI), i.e., an interfering transmission that is known in advance, may be cancelled regardless of whether KI is slightly or significantly weaker or stronger than the SI. We refer to this as *known interference cancellation* (KIC).

If a receiver already has one of the two packets involved in a collision, KIC enables it to recover the other packet. As explained before, there are many scenarios in which a node

overhears repeated transmissions of the same packet. If the receiver caches the packet upon its first transmission, then that receiver will be in a position to resolve a collision caused by the packet’s second transmission with any other packet. Suppose a node N overheard a packet p transmitted by the sender S . If the relay R forwards p and that collides with another packet q at N , then N can decode q using KIC. In other words, *when a KIC-enabled node overhears and caches a packet, it becomes effectively immune to the interference caused by the subsequent transmission of that packet.*

KIC is also applicable when two identical packets collide, which could happen in case of broadcast flooding. Employing a technique such as ZigZag decoding [7], the receiver can use the known (in the clear) bits to decode the unknown (interfered) bits of the packet iteratively. Though ZigZag was originally meant for extracting two packets from two separate collisions, the same technique can be applied to recover a packet from a collision between duplicate transmissions. Thus, KIC facilitates reliable broadcasts despite collisions, obviating the need for staggering broadcasts [11], [12]. Recently proposed Chorus [14] does precisely that of resolving collisions between identical broadcast transmissions limiting the novelty of KIC. However, KIC is a more generic approach with broader applicability to unicast and multicast scenarios too.

Another approach similar to KIC is Analog Network Coding (ANC), particularly in its application to a chain topology [9]. At a high level, one can view ANC, ZigZag, Chorus, and KIC as employing the same underlying interference cancellation technique but to different application scenarios (more on this in Section VI). Our contributions in this paper are as follows: 1) We observe that the reception of duplicate packets is the requisite condition for applicability of KIC. 2) We identify the scenarios in which KIC permits concurrent transmissions. 3) We demonstrate the feasibility of KIC by implementing it using USRP/GnuRadio testbed. 4) We evaluate the performance of KIC using QualNet [3] and show that KIC improves the throughput and the fairness of unicast and multicast.

Before delving into the details of KIC, we note that KIC caches bits not signals, and the repeated transmission can be at a different bit-rate. The rest of the paper is organized as follows. In the next section, we elaborate further on known interference cancellation. Section III lists the scenarios in which KIC offers performance improvements. The implementation of KIC on USRP/GnuRadio testbed is described in Section IV and it is evaluated in Section V. Section VI discusses the related work and Section VII concludes the paper.

II. KNOWN INTERFERENCE CANCELLATION

Traditionally, when one or more transmissions interfere with a signal of interest (SI) at a receiver, the SI can be decoded only if the SI is stronger than the interference and satisfies a certain signal-to-interference-and-noise (SINR) threshold. Recent advances [8] have shown that it is possible to model the interference, cancel it, and then extract the SI. With such an *interference cancellation* approach, a receiver can obtain the frame intended for it even when the SI is weaker than the interference and does not satisfy the required SINR threshold. While interference cancellation is expected to yield throughput improvements, the extent of the gains depend on the receiver's ability to accurately characterize the interference.

When a receiver has no a priori knowledge of the interference UI, cancellation is possible only if UI is significantly stronger than SI. Only then can the receiver decode UI, cancel it, and then decode SI. On the other hand, known interference (KI) can be cancelled by a receiver even if KI is only slightly stronger than SI. Moreover, even when KI is weaker than SI, canceling KI out will improve the SINR of SI, thereby increasing the feasible bitrate of SI. We refer to this as *known interference cancellation* (KIC). In the following, we contrast the cases of unknown and known interference cancellation in terms of a receiver's ability to decode its signal of interest.

Let S be the strength of SI at a receiver. Suppose B and N are bandwidth and noise of the channel. Without interference, the highest feasible bitrate for SI is $B \log_2(1 + \frac{S}{N})$. Suppose another transmission is causing interference I at the receiver. Without any interference cancellation, the receiver can decode SI if it is transmitted at a bitrate below $B \log_2(1 + \frac{S}{I+N})$.

With an ideal interference cancellation, the receiver can cancel I and decode SI at rate $B \log_2(1 + \frac{S}{N})$ as if there is no interference. This is possible if the interference is known, i.e., the receiver is informed of the interfering frame. On the other hand, if the receiver has no knowledge of the interfering frame, then such unknown interference has to be decoded first before canceling it. The receiver can do this only if the bitrate of the interfering transmission is below $B \log_2(1 + \frac{I}{S+N})$.

To summarize, unknown interference cancellation does not require any prior knowledge but the gains are limited. KIC, on the other hand, offers broader gains provided the receiver is already aware of the interfering frame. In the next section, we identify the scenarios that are inherently amenable for KIC.

III. APPLICATIONS OF KIC

KIC is possible only when the receiver already has a copy of the interfering frame. This may appear to be a stringent condition which is unlikely to happen in practice. However, this is common in wireless networks as nodes typically receive duplicate packets. In the following section, we list different scenarios and elaborate how they benefit from using KIC.

A. Multicast Streaming

Emerging wireless applications such as MobiTV [2] and electronic classrooms [5] are demanding link layer support for group communication. An access point (AP) is expected

to multicast content to all members of a group that subscribe to a common service. But compared to unicast, wireless multicast is more prone to hidden terminals. As the number of receivers increases, the likelihood of a collision at any one of the receivers also increases. Preventing a collision with a mechanism like RTS/CTS is not viable with multiple receivers. Retransmissions to recover from collisions are not attractive either, due to the ACK implosion problem. Therefore, it is desirable to make multicast streaming collision-proof.

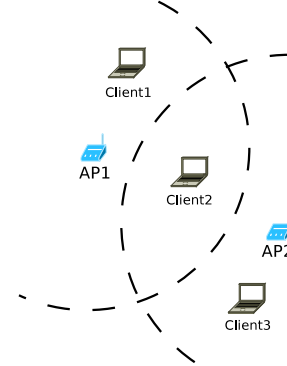


Fig. 1. AP1 and AP2 are hidden terminals w.r.t. Client2. Since the streams are identical, whenever there is a collision, the packets are identical or a client already has a copy of one of them. If AP1 and AP2 transmit the same packet simultaneously, Client2 can recover the packet using ZigZag-like decoding. Otherwise, it can decode a newer packet from the faster AP using the known older packet to cancel the interference from the slower AP.

Imagine a wireless network with 2 APs streaming the same content using multicast to 3 clients in a multicast group, as in Fig. 1. Since AP1 and AP2 are outside each other's carrier sense range, they are hidden terminals w.r.t. Client2. Whenever both AP1 and AP2 transmit simultaneously it results in a collision at Client2. Since both transmit the same stream, Client2 can still recover the packets from collisions using KIC as follows: i) The same packet is transmitted by both APs. This is similar to recovering from the broadcast collision discussed earlier using ZigZag-like decoding; ii) AP1 transmits a packet P_k while AP2 transmits P_{k+1} (Fig. 2a). In this case, it is likely that Client2 already received P_k from AP2 earlier. It can then use KIC to cancel P_k and decode P_{k+1} . iii) AP1 transmits a packet P_k while AP2 transmits P_{k-1} (Fig. 2b). This can also be resolved similar to the previous case but reversing the roles of APs. Thus, KIC can make multicasting more reliable without introducing additional control traffic.

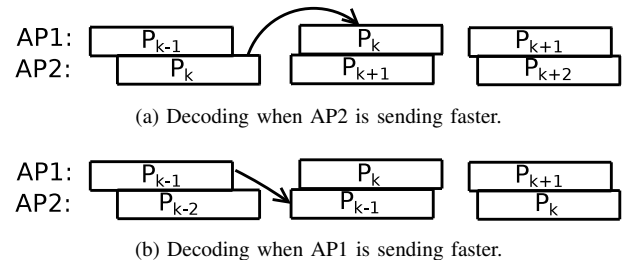


Fig. 2. Illustration of how a receiver resolves collisions of identical packet streams. Previously received packet is cancelled to extract the next packet.

B. Unicast Forwarding

In multi-hop wireless networks, such as wireless mesh networks, a unicast packet gets forwarded by each node along a path from the source to the destination. A forwarder essentially transmits the same packet it receives from the previous hop. Therefore, a node in the overlapping region of the previous and current hop may receive the same packet twice. This situation can be exploited by KIC to allow reception of another packet by a node in the overlapping region.

We illustrate this using an example in Figure 3. Suppose A is the source, C is the destination, and B is the forwarder between them. So, if A has a packet destined for C, it transmits the packet to B. This packet could be received by not only B but also a node in the dashed region such as G. When B forwards the packet to C, traditionally the same packet would be received again by G which is in the communication range of B. Instead, KIC permits concurrent reception of a packet at G from another node such as H. Thus, forwarding by B of a packet from A does not interfere with the reception by any node in the overlapping region between A and B.

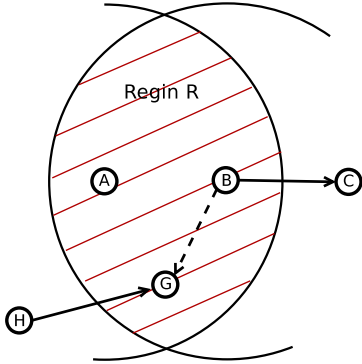


Fig. 3. Node G overhears a transmission from A to B. When B forwards the packet to C, node G can concurrently receive another packet from H using KIC. Any node in the dashed region that received A's original transmission is immune to the interference caused by B forwarding the packet.

A special case of this scenario is forwarding along a chain which was addressed earlier by Analog Network Coding [9]. Consider a chain $A \rightarrow B \rightarrow C \rightarrow D$, where only one pair of these nodes could be transmitting and receiving at a time, otherwise there would be a collision. KIC can address this *self-interference* problem. With KIC, forwarding of a packet, p_1 , from C to D can be concurrent with the reception of the next packet, p_2 , from A to B. Since B already has a copy of p_1 , using KIC it can extract p_2 from the resulting collision. In other words, with KIC as with ANC [9], every node along the chain could instantaneously be either a sender or receiver.

C. Packet Retransmission

Wireless networks employ link layer retransmissions for reliable delivery of unicast packets. A sender expects an immediate ACK after completing its DATA transmission. If a portion of the DATA transmission is lost or the ACK is not received correctly, the sender retransmits the DATA. It is possible that a nearby node overhears the DATA frame twice.

Without KIC, the neighboring node can not receive another packet during these overheard transmissions since that would result in a collision. With KIC, the neighboring node can concurrently receive another packet during the retransmission.

Consider the scenario shown in Figure 4 where A transmits a packet to B unsuccessfully. Suppose another nearby node, G, overheard this packet correctly. This is not unusual due to varying channel conditions around each node. During A's retransmission, G can receive another packet from H because it already has A's packet. Effectively, retransmissions by A will not interfere with receptions at any node that overheard A's original transmission, within the communication range of A. Put succinctly, *the cost of retransmissions can be minimized using KIC*. Retransmissions are frequent in wireless networks, therefore KIC can provide substantial throughput gains.

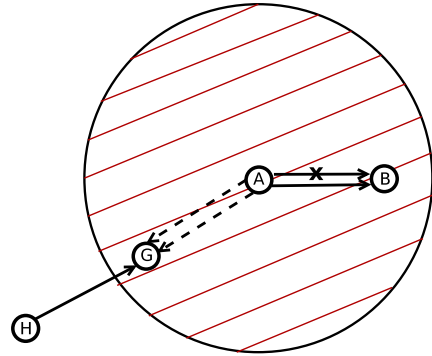


Fig. 4. Suppose A transmitted a packet which is lost at the intended receiver B but overheard correctly by G. Now G can receive A's retransmission of that packet concurrently with a packet from H using KIC. Effectively, any node in the dashed region that received A's first transmission is immune to the interference caused by its retransmission.

The above examples illustrate that there are many wireless network scenarios in which KIC can resolve collisions. However, there are two challenges in realizing KIC. First, the receiver should be able to detect that it already received one of the colliding packets. Second, it has to model the corresponding signal and cancel it. The following section discusses these challenges and ways to address them.

IV. KNOWN PACKET DETECTION AND CANCELLATION

One of the hurdles in employing KIC is identifying that a copy of one of the packets involved in the collision is already cached at the receiver. The key issue is that the packet being transmitted and its copy received earlier may not be identical but differ in some header fields and consequently CRC. In case of a packet retransmission, all header fields would be the same except for just one bit in the Frame Control field. With link layer multicasting, fields like group id or in-group sequence number will be identical among copies whereas the sender address would be different. Under unicast forwarding, most of the fields of the forwarded copy may be different from the original packet. Therefore, KIC requires that header fields and additionally some payload bits are in the clear to perform matching of colliding packets with the cached copies.

To facilitate the extraction of header fields, we adopt the approach of adding a trailer at the end of each packet. This trailer has the same content as the header, but in the opposite order. When one end of a packet is in collision, it is likely that the other end of this packet is in the clear. In order to decode the trailer without decoding the whole packet, a postamble is also added at the end of each packet, so the signal can be synchronized and decoded backwards.

A KIC-enabled receiver first tries to decode the arriving signal using the normal process. But if the CRC fails, it checks whether there is a collision by comparing the header and the trailer of the arriving signal. If they are not same, it is likely that they belong to two different colliding packets (while this approach misses out the collisions where a shorter packet completely overlaps with a longer packet, they are anyway not amenable for interference modelling and cancellation). Then, the receiver checks if the payload of one of the colliding packets is in its cache. It compares the payload bits in the clear at the head (tail) with the prefix (suffix) of cached payloads. If there is a match, that payload is wrapped with the header and the trailer. Effectively the receiver extracts header/trailer from the arriving signal and the payload from the cache to construct the known packet involved in the collision.

The next step is to generate the signal corresponding to the bits of the known collided packet. The preamble/postamble and the bits in the clear aid in modelling the wireless channel. We follow the same approach as the one in [8]. Instead of individually considering all the channel effects such as fading, frequency offset, and inter-symbol interference, we model them by a single function $R(t)$. This $R(t)$ is built for every combination of three consecutive symbols. For each combination, we average it over all its occurrences in the clear. Then the signal corresponding to the known bits is generated by applying $R(t)$. This signal is then subtracted from the arriving signal and the resulting signal is passed through normal decoding process. In case the number of payload bits in the clear are few and the match is incorrect, the CRC of the decoded packet would fail and the packet is discarded. While this is undesirable, it is no worse than the alternative since the packet would have been lost anyway without KIC. In most cases, sufficient bits would arrive in the clear and KIC would successfully resolve a collision with a known packet.

V. EVALUATION

We now validate the above theoretical ideas with experiments and simulations. First, we demonstrate the feasibility of KIC by implementing it on USRPs [4] and then evaluate its performance through simulations using QualNet [3].

A. USRP based Experiments

1) *Experimental methodology*: The experimental setting includes two USRPs as senders and one USRP in between as the receiver. The receiver is based on the ZigBee interference cancelling decoder as in [8] tuned to the channel centered at 2.43 GHz. We also use Gaussian Minimum Shift Keying (GMSK) as modulation and Frequency Shift Keying (FSK) as

the demodulation scheme. We manually created collisions by letting the senders transmit packets with pre-determined random payload periodically without carrier-sensing. The spacing between packets of each sender are controlled to ensure that their packets overlap sometimes and also that no more than two packets overlap at anytime. As discussed earlier, each packet is sent with the trailer in addition to the header.

Our experiments focus on the packet reception evaluation based on the offline decoding of the collided packets at the receiver. The arriving signals at the receiver are first filtered to eliminate instances when there is no collision. Then each collision instance (a total of 1600 collision instances in our experiments) is fed to the KIC decoder along with one of the colliding packets. The result is verified to see whether KIC decodes the other packet successfully. For comparison, the following decoders are also implemented at the receiver.

Resynchronizing (ReSync) decoder: It synchronizes based on the preamble, and decodes from the beginning. But it keeps looking for another stronger preamble, and resynchronizes as soon as it appears. We use the resynchronizing decoder to reflect the 802.11 receivers with the capture ability [10].

Unknown interference cancellation (UIC) decoder: This decoder is similar to the one implemented in [8]. It first decodes the stronger packet in the collision, then models and subtracts it from the signal, and finally decodes the other packet. Different from the KIC decoder, the stronger packet is always decoded first by the UIC decoder.

2) *Experimental Results*: We place two senders at equal distance from the receiver. Then slowly move one of the senders towards the receiver to create scenarios with a range of SINR values ($SINR_{unknown} - SINR_{known}$). In each position, we assume that the packets sent from each of the senders are known in turn by the receiver. As mentioned above, the results presented here correspond to 1600 collision instances where packets overlapped. We first measure the fraction of the bits of the packet in error as a function of the overlapping fraction of the packet. Figure 5 shows that KIC can decode the packet even when the extent of overlap is quite high.

We then compare the decoding performance of KIC with others as a function of SINR. Figure 6a shows the percentage of bits in error in the decoded unknown packet for each

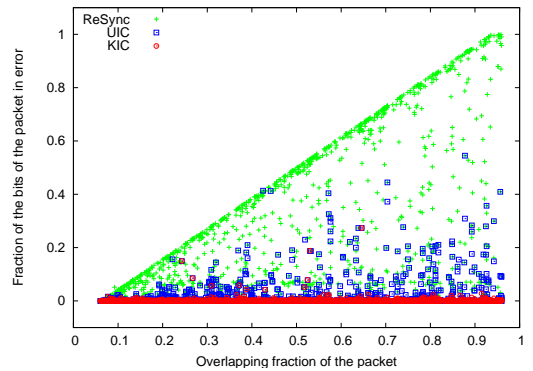


Fig. 5. KIC can decode the packet even with significant overlap.

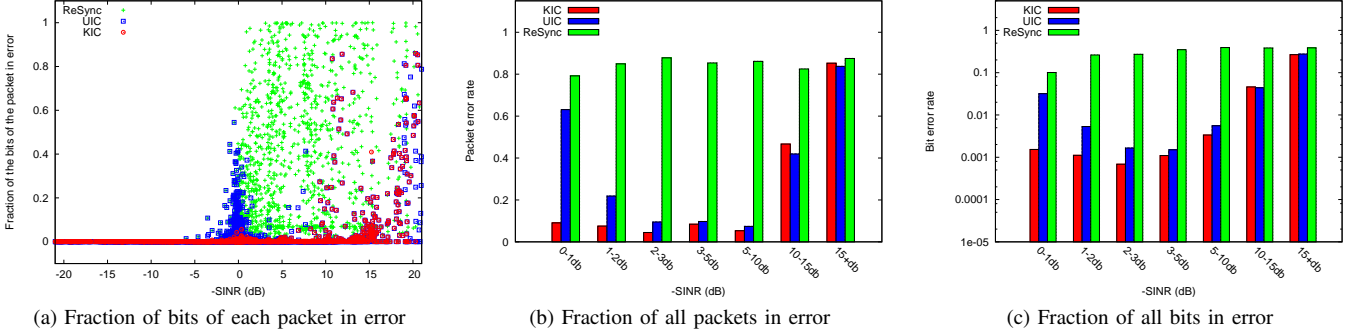


Fig. 6. Comparison of the performance of KIC with UIC and ReSync decoders under varying SINR ($SNR_{unknown} - SNR_{known}$) values (note that the x-axis of these graphs is $-SINR$). Only KIC works well when the signal strengths of known and unknown packets are similar ($-2dB < SINR < 2dB$).

collision. Figure 6b shows the packet error rate in different SINR regions, with the results of the same experiment, while Figure 6c shows the bit error rate. From these graphs, we can have the following observations:

All the decoders work well when $SINR > 3dB$. Every ZigBee symbol is encoded into 32 chips during the transmission. So the symbol can be decoded successfully even when several samples are in error. That is why ReSync decoder also can decode successfully without cancellation when the unknown packet in the collision is slightly stronger than the known one.

Both UIC and KIC are fine when $-10dB < SINR < -2dB$. In such a case, UIC can decode the stronger (which is known in case of KIC) packet and cancel it to decode the other packet. Therefore UIC and KIC behave similarly in this case.

Only KIC works when $-2dB < SINR < 2dB$. Although Zigbee has high error tolerance, when the signal strength of two packets are almost equal, the reception of the stronger packet still fails. This effect is observed in our experiments when the absolute value of SINR is less than 2dB. Since the stronger packet can not be decoded correctly, both ReSync decoder and UIC decoder can not be effective. But the KIC decoding does not depend on the decoding of the stronger packet, and hence works well even in this SINR range.

All decoders fail when SINR is too low. When $SINR < -10dB$, errors start to appear. Almost all packets fail when $SINR < -15dB$. When the known packet is too strong, any small modelling error causes enough damage to the remaining weaker signal of unknown packet resulting in a high error rate.

Note that these observations are based on a ZigBee decoder which has a high error tolerance. It can decode a packet successfully even for SINR as low as 3dB which is not the case with 802.11 [10]. We chose ZigBee for our USRP/GnuRadio experiments because of its ready availability. We expect the improvement with KIC over UIC to be higher for 802.11.

B. QualNet based Simulations

We also evaluated KIC using Qualnet simulator. We setup the MAC layer using 802.11b with 2Mbps data rate. When a collision includes a previously cached packet or two identical packets, we suppose KIC is used to decode the colliding signal. Based on our experimental results and ZigZag's experimental

results under 802.11 [7], we model whether KIC is successful or not based on the SNRs of the colliding packets as follows. If $SINR > 10dB$, capture takes effect and the loss rate of the unknown packet is the same as that using normal decoding. If $10dB > SINR > -10dB$, the unknown packet can be decoded with KIC, and we use a fixed loss rate of 5%, obtained from the previous experiments. When $-10dB > SINR > -15dB$, the loss rate linearly increases to 1. Finally, if $SINR < -15dB$, decoding of the unknown packet always fails.

1) *Unicast flows:* To examine the performance of KIC on a larger network, we tested it on a randomly generated 40-node network with randomly generated traffic. In these scenarios, KIC benefits from both retransmitting and forwarding, as discussed earlier. Figure 7a shows that KIC reduces the number of retransmissions and consequently the throughput. Figure 7b shows that the overall throughput is improved by KIC in scenarios with more than 10 flows, with a maximum gain of 24%. Figure 7c indicates that the lower throughput flows that suffer contention can benefit more from KIC. In another words, KIC can improve the fairness among flows.

2) *Multicast scenario:* Figure 8 shows the performance of clients in a simple multicast scenario, in which a series of receivers are evenly distributed between two wireless multicast forwarders. The multicast forwarders belong to the same multicast group and they are receiving traffic from the same multicast source through wired links. After applying KIC, the receptions of all clients improve, especially those clients in the middle of two senders, which otherwise receive very little traffic due to collisions.

VI. RELATED WORK

Interference cancellation has become prominent in wireless networking research recently due to the advances in software defined radio (SDR) [1]. In [8], a Successive Interference Cancellation (SIC) decoder is built to decode the collisions caused by hidden terminals. This is similar to what we termed UIC in this paper. As mentioned earlier, UIC works well only when the interference is much stronger than the signal of interest which restricts its applicability compared to KIC.

Self-interference cancellation [6], [13] has been proposed as a technique allowing a node to cancel its own transmitted signal and hence to successfully receive data while transmitting

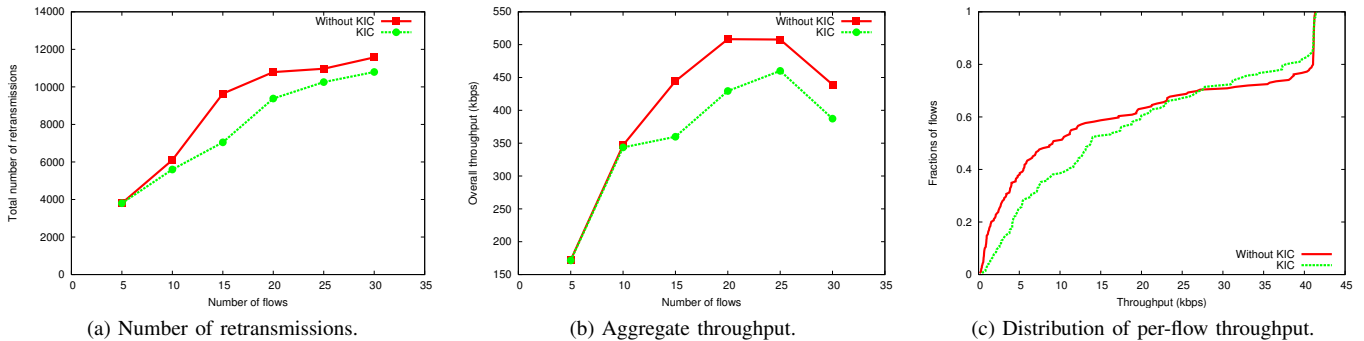


Fig. 7. Performance evaluation of KIC with unicast flows. KIC reduces retransmissions and improves throughput by up to 24%.

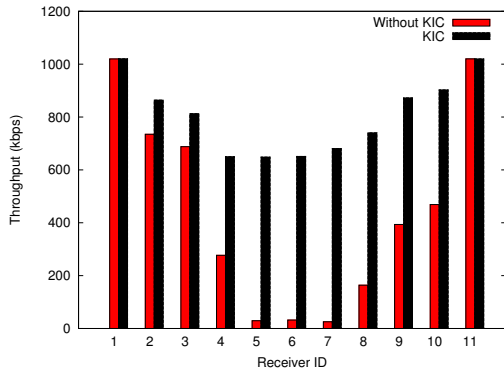


Fig. 8. Throughput of clients in between two multicast senders. KIC improves the multicast throughput particularly for the middle clients.

on the same channel. Unlike KIC, self-interference cancellation requires more than one antenna at the sender/receiver. Also, the self-signal is usually much stronger than the other signal, making the decoding after cancellation difficult.

ZigZag [7] iteratively uses the different clearly transmitted pieces among two consecutive collisions to decode two packets. Similar to ZigZag, KIC also focuses on duplicate transmissions. ZigZag applies when the first copy of two duplicate packets (a transmission and its retransmission) collides, and the second copy also collides with the same packet. KIC applies when the first copy of two duplicate transmissions is in the clear, and the second one is involved in a collision.

Recently, Zhang, et al proposed Chorus for wireless broadcasting [14]. Chorus can be considered an application of ZigZag to identical packets. We argue that KIC has broader applicability since it applies to any packet that has been received more than once at a node, regardless of whether these copies (of the same packet) are arriving at the same time.

Finally, ANC [9] is most relevant to our work. ANC discusses two scenarios: packet exchange and chain topology. ANC can exchange two packets in two time slots by relaying the signals instead of bits. It also can facilitate concurrent transmissions as in chain topology like KIC. At a high level, one can view ANC, ZigZag, Chorus, and KIC as employing the same underlying interference cancellation technique but to different application scenarios. KIC by caching bits of any

arbitrary packet is able to gain in many different scenarios from unicast to multicast and broadcast.

VII. CONCLUSIONS

This paper argues that duplicate receptions are fairly common in wireless networks and proceeds to exploit them with known interference cancellation (KIC). The key benefit of KIC is that a receiver that overheard a packet once becomes immune to the interference caused by the subsequent transmission of that packet. We presented the scenarios suitable for KIC, discussed how KIC makes multicast more reliable, and unicast more efficient. We demonstrated the feasibility of KIC by implementing and evaluating it on a testbed of 3 USRP/GnuRadio nodes. We have also simulated KIC using QualNet and shown that it improves throughput even with conventional MAC. We believe additional gains are possible if MAC is made aware of KIC so that it can permit more concurrent transmissions, which we plan to study further.

REFERENCES

- [1] Gnu radio project. <http://gnuradio.org/trac>.
- [2] Mobitv. Mobile television and radio service provider.
- [3] Qualnet Network Simulator. <http://www.scalable-networks.com/>.
- [4] Universal Software Radio Peripheral. <http://www.ettus.com>.
- [5] ANDERSON, R., CHUNG, O., DAVIS, K., PRINCE, C., RAZMOV, V., AND SIMON, B. Classroom presenter: a classroom interaction system for active and collaborative learning. In *Proc. IEEE WIPTE* (2006).
- [6] BLISS, D. W., AND ET AL. Simultaneous transmission and reception for improved wireless network performance. In *Proceedings of the IEEE/SP 14th Workshop on Statistical Signal Processing* (2007).
- [7] GOLLAKOTA, S., AND KATABI, D. Zigzag decoding: Combating hidden terminal in wireless networks. In *Proc. ACM Sigcomm* (2008).
- [8] HALPERIN, D., ANDERSON, T., AND WETHERALL, D. Taking the sting out of carrier sense: Interference cancellation for wireless lans. In *Proc. ACM Mobicom* (2008).
- [9] KATTI, S., GOLLAKOTA, S., AND KATABI, D. Embracing wireless interference: Analog network coding. In *Proc. ACM Sigcomm* (2007).
- [10] LEE, J., AND ET AL. An experimental study on the capture effect in 802.11a networks. In *WinTECH* (2007).
- [11] LOU, W., AND WU, J. On reducing broadcast redundancy in ad hoc wireless networks. *IEEE Transactions on Mobile Computing* 1, 2 (2002).
- [12] LOU, W., AND WU, J. Toward broadcast reliability in mobile ad hoc networks with double coverage. *IEEE Transactions on Mobile Computing* 6, 2 (2007).
- [13] RADUNOVIC, B., GUNAWARDENA, D., PROUTIERE, A., SINGH, N., BALAN, V., AND KEY, P. Efficiency and fairness in distributed wireless networks through self-interference cancellation and scheduling. Tech. Rep. MSR-TR-2009-27, Microsoft Research, Cambridge, Mar. 2009.
- [14] ZHANG, X., AND SHIN, K. G. Chorus: Collision resolution for efficient wireless broadcast. In *Proc. IEEE Infocom* (2010).