On Eliminating the Exposed Terminal Problem Using Signature Detection

Junmei Yao, Tao Xiong, Jin Zhang and Wei Lou
Department of Computing, The Hong Kong Polytechnic University, Hong Kong
{csjyao, cstxiong, csjzhang, csweilou}@comp.polyu.edu.hk

Abstract—Wireless networks are propelled to improve the network throughput effectively to face the challenge of sustaining the rapid growth of data traffic and the high density of wireless nodes. Exposed terminals are a main source in wireless networks that degrades the network throughput performance through excessively avoiding interferences and forbidding concurrencies. To combat the exposed terminal problem and exploit the concurrent transmissions in wireless networks, we present the design of Interference Resistant Multiple Access (IRMA) in this paper, which can achieve higher throughput compared to the 802.11 standard. By observing that nodes in current protocols waste transmission opportunities in two different scenarios, IRMA exploits the concurrency in two aspects. IRMA proposes a signature detection method in the physical layer to combat control frames' collisions, thus exploits the concurrency at the transmitter side. IRMA also designs a new NAV update scheme in the MAC layer to differentiate the interference ranges of different transmission links, thus exploits the concurrency of all non-interfering links. Experimental results based on USRP2 demonstrate the feasibility of the signature detection method, and simulations based on ns-2 show that IRMA outperforms the 802.11 standard and other protocols significantly.

Index Terms—Keywords: Wireless Networks; Exposed Terminals; Cross Layer Design; Signal Correlation.

1 Introduction

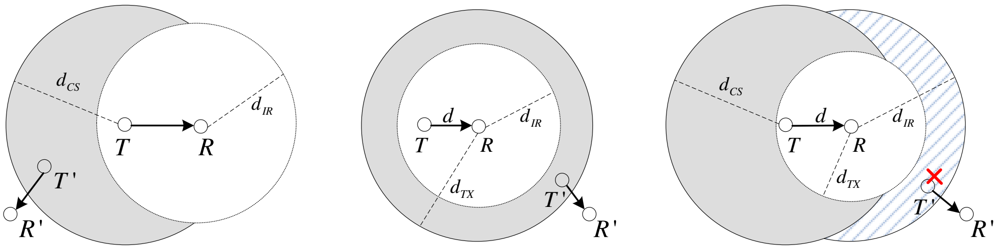
Nowadays, most wireless local area networks are organized with the 802.11 standard [1], in which nodes use the carrier sense multiple access (CSMA) to avoid collisions: Before transmitting, a sender senses the medium to determine if a nearby node is transmitting. If the channel is determined idle, the node proceeds the transmission; otherwise, the sender will defer until the end of the ongoing transmission. The CSMA can also be performed through a virtual mechanism. The virtual carrier sense uses the exchange of RTS and CTS frames to reserve the medium for the actual data transmission. The RTS and CTS frames contain a NAV field that defines the period of time that the medium is to be reserved for the transmission of the actual data frame and returned ACK. All nodes that receive the RTS or CTS frames will keep silence during the NAV time to avoid collisions. However, CSMA introduces the exposed terminal problem that causes performance degradation, as senders are prevented from transmitting data frames concurrently even when their transmissions have no mutual interferences [2].

This problem has attracted much attention [2]–[7] as wireless networks should improve the performance effectively to face the challenge of sustaining the rapid growth of data traffic and the high density of wireless nodes. Although the transmission rate in the physical layer has been increased through advancing new technologies [1], [8], the wasted transmission opportunities or collisions induced by CSMA impede their effectiveness remarkably.

Our observations on the 802.11 standard reveal that nodes waste transmission opportunities in two scenarios. The first scenario occurs due to the collision avoidance of control frames at the transmitter side. As shown in Fig. 1(a), although the data frame collision only exists at the receiver R, nodes that are in the carrier sense range d_{CS} of the transmitter T should also be prohibited to transmit concurrently, in order to avoid the ACK frame collision at T. Nodes in the grey area waste their transmission opportunities, as their concurrent transmissions will not interfere with R's reception of ongoing data. Specially, when there are only two links $T \rightarrow R$ and $T' \rightarrow R'$ in the system, as shown in Fig. 1(a), CSMA will lead to 50% performance degradation due to the forbidden of their concurrent transmissions.

The second scenario occurs due to no differentiation between interfering and non-interfering links. In wireless communications, a collision (or interference) event indicates the transmitted signal cannot be detected successfully, that means, the received Signal to Interference plus Noise Ratio (SINR) is below a threshold [9]. As indicated in [10], the interference range d_{IR} of the receiver is proportional to the transmitter-receiver distance d of the ongoing transmission link. As shown in Fig. 1(b), if d is fairly small, d_{IR} can be shorter than the transmission range d_{TX} [10]. However, under the 802.11 standard, nodes in the grey area are prohibited to transmit concurrently although they are more than d_{IR} from the receiver and will not interfere with the ongoing transmission link $T \rightarrow R$. Nodes waste transmission opportunities as the medium around the receiver has been simply reserved through the CTS frame. Specially, in a two link scenario shown in Fig. 1(b), CSMA also causes 50% performance degradation through forbidding the concurrency of $T \rightarrow R$ and $T' \rightarrow R'$.

Our observations also reveal that nodes may induce



(a) Nodes around the transmitter T are prohibited to transmit data.

(b) Nodes out of the interference range of the receiver R are prohibited to transmit data.

(c) Nodes within the interference range of R and out of carrier sense range of T are permitted to transmit data.

Fig. 1. Two scenarios that nodes waste transmission opportunities and one scenario that nodes induce collisions.

unnecessary collisions due to no differentiation between interfering and non-interfering links. As shown in Fig. 1(c), if d is relatively large, d_{IR} will be larger than the transmission range d_{TX}. However, under the 802.11 standard, nodes in the slash area are permitted to transmit packets although they are less than d_{IR} from the receiver and will definitely interfere with the ongoing transmission link $T \rightarrow R$. Nodes induce collisions as they cannot receive the CTS frame correctly. Ref. [10] intends to solve this problem through reducing the effective transmission range so as to make d_{IR} smaller than the real transmission range d_{TX}. However, this mechanism will obviously prohibit more concurrent transmissions and degrade the network throughput.

Recent studies that address the exposed terminal problem fall into solving one or two of the above scenarios. SDN [3] exploits non-interfering links that have no interferences at both the transmitter and receiver sides for concurrent transmissions. In SDN, each node constructs an interference graph by periodically exchanging power-exchange packets with nearby nodes. The node may transmit its own frame if there is no interference between its transmission link and any of current transmission links. However, SDN cannot determine interfering links effectively when d_{IR} is larger than d_{TX}. Meanwhile, it does not exploit transmission opportunities at the transmitter side.

CMAP [2] considers two scenarios in Fig. 1(a) and Fig. 1(b). It builds a conflict map for each node using empirical observations of packet loss and uses the map to differentiate between interfering and non-interfering links. By listening to the ongoing transmissions and consulting the map, nodes can decide whether to transmit data immediately or not. CMAP also exploits transmission opportunities around the transmitter side, and tries to mitigate the ACK collision at the transmitter through a window-sized ACK and retransmission protocol. However, in the scenario of Fig. 1(a), concurrent transmissions have a high ACK loss rate, which causes many redundant retransmissions and degrades the network throughput. Meanwhile, this protocol still cannot prevent collisions in the scenario of Fig. 1(c).

In this paper, we propose Interference Resistant Multiple Access (IRMA), a novel cross layer protocol, to improve the throughput of wireless networks. Our protocol can exploit the concurrency in both scenarios of Fig. 1(a) and Fig. 1(b). IRMA combats the control frame collision at the transmitter side by using a signature detection method, in which nodes use known symbol sequences, called as *signatures*, to convey information. When transmitting a control frame, nodes need to map the control information to dedicated signatures, attach the signatures to the frame at the physical layer and send the frame out. When receiving a control frame, nodes discern the signatures from the incoming signals and convert the signatures to the original control information. As signatures can be discerned in the presence of strong interferences, IRMA can exploit concurrent transmissions by using signature detection method to tolerate control frame collisions.

IRMA also exploits concurrent transmissions through differentiating between the interfering and non-interfering links in an easy way. IRMA allows the receiver to use the CTS frame to reserve the medium for the transmitter's data transmission for the NAV time. Only the nodes in the interference range of the receiver will update the NAV state for keeping silence. IRMA further uses a new channel access scheme for nodes to determine whether to initiate a transmission or not when they intend to send a data frame. Note that IRMA can differentiate between interfering and non-interfering links even when the interference range is larger than the transmission range, as the new designed CTS frame can be detected correctly in very low SINR environments. Thus, it can avoid collisions effectively in the scenario of Fig. 1(c).

The paper makes the following key contributions:

- IRMA is the first protocol that can fully exploit concurrent transmissions in the two scenarios shown in Fig. 1(a) and Fig. 1(b), and avoid collisions in the scenario shown in Fig. 1(c). IRMA exploits concurrency at the transmitter side as collided control frames can be detected correctly using the signature detection method. It also exploits concurrency in the network through differentiating between interfering and non-interfering links according to the ongoing transmission link, no matter the interference range is larger or smaller than the transmission range.
- IRMA discusses the scenario when nodes in the network support multiple transmission rates to improve the bandwidth efficiency. On the contrary, other protocols such as [2], [3] do not consider this situation.
- The paper quantifies the signature detection method

- through hardware experiments. The results demonstrate the feasibility of our signature design as the control frames' signatures can be detected correctly in the presence of strong interferences.
- The paper demonstrates IRMA's significant throughput improvement through simulations. The results show that IRMA can outperform both 802.11 standard protocols under different network topologies and different transmission rates.

The rest of this paper is organized as follows. Section 2 briefly introduces the background knowledge of the signature detection method. Section 3 gives an overview of the IRMA architecture and mechanism. Section 4 describes the design of IRMA in detail. Section 5 demonstrates the feasibility of the signature detection through hardware experiments. Section 6 evaluates the performance improvement of IRMA through simulations. Section 7 gives the related work. Section 8 concludes this paper and puts future work.

2 Preliminary

In this section, we will give some background of cross correlation used in the signature detection process.

The signature detection process in our protocol is to discern a known signature from the incoming signals, and then recover the original information. We propose to accomplish the process of discerning the signature by exploiting the cross correlation, which is commonly used for searching for a known feature in a long duration signal [9]. Cross correlation has already been used in preamble synchronization, which is to accomplish the physical carrier sense mechanism in the 802.11 standard [1]. It also has some applications in recent works such as [5], [11]–[15]. In the signature detection process, the cross correlation will be conducted between the incoming signal and the known signature. A spike may appear if the signature is presented in the incoming signal. Therefore, nodes can determine the presence or absence of a signature in the received signal through this cross correlation operation.

A wireless signal is typically described as a stream of complex numbers, and the bit sequence of the signal should be mapped into a series of complex symbols in the digital modulation process. At the receiver, after RF down-converter and sampler in the demodulation process, the signal is represented as a series of complex samples, which may differ from the transmitted symbol sequence in amplitude, phase, and frequency. Suppose $x[n]$ is the complex number that represents the n^{th} transmitted symbol, the corresponding received sample $y[n]$ can be denoted as:

$$y[n] = Hx[n]e^{j2\pi n\delta_f T} + w[n],$$

where H is the channel attenuation factor, δ_f is the frequency difference between the sender and receiver, T is the sampling period, and $w[n]$ is the background noise, which contains the thermal noise and interferences from other concurrent transmissions.

Suppose two nodes $S1$ and $S2$ transmit signals simultaneously, and are both received at node R. Suppose the transmitted symbol is $x_1[n]$ in $S1$, and $x_2[n]$ in $S2$, then the received signal at R is

$$y[n] = H_1x_1[n]e^{j2\pi n\delta_{f1}T} + H_2x_2[n]e^{j2\pi n\delta_{f2}T} + w[n].$$

As δ_f can be estimated based on the history information, according to which we can compensate the frequency offset of the received signal, we will omit it in the following equations.

Let the samples $s[k]$, $1 \leq k \leq L$, refer to a known sequence, and $\overline{s[k]}$ be the complex conjugate of $s[k]$. We can define the cross correlation of signals s and y at the position Δ as:

$$R_x(\Delta) = \sum_{k=1}^{L} \overline{s[k]}(H_1x_1[k+\Delta] + H_2x_2[k+\Delta] + w[k+\Delta]).$$

When the transmitted signal x_2 from $S2$ matches s at the position Δ', we can get:

$$R_x(\Delta') = \sum_{k=1}^{L} \overline{s[k]}(H_2x_2[k+\Delta']) = H\sum_{k=1}^{L}|s[k]|^2.$$

The correlation value $R_x(\Delta')$ is the sum of energy of this segment of signal, and it reaches a peak value if the known sequence appears in the received signal. If not, $R_x(\Delta')$ would be close to zero as the received signal is independent of the known sequence. The value $R_x(\Delta')$ can be normalized by the signal strength of s, as $R_N(\Delta') = \frac{R_x(\Delta')}{H\sum_{k=1}^{L}|s[k]|^2}$ [13]. Practically, the value $R_N(\Delta')$ is compared with a constant threshold β_{Corr} to detect the known sequence: If $R_N(\Delta')$ is above β_{Corr}, the known sequence is detected in the received signal at position Δ'.

Different thresholds may lead to different signature detection results. As shown in Fig. 2, a higher threshold $\beta_{Corr}(high)$ increases the probability of a false negative error (a signature that is in the received signal is missed), and a lower threshold $\beta_{Corr}(low)$ increases the probability of a false positive error (a signature that is not in the received signal is mistakenly detected). We should mitigate both errors in the signature detection design.

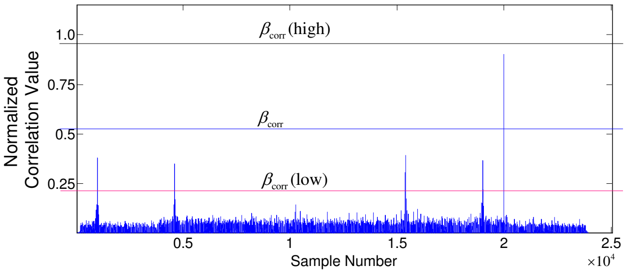


Fig. 2. Correlation threshold β_{Corr}.

3 Overview of IRMA

In this section, we first introduce the architecture of IRMA protocol and overview the IRMA mechanism through an example for ease of understanding, then discuss the scenario when multiple transmission rates can be supported in the

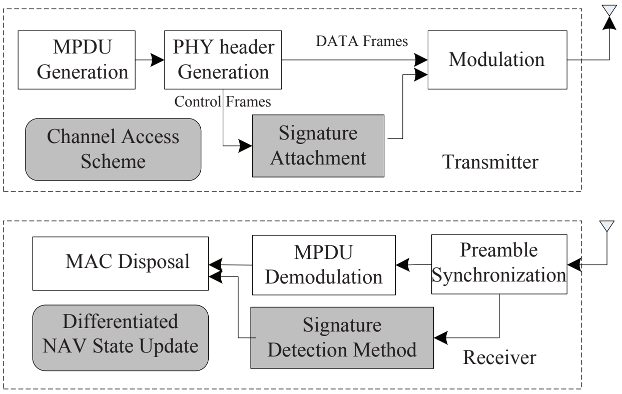


Fig. 3. System architecture of IRMA. Comparing with the 802.11 standard, IRMA needs additional grey blocks to accomplish the protocol.

network. Based on that, we summarize the key information that should be detected using signature detection method (SDM) in the case of collisions for the IRMA protocol design.

3.1 Architecture of IRMA

Fig. 3 briefly illustrates the architecture of IRMA. Compared with the 802.11 standard, IRMA needs new blocks to accomplish the protocol.

Under the 802.11 standard, when a transmitter begins to transmit a frame, it first generates a MAC protocol data unit (MPDU), then adds a physical layer header, finally transmits the frame out after modulation. For a control frame to be transmitted, IRMA will attach signatures, which represent specific control information, to the standard frame before modulation. A new channel access scheme in the MAC layer is designed to achieve more concurrent transmissions and avoid data collisions.

In the receiving process, after completing the preamble synchronization, the receiver begins to demodulate the samples in the MPDU field. At this time, IRMA lets the receiver perform the SDM to discern the signatures at specified positions of the incoming samples after the preamble, then recover the original information. The outputs of MPDU demodulation and signature detection are both used for the MAC disposal. A new differentiated NAV state update scheme in the MAC layer is designed to distinguish the interfering and non-interfering links around a node, while the results can assist the channel access scheme to make proper decisions.

3.2 Overview of IRMA Behavior

IRMA uses RTS/CTS/DATA/ACK four-way handshake mechanism for the medium access and data transmission. Physical carrier sense is disabled by IRMA as the interfered control frames at the transmitter side can be detected correctly using SDM. Moreover, all the new components in Fig. 3 work collaboratively to increase concurrent transmissions.

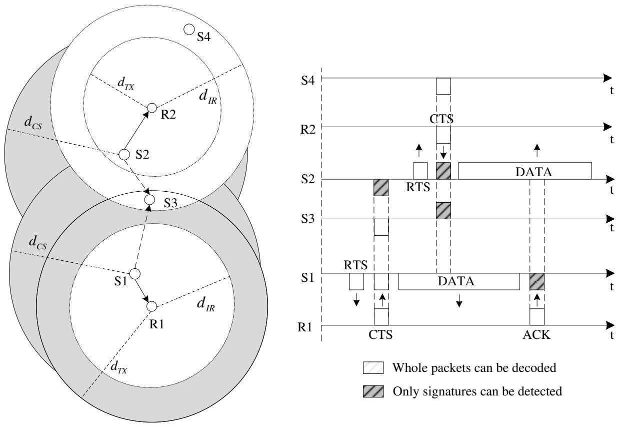


Fig. 4. A scenario of IRMA, where $S1$ and $S2$ are exposed terminals when they transmit data to $R1$ and $R2$ respectively. IRMA permits their concurrent transmissions. Note that $S3$ is out of the interference range d_{IR} of $R1$, but in the interference range d_{IR} of $R2$.

For ease of understanding, we illustrate the IRMA mechanism in a typical scenario with the time sequence diagram of each node, as shown in Fig. 4: $S1$ and $S2$ are exposed terminals as they intend to transmit data to $R1$ and $R2$, respectively. Their concurrent transmissions are permitted by IRMA. $S3$ can interfere with $R2$'s data reception, but it cannot interfere with $R1$'s reception. $S4$ may also interfere with $R2$'s data reception. Comparing to the 802.11 standard, IRMA works differently, which is illustrated as follows:

- If $S1$ wants to transmit data to $R1$, the transmission can be permitted by the channel access scheme as $S1$'s NAV state is zero. It then sends a RTS frame after a backoff time to initiate the transmission, and begins to transmit data frame after receiving the CTS feedback from $R1$ successfully. Although $S3$ (which acts as T' in Fig. 1(b)) is in the transmission range d_{TX} of $R1$ and can receive this CTS message, it should not update its NAV state, as it is outside the interference range d_{IR} of $R1$. Note that although $S2$ is out of d_{TX} of $R1$, it can also detect this CTS message correctly using SDM. It should not update its NAV state as it is also outside d_{IR} of $R1$.
- During the data transmission from $S1$ to $R1$, although $S2$ (which acts as T' in Fig. 1(a)) is in the carrier sense range d_{CS} of $S1$ and determines the channel to be busy, it can be permitted to transmit data to $R2$ by the channel access scheme as its NAV state is zero. $S2$ then sends a RTS frame after a backoff time to initiate its transmission. The corresponding CTS feedback from $R2$ will be interfered at $S2$ by the data transmission from $S1$. However, this interfered CTS can be detected correctly using SDM. Therefore, $S2$ can continue the data transmission to $R2$ accordingly. Note that $S3$ and $S4$ (both act as T' in Fig. 1(c) here) can also detect the CTS frame correctly using SDM and get the NAV information, although they are out of the transmission range of $R2$. They should update

their NAV states to keep silence, as they are both in the interference range d_{IR} of $R2$.

- After finishing the reception of the data frame, $R1$ will reply an ACK to $S1$. The ACK frame would be interfered at $S1$ by the data transmission from $S2$. This interfered ACK can be detected by $S1$ using SDM, completing the transmission successfully.

From this scenario, we can see that three key control information should be detected correctly from the CTS/ACK frames even when they are under collisions. The first one is the receiver address (RA) of the CTS/ACK frames. The nodes (such as $S1$ and $S2$ in Fig. 4) should be able to obtain the RA information in the CTS frame to check if it is the target of this frame even when the CTS frame is interfered by other transmissions. Meanwhile, the RA information in the ACK frame notifies the node that its data frame has been received correctly. The second one is the NAV information in the CTS frame, with which nodes that do not involve in the RTS/CTS handshake (such as $S3$ in Fig. 4) can update their NAV state to keep silence. The third one is the interference range of the ongoing transmission link, which should be obtained by the nearby nodes as an input to the channel access scheme so that they can make a proper decision on accessing the channel concurrently while avoiding interferences.

3.3 Multi-rate Support

According to the 802.11 standard [1], nodes in wireless networks can support multiple data rates through different modulation schemes and coding rates. A higher data rate can bring a dramatic increase in bandwidth efficiency. TABLE 1 lists several data rates and modulation schemes that 802.11a supports [1], [16]. The choice of which rate to be used depends on the current channel environment. The rate of each transmission link in the network can either be configured manually or selected automatically [17].

TABLE 1
Several Data Rates and Modulation Schemes That 802.11a Supports.

Data rate ($Mbps$)	Modulation scheme	Coding rate	SINR threshold(dB)
6	BPSK	1/2	6.02
9	BPSK	3/4	7.78
12	QPSK	1/2	9.03
18	QPSK	3/4	10.79
24	16-QAM	1/2	17.04
36	16-QAM	3/4	18.80
48	64-QAM	1/2	24.05
54	64-QAM	3/4	24.56

The 802.11 standard supports multiple data rates as follows [1]: All the control frames and DATA frames have a PLCP header, which contains an 8-bit signal field that indicates the selected rate at which bits in the MAC layer will be transmitted. When a node begins to transmit a frame, it first transmits the PLCP preamble and PLCP header at the minimum rate (such as 6$Mbits/s$ in 802.11a), then switches to the selected rate in the signal field to transmit the following bits. In the receiving process, after detecting the PLCP preamble and PLCP header correctly, the node switches to the rate in the signal field to receive the following bits.

In IRMA, different data rates will severely affect the behavior of nearby nodes around the transmission link. As shown in TABLE 1, the SINR threshold increases along with the increase of the transmission rate when different modulation schemes are adopted. That means, in Fig. 4, although $S3$ is out of the interference range of $R1$ when the transmission rate of $S1 \rightarrow R1$ is low (such as 6Mbits/s in 802.11a), it can have a high probability to be in the interference range of $R1$ when the transmission rate increases to a higher value. Meanwhile, the NAV information should also be adjusted according to the rate. As the transmission rate will strongly affect the NAV information and interference range, it should be carried in the CTS frame, and also be detected correctly in the case of collisions, so that the nearby nodes can make a proper decision to exploit concurrency and avoid interferences.

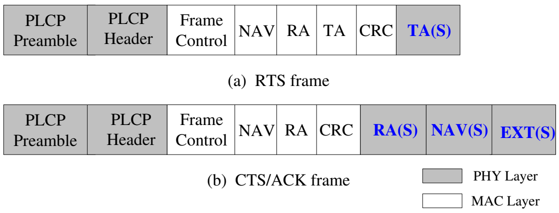
4 Design of IRMA Protocol

This section describes the design of the IRMA protocol. We first accomplish the control frame design according to the analysis in Section 3, then we discuss the new blocks of IRMA (gray-color blocks shown in Fig. 3), including signature attachment, signature detection method (SDM), channel access scheme, and differentiated NAV state update. In the end, we discuss an address conflict problem when multiple nodes choose the same signature as their own addresses.

4.1 Control Frame Design

In Section 3, we have summarized four key control information that should be detected correctly when a collision occurs: the receiver address (RA), the NAV information, the interference range and the transmission rate. Therefore, we design new control frames to make these control information detectable using SDM. We add some new fields to the 802.11 standard control frames, as shown in Fig. 5. In the transmitting process, these new fields are filled with signatures that carry specific control information. In the receiving process, nodes can recover the information after discerning signatures in corresponding fields.

For the new RTS frame, we attach a new field called TA(S) to the tail of the frame in the physical layer, as shown in Fig. 5(a). A transmitter will assign a signature, which represents its own address, in the TA(S) field of the RTS frame. For the new CTS or ACK frame, three new fields RA(S), NAV(S) and EXT(S), which are also filled with signatures, are attached to the tail of the frame, as shown in Fig. 5(b). RA(S) indicates the receiver address of the frame, which will be filled directly with the TA(S) signature derived from the received RTS frame. NAV(S) indicates the NAV information represented by a signature. EXT(S) carries the combined information of both the data



(a) RTS frame

(b) CTS/ACK frame

Fig. 5. The format of new RTS/CTS/ACK frames.

rate and interference range (IR) of the ongoing transmission link[1].

Each of the TA(S)/RA(S), NAV(S) and EXT(S) needs a group of global-unique signatures to represent their information. We design a signature set $S_{Addr} = \{s_1, \ldots, s_p\}$ for TA(S)/RA(S). A node can randomly select a signature $s_i (i = 1, 2, ..., p)$ from the set as its own signature, and put it in the TA(S) field when sending a RTS frame.

We then design a signature set $S_{NAV} = \{s_1, \ldots, s_q, s_{ACK}\}$ for NAV(S), where $s_k (k = 1 \sim q)$ represents the NAV time, and s_{ACK} is a unique signature specially used to differentiate the CTS and ACK frames. With the s_{ACK}, the node, such as $S3$ in Fig. 4, will not misinterpret the $R1$'s ACK feedback as a CTS frame when a collision occurs.

We also design a signature set $S_{EXT} = \{S_{m \times n}\}$ for EXT(S), to represent the combination of the data rate and interference range:

$$\mathbf{S_{m\times n}} = \begin{pmatrix} s_{11} & s_{12} & \ldots & s_{1n} \\ s_{21} & s_{22} & \ldots & s_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{m1} & s_{m2} & \ldots & s_{mn} \end{pmatrix},$$

where s_{ij} is a signature, the rate indicator i represents a data transmission rate, and the IR indicator j represents an interference range. Here, m and n denote the maximal values of i and j, respectively.

We next illustrate the design of the NAV(S) and EXT(S) in detail. To simplify the description of the system, we denote the selected data rate as v_b, the corresponding transmission range as $d_{TX}(v_b)$, and the corresponding SINR threshold as $\beta_{SINR}(v_b)$, respectively.

4.1.1 NAV(S) Design

We design a set of signatures to represent different transmission durations (the NAV time). As the length of MPDUs cannot exceed a threshold l_{max} according to the 802.11 standard, the MPDU transmission time is upper-bounded by $l_{max} \cdot 8/v_b$. We define the maximum data frame transmission time as $t_{max} = l_{max} \cdot 8/v_b + t_{PHY}$, where t_{PHY} is a constant time for transmitting the PLCP preamble and PLCP header. We divide t_{max} into q segments, each of which lasts for $L_{NAV} = \lceil \frac{t_{max}}{q} \rceil$. Therefore, each NAV time t_{NAV} can be mapped to a specific NAV indicator $k = round(\frac{t_{NAV}}{L_{NAV}})$, and s_k in the signature set S_{NAV} will be filled in the NAV(S) field of the CTS frame to carry the NAV time information.

4.1.2 Data Rate Indicator Design

The data rate indicator i represents the selected data rate v_b for the following data frame transmission. All the nearby nodes received the CTS frame should firstly obtain the data rate information, then recover the NAV time and interference range accordingly.

Each 802.11 standard can support different data rates v_b. In this paper, we use the 802.11a as an example. It can only support eight v_b values ranging from $6 Mbits/s$ to $54 Mbits/s$, as shown in TABLE 1. We set the indicator i with the value of $0 \sim 7$ to represent the eight v_b values respectively.

4.1.3 IR Indicator Design

The IR indicator j represents the interference range of a receiver of the ongoing transmission link, with which each node around the receiver can make a proper decision about whether its transmission will interfere with the ongoing transmission link.

In this paper, we use the two-ray ground propagation model [18], which is widely adopted in wireless network research studies (such as [3], [10]) as well as the network simulators (such as ns-2 [19]). Based on this model, the receiving power P_r of a signal is inversely proportional to the distance d between the transmitter and receiver, i.e., $P_r = P_t G_t G_r \frac{h_t^2 h_r^2}{d^\alpha}$, where P_t is the transmission power, G_t and G_r are antenna gains of the transmitter and receiver respectively, h_t and h_r are the heights of both antennas, α is a factor larger than 2 and reflects the attenuation degree of the signal. Here, we assume all the nodes in the network are homogeneous, i.e., all the radio parameters are the same at each node, all the antenna heights are the same, and all the nodes have the same fixed transmission power P_t. Thus, we can simplify the equation to calculate P_r as $P_r = c\frac{P_t}{d^\alpha}$, where c is a constant. We also assume the radio channel is symmetric, i.e., the signal has the same propagation attenuation in both directions. Here we should admit that if the assumptions do not hold, the calculated interference range in the following parts may be deviated from the real one, thus either leading to collisions or losing some concurrent transmission opportunities, both of which will pull down IRMA's performance.

If a node receives a RTS frame, the receiving power P_r is obtained, and the data rate can be determined to be v_b. The node first uses P_r to compute its distance d from the sender. Then, according to the physical interference model, it computes the interference range d_{IR} of this transmission link using the formula below:

$$SINR = \frac{c\frac{P_t}{d^\alpha}}{c\frac{P_t}{d_{IR}^\alpha} + P_I + P_N} \geq \beta_{SINR}(v_b),$$

where P_I indicates the cumulated interference power from other concurrent transmissions, P_N is the thermal noise power and it can be ignored. Suppose P_I is negligible

[1] Though we can use two different signatures to carry both information, we use one EXT(S) signature in this paper to reduce the introduced transmission overhead of signatures.

comparing with $c\frac{P_t}{d_{IR}^{\alpha}}$, we have:

$$SINR \approx \frac{c\frac{P_t}{d^{\alpha}}}{c\frac{P_t}{d_{IR}^{\alpha}}} = (\frac{d_{IR}}{d})^{\alpha} \geq \beta_{SINR}(v_b),$$

which means only nodes that are at least $d_{IR} = d \cdot \sqrt[\alpha]{\beta_{SINR}(v_b)}$ away from the receiver are permitted to transmit concurrently. Note that the threshold $\beta_{SINR}(v_b)$ is related to the selected data rate v_b. Generally, the transmission link with higher v_b has a larger interference range.

Once v_b is determined, the transmission range $d_{TX}(v_b)$ is also fixed. Then, the maximum interference range is

$$d_{IR\_max} = d_{TX}(v_b) \cdot \sqrt[\alpha]{\beta_{SINR}(v_b)}. \quad (1)$$

We divide d_{IR_max} into n segments, each of which has a length $L_{IR} = \lceil \frac{d_{IR\_max}}{n} \rceil$, then each interference range d_{IR} can be mapped to a specific IR indicator $j = \lceil \frac{d_{IR}}{L_{IR}} \rceil$.

It is noted that in certain real scenarios, the real interference range may be larger than the calculated d_{IR}, which is $d \cdot \sqrt[\alpha]{\beta_{SINR}(v_b)}$, due to multiple nodes' concurrent transmissions (P_I is not negligible). This problem can be partially mitigated since the interference range information carried by j is generally larger than d_{IR}. Moreover, the problem can also be mitigated by using a higher $\beta_{SINR}(v_b)$ when calculating d_{IR}, which makes the transmitter convey a larger interference range information in the CTS. The value of $\beta_{SINR}(v_b)$ that can maximize the performance should be determined by the real network scenarios.

4.2 Signature Attachment

Fig. 3 shows that when a transmitter/receiver intends to transmit a control frame RTS/CTS/ACK, it will generate specific signatures and put them in the corresponding fields according to the format of these frames shown in Fig. 5.

When a transmitter intends to send a RTS frame for initiating a transmission, it randomly selects a signature from the signature set S_{Addr} as the TA(S), trying to make the signature unique in the vicinity of the transmitter/reciever each time.

Upon receiving a RTS frame, a node first checks if it is the designated receiver. If the node is the designated receiver, it will generate a CTS frame with required signatures. It fills the RA(S) field of the CTS frame with the TA(S) that is directly obtained from the received RTS frame.

For the NAV(S) field of the CTS frame, it calculates its NAV time t_{NAV} according to the NAV time set in the RTS frame, by subtracting the SIFS and the transmission time of the CTS frame. It then calculates L_{NAV} to be $\lceil \frac{t_{max}}{q} \rceil$, according to the selected data transmission rate v_b. The resultant time duration is mapped to a NAV indicator $k = \lceil \frac{t_{NAV}}{L_{NAV}} \rceil$, and the signature s_k of S_{NAV} is put in the NAV(S) field.

For the EXT(S) field, the node first sets the rate indicator i according to the selected data rate v_b. It then calculates the interference range d_{IR} through the receiving power of the RTS frame and v_b, then maps d_{IR} to an IR indicator $j = \lceil \frac{d_{IR}}{L_{IR}} \rceil$. The signature s_{ij} of $S_{m \times n}$ is put in the EXT(S) field. The node finally replies this CTS frame to the transmitter.

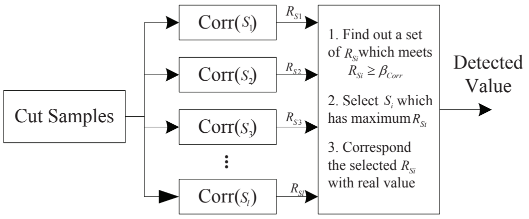

Cut Samples
Corr(S_1)
Corr(S_2)
Corr(S_3)
Corr(S_l)
R_{S1}
R_{S2}
R_{S3}
R_{Sl}
1. Find out a set of R_{Si} which meets $R_{Si} \geq \beta_{Corr}$
2. Select S_i which has maximum R_{Si}
3. Correspond the selected R_{Si} with real value
Detected Value

Fig. 6. The signature detection method that discerns a known signature s_i from the incoming samples and recovers the original information. The signature set here is $\{s_1 \dots s_l\}$.

Similarly, when the node intends to reply an ACK frame after a successful data reception, besides the RA(S) field, it puts the s_{ACK} signature in the NAV(S) field and then broadcasts it.

4.3 Signature Detection Method

Since signatures that carry useful information are attached in control frames when they are transmitted, nodes can use the SDM to discern these signatures from incoming control frame's samples and recover the original information.

We construct three signature sets S_{Addr}, S_{NAV} and S_{EXT}, containing p, $q+1$ and $m \times n$ signatures, respectively. To illustrate the SDM in general, we assume the number of signatures in a signature set is l. The SDM first discerns which signature can be found in the incoming samples. As shown in Fig. 6, the cross correlation is conducted between the incoming samples and each of the l known signatures, the outputs of l correlation values $R_{s_1}, \dots, R_{s_l}$ are compared with the threshold β_{Corr}. We select the s_i which has the maximum value R_{s_i} among those ones that exceed β_{Corr}. Fig. 7 demonstrates an example of SDM in which l equals to 16. Although both R_{s_5} and R_{s_8} are over the threshold β_{Corr}, we decide s_5, whose correlation result is maximum, is in the received signal.

As all the signatures have their fixed positions after the preamble field in the control frames, as shown in Fig. 5, nodes can easily obtain the positions of signatures by offsetting the fixed number of samples after the position of the preamble is determined. Thus, the SDM only needs to cross-correlate the "cut samples" (i.e., the fixed-length samples at certain positions) of the incoming signal with the known signatures. This mechanism makes the computational complexity of the SDM in the order of the size of the signature set.

For the TA(S), since IRMA does not permit the RTS frame to be collided at the receiver, TA(S) can be easily decoded by the receiver.

For the detection of the RA(S), the correlation process should just be performed one time between the incoming cut samples at the RA(S) field and its own signature. The node simply determines itself to be the designated receiver of the frame if the correlation value exceeds β_{Corr}.

For the detection of the NAV(S), the node should perform the correlation process $q+1$ times between the incoming cut

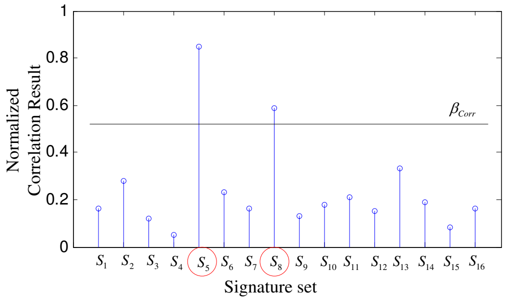


Fig. 7. An example of the signature detection method.

samples at the NAV(S) field and each signature in S_{NAV}. If the discerned signature is s_{ACK}, the node determines the received frame to be an ACK. If the signature is a signature s_k in S_{NAV}, the node determines the received frame to be a CTS, then calculates L_{NAV} to be $\lceil\frac{t_{max}}{q}\rceil$, and converts the NAV time to be $k \cdot L_{NAV}$. Note that t_{max} is determined by the data rate v_b, which can be obtained after detecting the EXT(S) field.

For the detection of the EXT(S), the node should perform the correlation process $m \times n$ times between the incoming cut samples at the EXT(S) field and each signature in S_{EXT}. If the signature is a signature s_{ij} of $S_{m\times n}$, the node first determines the data rate of the corresponding transmission link to be v_b according to the rate indicator i, then calculates the L_{IR} to be $\lceil\frac{d_{TX}(v_b)\cdot\sqrt[\alpha]{\beta_{SINR}(v_b)}}{n}\rceil$, and converts the interference range to be $j \cdot L_{IR}$.

4.4 Channel Access Scheme

IRMA disables the physical carrier sense and only relies on the NAV state, which is set by the virtual carrier sense, to avoid the interference caused by data transmissions. When a node intends to send a data frame, it first checks the NAV state and waits the NAV state to become zero. Then, it will initiate a RTS transmission after a backoff time.Meanwhile, when a node receives a RTS and detects the frame correctly, it should also check the NAV state, and respond a CTS after SIFS if the NAV state is zero.

The NAV state in IRMA is merely updated by the NAV or NAV(S) field in the CTS frame, which is different from the mechanism used in the 802.11 standard, where the NAV state is updated by the NAV field in either the RTS or CTS frame. Since IRMA permits concurrent transmissions at the transmitter side, the NAV field in the RTS frame is not used to update the NAV state. It is also conditional in IRMA to use the detected NAV or NAV(S) value in the CTS frame to update the NAV state, as it may make nodes miss transmission opportunities. As shown in Fig. 4, when $R1$ replies a CTS to $S1$, $S2$ also detects the EXT(S) correctly. As $S2$ does not interfere with $R1$'s data reception, it will miss a transmission opportunity if its NAV state is updated by the NAV(S) field of the received CTS. To make a proper channel access decision, we propose the following differentiated NAV state update scheme to solve this problem.

4.5 Differentiated NAV State Update

In this scheme, upon receiving a CTS frame, nodes that are in the interference range of the ongoing transmission link should update their NAV states, while other nodes do not update the NAV states so that they do not waste transmission opportunities.

When the EXT(S) field in the CTS frame is detected to be s_{ij}, the transmission rate can be determined to be v_b according to the indicator i, and the interference range d_{IR} can be determined to be $j \cdot L_{IR}$, where $L_{IR} = \lceil\frac{d_{TX}(v_b)\cdot\sqrt[\alpha]{\beta_{SINR}(v_b)}}{n}\rceil$. We use a threshold β_{rCTS} to represent the signal strength at a position which is d_{IR} away from the CTS transmitter, that is, $\beta_{rCTS} = P_t G_t G_r \frac{h_t^2 h_r^2}{d_{IR}^{\alpha}}$. By comparing the signal strength of the received CTS frame with β_{rCTS}, a node can decide whether its concurrent transmission would interfere with the ongoing transmission link or not. The signal strength of the CTS frame in the presence of an interference can be obtained through an easy way: As a sharp change appears in the amplitude variation of the received signal when a new frame arrives [20], together with the signal strength before and after the sharp change, we can determine what the frame's power level is.

For the scenario shown in Fig. 4, both $S1$ and $S3$ can receive the CTS from $R2$, but the signal strength of this frame is above β_{rCTS} at $S3$, and below β_{rCTS} at $S1$. Thus, $S3$ concludes that it should update its NAV state, while $S1$ would not yet.

The NAV state of each node can be updated by the control information carried in a CTS frame in two different ways: It can be updated by the NAV field of the frame at the MAC layer; it can also be updated by the NAV(S) field of the frame at the physical layer. As the NAV field carries more precise NAV time information than the NAV(S) field, the former has a higher priority in the NAV state update process. If the frame can be correctly decoded at the MAC layer, the node's NAV state will be updated by the decoded NAV value; otherwise, it will be updated by the NAV time determined by the NAV(S) field, that is, $k \cdot L_{NAV}$, where s_k is the discerned signature in the NAV(S) field, and $L_{NAV} = \lceil\frac{t_{max}}{q}\rceil$. Note that the inconsistent NAV time carried in the NAV(S) field will not change the node's channel access opportunity to a great extent. For example, if l_{max}=1500bytes, $v_b = 6Mbits/s$ and $q = 150$, the maximum difference induced by NAV(S) is $e_{max} = \frac{L_{NAV}}{2} \approx \frac{1500\times 8}{6\times 10^6\times 150\times 2} \approx 6.5us$. This value is even smaller than one time slot ($9us$) used in the backoff process.

4.6 Signature Detection Range Analysis

IRMA prohibits nodes that are within the interference range but outside the transmission range of the ongoing link from transmitting data, so as to avoid collisions, as shown in Fig. 1(c). Thus, nodes within the interference range d_{IR} should have the ability to detect the CTS frame correctly to keep silent. In this part, we will give the theoretical analysis about it.

To simplify the analysis, we define the signature detection range d_S as the maximum range within which

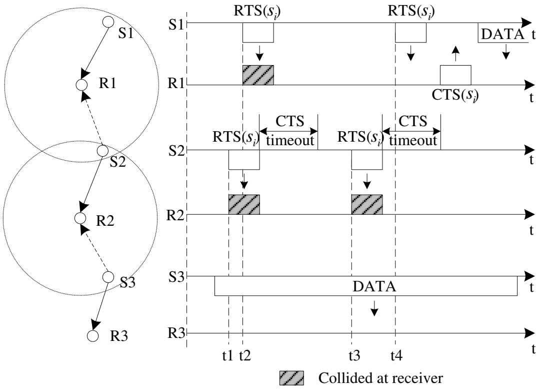


Fig. 8. A potential address conflict scenario. The two circles indicate the transmission range of $R1$ and $R2$, respectively.

signatures can be detected correctly. Here we first formulate d_S, then compare it with the maximum interference range d_{IR_max}, which has been discussed in Section 4.1.3.

According to the physical interference model, a data frame can be detected correctly if the SINR of the received signal power is above a threshold β_{SINR}, that is:

$$SINR = \frac{P_r(data)}{P_I^1 + P_N} = \frac{c\frac{P_t}{d^\alpha}}{P_I^1 + P_N} \geq \beta_{SINR},$$

where P_I^1 indicates the cumulated interference power at the data frame's receiver side. When $d = d_{TX}$, we set $SINR = \beta_{SINR}$, then we have:

$$d_{TX} = \sqrt[\alpha]{\frac{c \cdot P_t}{(P_I^1 + P_N) \cdot \beta_{SINR}}}. \tag{2}$$

We define the signature detection threshold β_{SINR_S} as the threshold that, a signature can be detected correctly through SDM if its SINR is above β_{SINR_S}, that is:

$$SINR = \frac{P_r(signature)}{P_I^2 + P_N} = \frac{c\frac{P_t}{d^\alpha}}{P_I^2 + P_N} \geq \beta_{SINR\_S},$$

where P_I^2 indicates the cumulated interference power at the signature's receiver side. When $d = d_S$, we set $SINR = \beta_{SINR\_S}$, then we have:

$$d_S = \sqrt[\alpha]{\frac{c \cdot P_t}{(P_I^2 + P_N) \cdot \beta_{SINR\_S}}}. \tag{3}$$

Suppose $P_I^1 = P_I^2$. Combining Eq. (2) and Eq. (3), we get:

$$d_S = \sqrt[\alpha]{\frac{\beta_{SINR}}{\beta_{SINR\_S}} \cdot \frac{P_I^1}{P_I^2}} \cdot d_{TX} = \sqrt[\alpha]{\frac{\beta_{SINR}}{\beta_{SINR\_S}}} \cdot d_{TX}. \tag{4}$$

According to the experiment results in Section 5.2.2, the signatures can be detected correctly with the probability of about 100% when the signature length is 160bits and the SINR is above -10dB, thus we set $\beta_{SINR\_S} = -10dB = 0.1$. Through Eq. (1) and Eq. (4), we get:

$$d_S = \sqrt[\alpha]{\frac{\beta_{SINR}}{0.1}} \cdot d_{TX} > \sqrt[\alpha]{\beta_{SINR}} \cdot d_{TX} = d_{IR\_max}.$$

Therefore, we can conclude that the signature detection range is always larger than the interference range, and the nodes within the interference range can detect the CTS frame correctly to keep silent. Note that P_I^1 may not equal to P_I^2 because of the different environments at the data frame's receiver side and the signature's receiver side in the real network, if $P_I^1 \gg P_I^2$ (e.g. , $P_I^1 > 10 \cdot P_I^2$), the signature detection range may be smaller than the interference range, leading to a collision to the ongoing link. However, we consider this case is rare since it does not appear in the simulations.

4.7 Address Conflict Analysis

Though each transmitter independently chooses its signature TA(S) from S_{Addr} each time when it transmits a RTS frame, it still has a small probability to pick up the same signature used by other nearby nodes, leading to a potential address conflict problem. Fortunately, this address conflict can be naturally resolved in our protocol due to the CTS timeout mechanism that has already been specified in 802.11 standard [1].

We illustrate this mechanism through a scenario shown in Fig. 8. Suppose $S2$ sends a RTS frame to $R2$ using a signature s_i in the TA(S) field, this RTS is collided at $R2$ with a data frame sent from $S3$, leading to no CTS feedback from $R2$ to $S2$. $S2$ will wait for the CTS feedback only for a CTS timeout interval, then it can initiate a retry if it fails to receive the CTS. The CTS timeout interval is always set to be SIFS plus the transmission time of CTS [1]. Suppose $S1$ intends to send a data frame to $R1$, and it also selects the same signature s_i in the TA(S) field of the corresponding RTS frame. However, $S2$ will not misinterpret $R1$'s CTS as its own feedback although this CTS's RA(S) field has the same value s_i as the one used in $S2$'s RTS. We consider there are two possible situations in this scenario. The first one is that both RTS frames are collided. As shown in Fig. 8, when $S2$ sends a RTS at $t1$, and $S1$ sends a RTS at $t2$, there is no CTS feedback as the two RTS frames are both collided. The second situation is that $S2$ experiences a CTS timeout before receiving a CTS frame. As shown in Fig. 8, when $S2$ initiates a retry and sends a RTS at $t3$, and $S1$ sends a RTS at $t4$, $S2$ will not misinterpret $R1$'s CTS as its own feedback as this CTS is not within its CTS timeout interval. $S1$ then transmits its data frame after receiving the CTS, without any collision at $R1$.

Meanwhile, since the medium has been reserved by the RTS/CTS handshake before an actual data transmission, no conflict will appear in the ACK frame transmission.

5 Experiment Evaluation

In this section, we evaluate the feasibility of using the SDM to detect control frames' signatures in the presence of interference through hardware experiments. The experiments use Universal Software Radio Peripheral 2 (USRP2) package tool and the open source GNU Radio for the signal processing blocks. We implement IRMA on an 8-node USRP2 testbed and the topology is randomly set up in our labs. Each node is a USRP2 connected to a commodity PC that configured GNU Radio, and each USRP2 operates at 2.4GHz with a sample rate of 2M samples/sec. We choose DBPSK as the modulation method in the experiment.

A real time performance evaluation based on USRPs is difficult because of the hardware delays in obtaining samples from the RF front-end to the connected PC, and also the artificial software delays induced by GNU Radio. Therefore, we also resort to trace-based evaluation that is used in [5], [11], [13], [14]. Each node saves all the incoming samples for off-line processing.

For each experiment, we pick up four nodes to form two links, each of which has a sender and a receiver. The two selected senders should be exposed terminals and IRMA permits their concurrent transmissions, so as to generate control frame collisions at the sender side. Different SINR environments are tested for SDM by adjusting the transmission power of one sender and fixing that of the other.

Similar to CSMA/CN [13], we also use two metrics, the false negative error rate and false positive error rate, to measure the performance of SDM in this part. The difference from CSMA/CN is that, we focus on designing the signature set that will be used in this protocol. Thus, we need to strike a balance among the ability of combating interferences, the signature length and the size of signature set through this experiment.

5.1 Threshold β_{Corr}

The SDM determines if a known signature is found in the incoming samples by performing the cross correlation between the two signals. In the correlation process, the normalized correlation peak is always detected by comparing the peak value with a threshold β_{Corr}, as described in Section 2. A higher threshold $\beta_{Corr}(high)$ can lead to more false negative errors, and a lower threshold $\beta_{Corr}(low)$ can lead to more false positive errors. Both errors will make the CTS or ACK receivers get wrong information, leading to either collisions in data packet transmissions or failure to exploit concurrent transmissions, thus degrading the network throughput. To make a tradeoff between the two errors, we adjust the threshold β_{Corr} of the normalized correlation process to be ψ, where ψ is set to be 0.55 in the experiment.

Empirically, the false negative error rate is closely related to L and SINR, as a shorter L or lower SINR would lead to a lower correlation peak. The false positive error rate is more affected by the Hamming distance between the signature and the correlated incoming samples, as a shorter distance would lead to a higher false correlation peak. In the experiment, we try to mitigate both error rates from the two aspects.

5.2 Signature Detection Evaluation

In this part, we quantify SDM's ability to detect signatures at the presence of strong interferences. We also demonstrate that the size of the signature set is large enough to meet the protocol's requirements. Our experiment starts from mitigating the two errors in the correlation process.

5.2.1 *False Negative Error*

In order to quantify how the false negative error can be affected by the signature length L and SINR, we test three sets of data with three signature lengths L under different SINR environments and channel conditions. For each L, we conduct the experiment for ten times in two places, and each time we select four different nodes from the testbed to form the two links. As shown in Fig. 9, the results clearly show that the correlation spike appears even under strong interferences where SINR is $-20dB$. The longer the signature is, the easier the signature is to be detected. Moreover, when both the SINR value and L are fixed, the correlation value has a variance in a certain range, which is induced by various channel environments. In the following parts, we will use the average correlation results to calculate the false negative and false positive error rates.

Fig. 10 demonstrates the false negative error of the three sets of data. The result shows that a longer signature (such as 200 bits) can have a lower false negative error under the same SINR, and the error decreases significantly when the SINR increases.

We use the SDM to detect signatures in the presence of interferences. The lower SINR the SDM can support, the more transmission opportunities the nodes can explore. Here we make a tradeoff between the SDM's detection ability and the signature length L. We set the minimum SINR that the SDM can support to be $-10dB$. To minimize the false positive error, we select the L to be 160 bits, then the error rate is below 0.3% when the SINR is above $-10dB$.

Note that when L is 160 bits and SINR is below $-10dB$, the false negative error rate is a little more than that was tested by CSMA/CN, that's because the correlation result has a wider fluctuation range when SINR is lower (as shown in Fig. 9), thus the calculated positive/negative error rates will be more affected by the channel characteristics in the experiment.

5.2.2 *False Positive Error*

Here we explore how to mitigate the false positive error in the correlation process. Fig. 11 shows both false positive error rate and false negative error rate when L is 160 bits. The result indicates that the SINR has almost no effect on the false positive error when the SINR is above $-14dB$, as

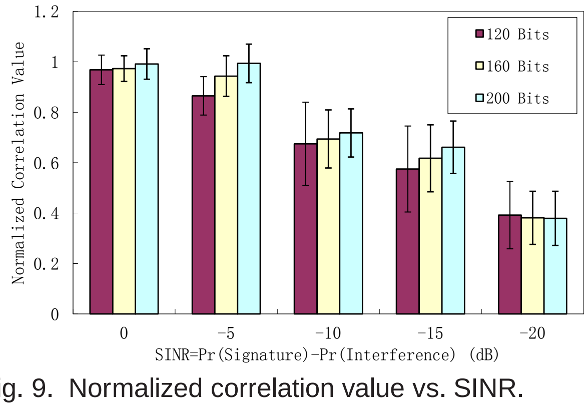


Fig. 9. Normalized correlation value vs. SINR.

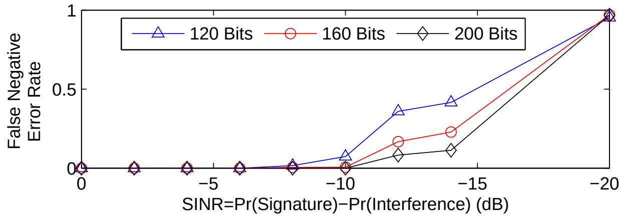


Fig. 10. False negative error rate.

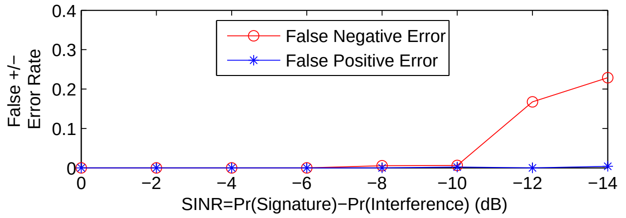


Fig. 11. False positive/negative error rates for signatures with 160 bits.

there is enough Hamming distance between the signature and the correlated samples. As shown in Fig. 11, when the Hamming distance is 52, the false positive error rate is below 0.5% when the SINR is $-14dB$.

Table 2 shows the false positive error rates under various Hamming distances when the SINR is $-10dB$, which indicates that the false positive error rate decreases when the Hamming distance increases.

TABLE 2
False positive error rates under various hamming distances (SINR=$-10dB$).

Hamming distance	34	40	46	52
False positive error rate	0.170	0.047	0.008	0.002

When we set the signature length to be 160 bits and the minimum Hamming distance between any two signatures to be 52, the SDM can achieve a very low signature detection error rate (less than 1%) even when the SINR is $-10dB$. We use the pseudo-noise code in this paper to accomplish the signature design. We have more than 200 signatures with 160 bits and the Hamming distance between any pairs of them is above 52.

6 Performance Evaluation

In this section we evaluate IRMA's throughput improvement in wireless networks compared with 802.11 standard and three recent protocols under two topology scenarios, a linear topology and a random topology. The two mechanisms of CSMA in the 802.11 standard that we choose to compare are (1) PCS, which uses the standard's physical carrier sense mechanism to access a wireless channel, (2) PCS+VCS, which uses both the physical and virtual carrier sense mechanisms to access a wireless channel. The three protocols we choose to compare are CMAP [2], SDN [3] and 802.11ec [5]. CMAP and SDN are two typical recent protocols that solve the exposed terminal problem, as described in Section 1, and 802.11ec is a recent protocol that exploits cross correlation to avoid collisions and improve the network throughput. We implement all the protocols in ns-2.

For IRMA, we do not implement the signature detection process in ns-2, but utilize the experiment results and the SINR of the received signal to determine whether the control frames can be detected or not. We implement it as follows: If $SINR > -5dB$, the signatures can be obtained correctly with the probability of 100%; if $-10dB < SINR \leq -5dB$, the probability is 99%; otherwise, if $SINR \leq -10dB$, the signatures will be ignored. Note that the strength of interference used to calculate the SINR is measured as the accumulated signal strengths from all other transmitting nodes, which has already been implemented by ns-2. In the simulations, we use manually configured data rate and do not implement the automatic rate adaption mechanism. As IRMA has no modification to the data frame transmissions, we consider the throughput gain of IRMA benefitted from the rate adaption is the same as any other protocols.

Table 3 lists the basic configuring parameters used in our simulation.

TABLE 3
Simulation parameters.

Parameter	Value	Parameter	Value
Preamble	$20\mu s$	SIFS	$16\mu s$
Time slot	$9\mu s$	DIFS	$34\mu s$
Signature	$13.3\mu s$	CWmax	$1023\mu s$
p/q	20/150	CWmin	$15\mu s$
m	8	n	16

The aim of our simulations is to discover how each protocol can exploit concurrency and avoid collisions to improve the network performance in different transmission rates. Hence, we select three values of v_b defined in the 802.11a standard to evaluate the performance of each protocol in the two simulation scenarios, the corresponding transmission ranges $d_{TX}(v_b)$, carrier sense ranges $d_{CS}(v_b)$ and the SINR thresholds $\beta_{SINR}(v_b)$ are all listed in Table 4. Note that we use default SINR thresholds in ns-2 in the simulation, thus the values of $\beta_{SINR}(v_b)$ may be slightly different from that in Table 1. We let the PLCP preamble and header fields that are in the physical layer be transmitted using basic modulation (BPSK), that means, the transmission rate for these fields are fixed to $6Mbits/s$, all the transmission rate of control frames and the data frames in the MAC layer will be changed to the configured rate.

TABLE 4
Three transmission rates selected in the simulation.

v_b	$d_{TX}(v_b)$	$d_{CS}(v_b)$	$\beta_{SINR}(v_b)$
$6Mbits/s$	$500m$	$600m$	$5.0dB$
$24Mbits/s$	$150m$	$600m$	$15.0dB$
$48Mbits/s$	$50m$	$600m$	$25.0dB$

6.1 Linear Topology

We first conduct the simulation under a four-node linear topology to evaluate the effectiveness of IRMA compared

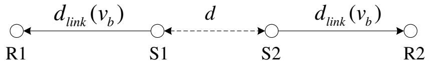


Fig. 12. A linear topology, where four nodes $R1$, $S1$, $S2$ and $R2$ form a line. The distance between $S1$ and $R1$ as well as that between $S2$ and $R2$ are both set to $d_{link}(v_b)$. The distance between $S1$ and $S2$, denoted by d, is varied from $50m$ to $700m$.

with the other five protocols and also their constraints. As shown in Fig. 12, the network contains two link pairs $S1 \rightarrow R1$ and $S2 \rightarrow R2$. The sender-receiver distances of both links $d_{link}(v_b)$ are fixed to $250m$, $80m$ and $30m$ when the transmission rate v_b is set to be $6Mbits/s$, $24Mbits/s$ and $48Mbits/s$, respectively. The distance between $S1$ and $S2$ is denoted by d, which will be varied from $50m$ to $700m$. Each link pair has a CBR (Constant Bit Rate) flow set up at the sender to be transmitted to the receiver. We evaluate the network throughput by adjusting the distance d. The packet delivery rate (flow rate) and packet length will also be changed in the simulation to get more detailed evaluation. Note that the packet length here means the length in the upper layer; obviously, the frame length should be a little longer as additional headers will be attached in the MAC layer and physical layer.

6.1.1 The impact of distance d

The IRMA protocol can exploit concurrent transmissions when there is no mutual interference in their data frame receptions, and can avoid collisions when mutual interferences exist. Therefore, we first evaluate the impact of the distance d in the following simulation. The simulation is conducted for three times, each with a different transmission rate listed in Table 4, according to which the sender-receiver distance $d_{link}(v_b)$ should also be adjusted. The packet length is fixed to 1500 bytes.

Fig. 13 shows the aggregate throughput of the network with three transmission rates. The simulation results can be summarized into six cases:

Case 1: Only one transmission is permitted by all protocols when the following conditions hold:

$$d < d_{link}(v_b) \cdot (\sqrt[k]{\beta_{SINR}(v_b)} - 1),$$

$$d \leq d_{TX}(v_b).$$

In this case, each sender is in the interference range of the other transmission link, and it is in the transmission range of the other sender, concurrent transmissions can be prohibited successfully by all protocols to avoid the interferences to the reception of the data frame, such as the scenario when $d \leq 250m$ in Fig. 13(a), the scenario when $d \leq 150m$ in Fig. 13(b) and the scenario when $d \leq 50m$ in Fig. 13(c).

Case 2: Only one transmission should be permitted but CMAP and SDN induce collisions when the following conditions hold:

$$d < d_{link}(v_b) \cdot (\sqrt[k]{\beta_{SINR}(v_b)} - 1),$$

$$d > d_{TX}(v_b).$$

In this case, each sender is in the interference range of the other transmission link and concurrency should be prohibited. As each sender is out of the transmission range of the other sender, it can not decode the packets from the other link correctly. IRMA and 802.11ec can avoid collisions successfully as the CTS information , which is carried by signatures in IRMA and by primitives in 802.1ec, can be obtained correctly in very low SINR environment. PCS and PCS+VCS can also avoid collisions through physical carrier sense. However, as both CMAP and SDN disable physical carrier sense, when a sender can not detect the packets from the other transmission link correctly, it will decide there is no conflict and initiate transmissions, leading to mutual interferences. The corresponding scenarios are $200 \leq d \leq 350m$ in Fig. 13(b) and $100 \leq d \leq 500m$ in Fig. 13(c).

Case 3: Concurrent transmissions are permitted by IRMA and CMAP but prohibited by other protocols when both the following conditions hold:

$$d_{link}(v_b) \cdot (\sqrt[k]{\beta_{SINR}(v_b)} - 1) < d < d_{link}(v_b) \cdot (\sqrt[k]{\beta_{SINR}(v_b)}),$$

$$d < d_{TX}(v_b).$$

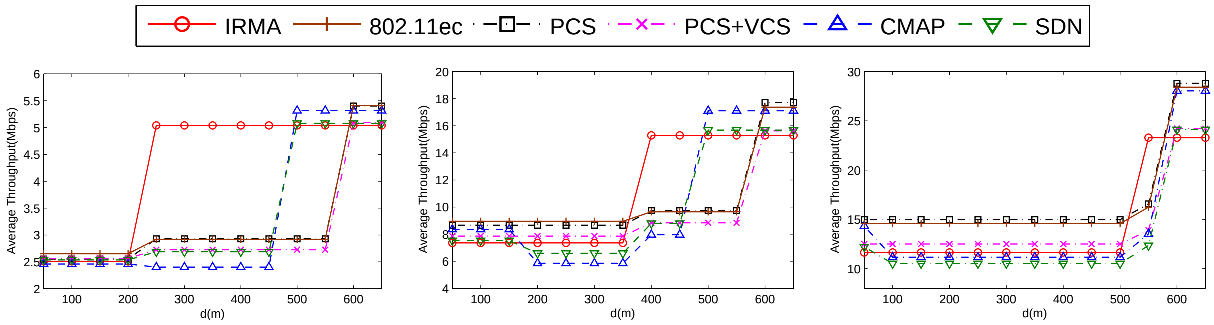
As the two transmission links have no mutual interferences to their data frame receptions, both IRMA and CMAP can exploit concurrent transmissions under this case, such as the scenarios when $250 \leq d \leq 400m$ in Fig. 13(a). However, the performance of CMAP in this scenario is even lower than that in Case 1, because of spurious retransmissions due to ACK collisions. Although CMAP design a windowed-ACK mechanism to reduce the ACK collisions, and this mechanism really can increase the throughput from about $1.6Mbits/s$ to about $2.4Mbits/s$, it can not reach the approximately 2× performance improvement as IRMA. For the other protocols, PCS, PCS+VCS and 802.11ec can only permit one link's transmission as the sender will determine the channel to be busy after physical carrier sense; SDN also only permit one link's transmission to avoid control frame collisions at the transmitter side.

Case 4: Concurrent transmissions are permitted by IRMA, CMAP and SDN but prohibited by other protocols when both the following conditions hold:

$$d_{link}(v_b) \cdot (\sqrt[k]{\beta_{SINR}(v_b)} - 1) < d < d_{link}(v_b) \cdot (\sqrt[k]{\beta_{SINR}(v_b)}),$$

$$d_{TX}(v_b) < d < d_{CS}(v_b).$$

In this case, IRMA can exploit concurrency as each sender is out of the interference range of the other link. However, CMAP and SDN permit concurrency just because one node can not correctly decode the packets from the other link and determine there is no conflict, such as the scenario of $400m \leq d \leq 500m$ in Fig. 13(b) and $d = 550m$ in Fig. 13(c). The performance of these two protocols in this case is lower than that of IRMA as both protocols face control frame collisions. PCS, PCS+VCS and 802.11ec also can only permit one link's transmission due to physical carrier sense.



(a) $v_b = 6Mbits/s$

(b) $v_b = 24Mbits/s$

(c) $v_b = 48Mbits/s$

Fig. 13. Average throughput in terms of d under three transmission rates in the linear topology.

Case 5: Concurrent transmissions are exploited by IRMA, CMAP and SDN successfully but prohibited by other protocols when both the following conditions hold:

$$d > d_{link}(v_b) \cdot (\sqrt[k]{\beta_{SINR}(v_b)}),$$

$$d < d_{CS}(v_b).$$

In this case, IRMA, CMAP and SDN can exploit concurrency as there is no mutual interferences for both data frame and control frame receptions, such as the scenario of $500 \leq d < 600m$ in Fig. 13(a) and Fig. 13(b). The performance of CMAP is a little higher than IRMA and SDN because of no overhead in transmitting RTS and CTS frames. PCS, PCS+VCS and 802.11ec can only permit one link's transmission due to physical carrier sense.

Case 6: Two links can transmit independently when the following condition holds:

$$d > d_{CS}(v_b).$$

In this case, the two transmission links are independent from each other. They can transmit simultaneously without any interferences.

We should note that Fig. 13 just shows the throughput of this scenario when the sender-receiver distance d_{link} is about one-half of the transmission range $d_{TX}(v_b)$ in each transmission rate. Obviously, if d_{link} is set to be a smaller value, concurrent transmissions can be exploited in a larger area because of the shorter interference range.

6.1.2 The impact of transmission rates

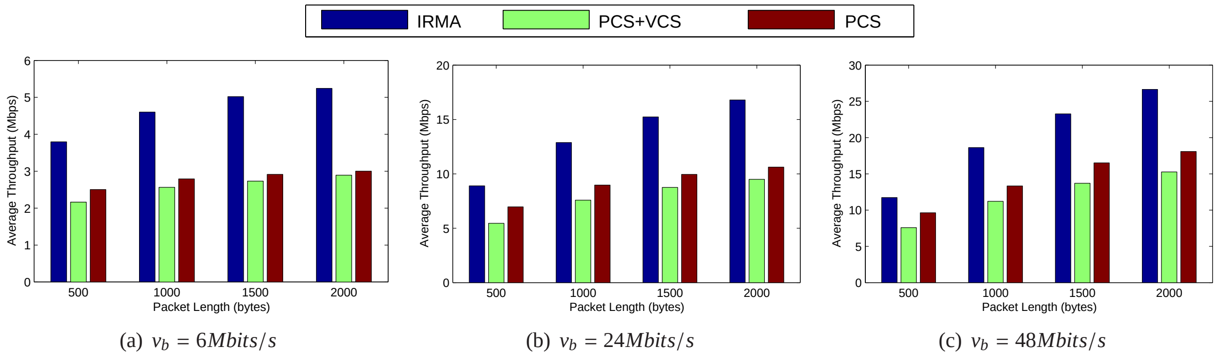
As the interference range of a transmission link with distance $d_{link}(v_b)$ can be calculated as $d > d_{link}(v_b) \cdot (\sqrt[k]{\beta_{SINR}(v_b)})$, a higher transmission rate that corresponds to a higher β_{SINR} value may lead to a larger interference range. Thus, the situation of the interference range larger than the transmission range (as described in Fig. 1(c)) will more easily occur at a higher transmission rate. For example, it will happen even when $d_{link}(v_b) = 3m$ and $v_b = 48Mbits/s$. CMAP and SDN are more vulnerable to this situation, leading to a performance degradation due to collisions. On the contrary, IRMA and 802.11ec can combat this problem successfully as their CTS and ACK information can be detected in very low SINR environments. PCS and VCS+PCS can also partially handle this problem as the carrier sense range is always much larger than the transmission range, collisions may be avoided due to the physical carrier sense mechanism.

Meanwhile, we should admit that the throughput improvement of IRMA decreases along with the increases of the transmission rate. As shown in Fig. 13(c), when the rate is $48Mbits/s$, the performance improvement of IRMA is significantly lower than twice over PCS or PCS+VCS due to the reason that the overhead induced by signatures is much larger in a higher transmission rate.

6.1.3 The Impact of Signatures

IRMA adds signatures in the control frames to convey the information that can be detected when a collision occurs, as shown in Fig. 5. Introducing the signatures into control frames will cause throughput degradation. To evaluate the overhead induced by signatures, we conduct three simulations with three transmission rates shown in Table 4, and the packet lengths are 500, 1000, 1500 and $2000bytes$, respectively. We set the distance d to a specific value so that the conditions of Case 2 or Case 3 can hold. Under these conditions, the throughput gain of IRMA can theoretically be up to twice over PCS+VCS as concurrent transmissions of two links are exploited. However, IRMA's throughput cannot reach the expected value due to the overhead induced by transmitting signatures. The simulation results are shown in Fig. 14.

Fig. 14 indicates that the throughput gain of IRMA over 802.11 standard decreases while the transmission rate increases. For example, when the packet length is $2000bytes$, the throughput gain is about twice when the data rate is $6Mbits/s$, but this value decreases to 74% when the data rate is $24Mbits/s$, and decreases to 47% when the data rate is $48Mbits/s$. Fig. 14 also indicates that the throughput gain increases along with increases of the packet length. For example, when the transmission rate is $6Mbits/s$, the throughput gain is about 51% when the packet length is $500bytes$, much lower than that of 100% when the packet length is $2000bytes$. That is because the time proportion in transmitting the control frames increases with the data rate, and decreases with the packet length.



(a) $v_b = 6Mbits/s$ (b) $v_b = 24Mbits/s$ (c) $v_b = 48Mbits/s$

Fig. 14. Average throughput in terms of packet length under three transmission rates in the linear topology. Concurrent transmissions are exploited in this scenario.

6.1.4 *Constrains of protocols*

IRMA, SDN and CMAP will have no throughput improvement in a sparse network where there is no exposed terminal problem and no additional concurrent transmission can be exploited. In Fig. 12, when the distance d is larger (such as $700m$) and the two links have no mutual interference, PCS can have the best performance among all protocols due to the overhead induced by transmitting control frames in other protocols. We should admit that IRMA has the lowest performance in this scenario because of the overhead induce by signatures. The simulation results indicate that IRMA, SDN and CMAP are not suitable for sparse networks, where they will even reduce the network throughput.

802.11ec can tolerate control frame collisions and reduce the duration of the control frame transmissions, but it can not exploit concurrent transmissions. Furthermore, comparing with PCS and PCS+VCS, more concurrent transmissions will be prohibited by this protocol due to the high detection ability of the CTS primitive. Thus, 802.11ec is not suitable for intensive networks.

6.2 Random Topology

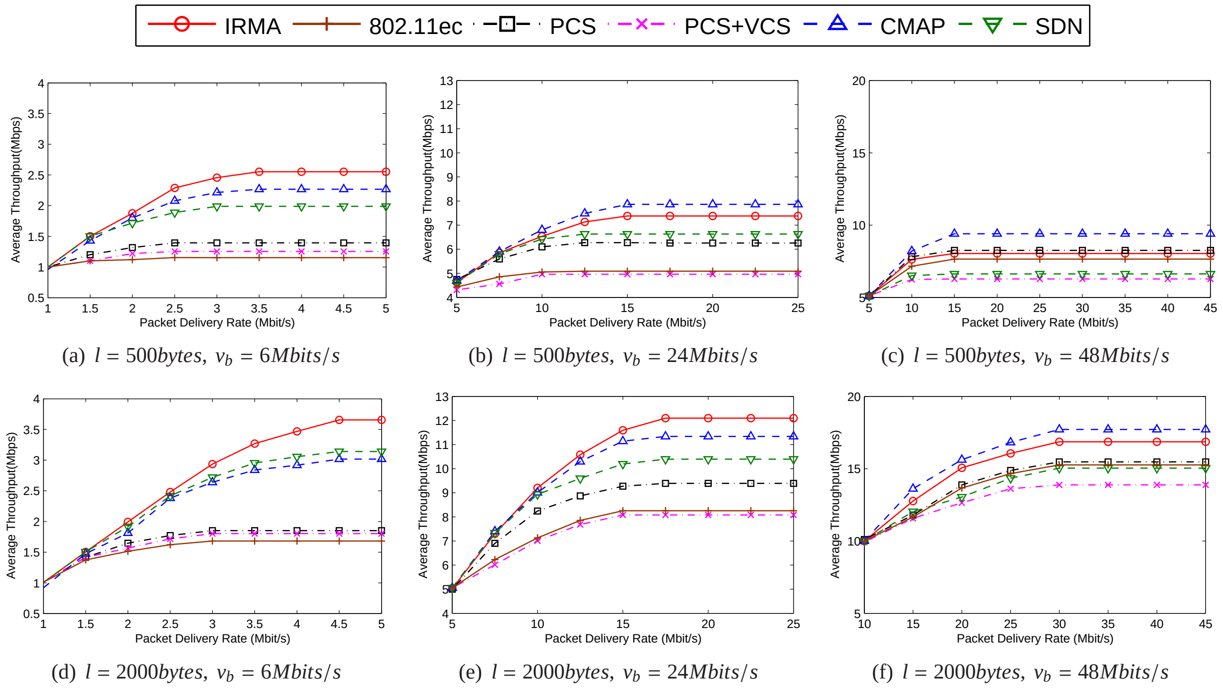
In this experiment we evaluate the performance of IRMA compared with other protocols in a general scenario where the network topology is randomly generated. We set up 10 transmitter-receiver link pairs in a $1000m \times 1000m$ area for three times to derive three configurations. For each configuration, nodes will use one transmission rate v_b listed in Table 4 to transmit packets. To set up the 10 link pairs, we first randomly generate one link (two nodes) in the area, calculate their distance and compare with the transmission range $d_{TX}(v_b)$, the link will be reserved if the distance is shorter than $d_{TX}(v_b)$; otherwise, it will be dropped. This process will be repeated for 10 times to generate 10 links in the network.

Fig. 15 shows the average throughput of IRMA comparing with the other five protocols for different packet delivery rates when the packet length is $500bytes$ and $2000bytes$, respectively. The figure indicates that the average throughput of all protocols increases along with the increases of the packet delivery rate and packet length, and the throughput is always up-bounded when the packet delivery rate reaches a specific value.

Fig. 15 also shows that PCS+VCS has the lowest performance in all the configurations, even comparing with the PCS protocol. The reason is that the hidden terminal problem is not so serious in dense networks, where PCS can avoid collisions in most cases through the physical carrier sense. 802.11ec also has low performance especially at a lower transmission rate, because, although it can tolerate control frame collisions and reduce transmission durations of control frames, it may prohibit more concurrent transmissions even comparing with PCS and PCS+VCS due to the high detection ability of the CTS primitive.

We can see that IRMA, CMAP and SDN can improve the network performance through exploiting concurrent transmissions comparing with both PCS and PCS+VCS, and IRMA can outperform other protocols in most cases, but the throughput gain decreases along with the increase of transmission rate and increases along with the increase of packet length. As shown in Figs. 15(a) and 15(d), when the transmission rate is $6Mbits/s$, IRMA's throughput gain is about 83% over PCS and 103% over PCS+VCS when the packet length is $500bytes$. These values increase to 98% and 112% respectively when the packet length is $2000bytes$. When the transmission rate increases to $24Mbits/s$, as shown in Figs. 15(b) and 15(e), the throughput gain decreases to about 17.9% over PCS and 48.7% over PCS+VCS when the packet length is $500bytes$, and about 31.5% over PCS and 50.7% over PCS+VCS when the packet length is $2000bytes$. We can also see from Figs. 15(b) that CMAP can even outperform IRMA a little at this situation. The reason is that the overhead induced by signatures in IRMA is much larger when shorter packet lengths and higher transmission rates are configured.

According to the analysis in Section 6.1, the interference range is more likely larger than the transmission range at a higher transmission rate scenario because of the higher SINR threshold, making CMAP and SDN face a high probability of collisions. In this random topology scenario, when the transmission rate increases to $48Mbits/s$, the throughput gain of SDN really decreases, but that of CMAP surprisingly increases on the contrary, as shown



(a) $l = 500bytes,\ v_b = 6Mbits/s$
(b) $l = 500bytes,\ v_b = 24Mbits/s$
(c) $l = 500bytes,\ v_b = 48Mbits/s$
(d) $l = 2000bytes,\ v_b = 6Mbits/s$
(e) $l = 2000bytes,\ v_b = 24Mbits/s$
(f) $l = 2000bytes,\ v_b = 48Mbits/s$

Fig. 15. Average throughput in terms of packet delivery rate in the random topology, under three transmission rates and two packet lengths.

in Figs. 15(c) and 15(f). After analyzing the throughput of each link, we find that CMAP faces a more serious unfairness issue in this scenario, even comparing with PCS and PCS+VCS. Fig. 16 shows a snapshot of the throughput of ten links for each protocol when the transmission rate is $48Mbits/s$ and the packet length is $2000bytes$. We see that CMAP makes five links have very high throughputs and three links have close-to-zero throughputs. This unfairness situation largely avoids collisions in the network, and allows the network to achieve a higher average throughput. We also see from Fig. 16 that IRMA, SDN and 802.11ec are relative fair among all the links.

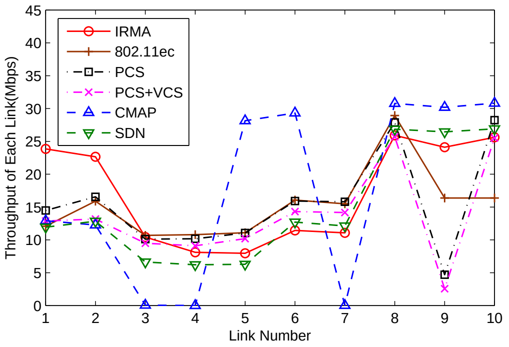


Fig. 16. The throughput of each link when $l = 2000bytes$ and $v_b = 48Mbits/s$.

Generally speaking, IRMA can outperform the other protocols in most cases, as concurrent transmissions exploited by this protocol lead to significant performance improvement, despite the overhead induced by signatures and control frames. Meanwhile, IRMA has a high fairness performance as it can avoid collisions successfully.

7 Related Work

Throughput improvement in wireless networks is an everlasting topic and attracts intense research interests. Prior work mainly falls in the following two categories:

7.1 Exploit Concurrent Transmissions

Physical carrier sense is widely used in wireless networks to avoid interferences, this mechanism is well known to have a very low performance. Brodsky and Morris [21] recently presented a theoretical model to analyze a two-sender carrier sense performance and concluded that this performance is close to optimal for radios with an adaptive bit rate. However, this model only considers situations with two contending senders, and it ignores MAC-level mechanisms such as the ACK and backoff. The result may not reflect real network conditions.

Throughput improvement through exploiting concurrent transmissions has shown much potential. MACA [22] exploits transmission concurrency by using a three-way handshake RTS/CTS/DATA mechanism. It disables the physical carrier sense to permit more concurrent transmissions. As there is no ACK in the protocol, it does not need to consider collisions at the transmitter side. MACAW [23] shows that ACK is essential in the wireless networking as the packet loss recovery can be done much faster in the MAC layer, and the loss always appears in the presence of intermittent noises which are likely presented in the wireless network. The 802.11 standard inherits the four-way handshake protocol of MACAW; therefore, it can avoid

ACK collisions at the transmitter side.

Recently, besides CMAP and SDN [2], [3], many approaches are proposed to maximize transmission concurrency for throughput improvement in wireless networks [4], [24]–[27]. C-MAC [4] disables the carrier sense to maximize the number of concurrent transmissions. RTSS/CTSS [26] lets nodes differentiate between interfering and non-interfering links through an offline training, which is just applicable to the line topology. MACA-P [27] tries to avoid the control frames' collisions by scheduling them properly. Based on the exchanges of RTS/CTS, it schedules multiple transmissions in parallel to increase concurrent transmissions as well as avoiding their collisions. However, besides introducing a significant protocol overhead for information exchanges, it does not differentiate the interference ranges of different transmission links.

7.2 Exploit Cross Correlation

Some other studies [5]–[7], [11]–[14], [20], [28]–[35] emphasize on a strategy that aims to recover the collided packets, instead of avoiding collisions. Especially, the technology of cross correlation has been leveraged to combat collisions.

Jamieson and Balakrishnan [28] propose a partial packet recovery mechanism to recover the whole packet via SoftPHY. The SoftPHY interface can analyze the symbol level information at the physical layer to identify the corrupted bits in the collided packet. ZigZag [11] works under 802.11 protocol that supports different transmission rates, and can deal with general collisions (i.e., collisions occur in the AP-Station model where all stations are within AP's transmission range) to combat hidden terminals, but it can not solve the exposed terminal problem. Symphony [12] encourages collision of packets among transmitters at APs and cooperatively decode all the packets by utilizing a Zigzag-like decoding process and the coordination information among APs, so as to improve the WLAN's throughput. However, it also does not exploit concurrency and solve the exposed terminal problem.

SIC [20] exploits a well-known capture effect technique, in which a strong interfering signal can be successfully received during collision, to recover an inferior strong signal if its SINR is above the threshold after decoding and subtracting the strongest one. Similar techniques are applied in [31]–[33] to solve the collisions under different scenarios. In [34], a MAC protocol is designed based on SIC and the full duplex communication technique to improve the throughput and fairness of of wireless networks. However, the stringent requirement of full duplex communications for supporting SIC seems hardly accomplishable with low cost in a near future. Moreover, the throughput improvement from SIC is reported not promising according to a recent work [35].

CSMA/CN [13] exploits the cross correlation to detect the collision notification information, which is sent by the receiver when detecting a collision and transmitted concurrently with the data packet, making the transmitter defer its packet transmission immediately. However, this protocol can only avoid packet collisions at the receiver side to solve the hidden terminal problem, it does not combat the exposed terminal problem nor exploits concurrent transmissions.

RTS/S-CTS [14] presents a symbol-level detection mechanism that can combat both the CTS collision problem and the remote hidden terminal problem, as the symbol sequences that carry useful information in the new S-CTS packet can be detected in very low SINR and SNR environments. It can improve the network throughput through avoiding collisions during packet transmissions. 802.11ec [5] also exploits the cross correlation to accomplish the control frames' transmissions. Comparing with the 802.11 standard, this protocol uses three kinds of primitives to convey the RTS, CTS and ACK information. As the primitives can tolerate strong interferences, and the duration of these sequences is much less than that of the corresponding packets, this protocol can improve the network throughput through both avoiding collisions and reducing the transmission overhead of the control frames. However, both protocols cannot combat the exposed terminal problem.

Comparing with the protocols in Section 7.1 which intend to solve the exposed terminal problem, and comparing with the protocols in Section 7.2 which exploit the cross correlation technology in other contexts, IRMA is the first protocol that exploits the cross correlation to combat the exposed terminal problem and maximize concurrent transmissions. Nodes in IRMA can take the real interference range into consideration when accessing their channels, and control frame's collisions at the transmitter side can be tolerated, leading to more concurrent transmissions and a higher throughput.

8 Conclusion and Future Work

In this paper, we identify that nodes in the 802.11 standard waste transmission opportunities in two scenarios and induce collisions in one scenario, we then propose IRMA to exploit transmission concurrency and avoid interferences in all the scenarios. IRMA employs new components to work collaboratively to increase network's transmission concurrency. We propose the signature detection method in the physical layer to combat control frame's collisions at the transmitter side. We propose a channel access scheme to permit concurrent transmissions while avoid data reception interferences. We show the feasibility of signature detection method via hardware experiments. We also show the significant throughput improvement over the two 802.11 standard and three recent protocols by ns-2.

In our current work, we just propose IRMA to solve the exposed terminal problem in wireless networks by leveraging the signature detection method. We consider this physical layer mechanism can also be exploited to benefit more wireless communication systems, such as cellular networks, RFID systems, and so on.

Acknowledgments

This work was supported in part by Hong Kong RGC GRF grant PolyU-521312, Hong Kong PolyU (A-PL84, 4-BCB6, M-N020), and NSFC grant No. 61272463.

References

[1] IEEE Computer Society. 802.11, "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," 2007.

[2] M. Vutukuru, K. Jamieson, and H. Balakrishnan, "Harnessing exposed terminals in wireless networks," in *Proc. USENIX NSDI*, 2008, pp. 59–72.

[3] L. B. Jiang and S. C. Liew, "Improving throughput and fairness by reducing exposed and hidden nodes in 802.11 networks," *IEEE Trans. Mobile Comput.*, vol. 7, no. 1, pp. 34–49, Jan. 2008.

[4] M. Sha, G. Xing, G. Zhou, S. Liu, and X. Wang, "C-MAC: Model-driven concurrent medium access control for wireless sensor networks," in *Proc. IEEE INFOCOM*, 2009, pp. 1845–1853.

[5] E. Magistretti, O. Gurewitz, and E. W. Knightly, "802.11ec: Collision avoidance without control messages," in *Proc. ACM MobiCom*, 2012, pp. 65–76.

[6] L. Wang, K. Wu, and M. Hamdi, "Combating hidden and exposed terminal problems in wireless networks," *IEEE Trans. Wireless Commun.*, vol. 11, no. 11, pp. 4204–4213, Nov. 2012.

[7] L. Wang and K. Wu, "Attached-RTS: Eliminating exposed terminal problem in wireless networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 7, pp. 1289 – 1299, Jul. 2012.

[8] IEEE Computer Society. 802.11n, "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment 5: Enhancements for higher throughput," 2009.

[9] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge University Press, 2005.

[10] K. Xu, M. Gerla, and S. Bae, "Effectiveness of RTS/CTS handshake in IEEE 802.11 based ad hoc networks," *Elsevier Ad Hoc Networks*, vol. 1, no. 1, pp. 107–123, Jul. 2003.

[11] S. Gollakota and D. Katabi, "Zigzag decoding: Combating hidden terminals in wireless networks," in *Proc. ACM SIGCOMM*, 2008, pp. 159–170.

[12] T. Bansal, B. Chen, P. Sinha, and K. Srinivasan, "Symphony: Cooperative packet recovery over the wired backbone in enterprise WLANs," in *Proc. ACM MobiCom*, 2013, pp. 351–362.

[13] S. Sen, R. R. Choudhury, and S. Nelakuditi, "CSMA/CN: Carrier sense multiple access with collision notification," in *Proc. ACM MobiCom*, 2010, pp. 25–36.

[14] T. Xiong, J. Zhang, J. Yao, and W. Lou, "Symbol-level detection: A new approach to silencing hidden terminals," in *Proc. IEEE ICNP*, 2012, pp. 1–10.

[15] X. Zhang and K. G. Shin, "E-MiLi: Energy-minimizing idle listening in wireless networks," in *Proc. ACM MobiCom*, 2011, pp. 205–216.

[16] T. Y. Lin and J. C. Houl, "Interplay of spatial reuse and SINR-determined data rates in CSMA/CA-based, multi-hop, multi-rate wireless networks," in *Proc. IEEE INFOCOM*, 2007, pp. 803–811.

[17] G. Holland, N. Vaidya, and P. Bahl, "A rate-adaptive MAC protocol for multi-hop wireless networks," in *Proc. ACM MobiCom*, 2001, pp. 236–251.

[18] T. Rappaport, *Wireless communications: Principles and practice*.

[19] ns-2, "The Network Simulator - Version 2." [Online]. Available: http://www.isi.edu/nsnam/ns/

[20] D. Halperin, T. Anderson, and D. Wetherall, "Taking the sting out of carrier sense: Interference cancellation for wireless LANs," in *Proc. ACM MobiCom*, 2008, pp. 339–350.

[21] M. Z. Brodsky and R. T. Morris, "In defense of wireless carrier sense," in *Proc. ACM SIGCOMM*, 2009, pp. 147–158.

[22] P. Karn, "MACA - a new channel access method for packet radio," in *Proc. the ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, 1990, pp. 134–140.

[23] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A media access protocol for wireless LANs," in *Proc. ACM SIGCOMM*, 1994, pp. 212–225.

[24] S. B. Eisenman and A. T. Campbell, "E-CSMA: Supporting enhanced CSMA performance in experimental sensor networks using per-neighbor transmission probability thresholds," in *Proc. IEEE INFOCOM*, 2007, pp. 1208–1216.

[25] M. Cesana, D. Maniezao, and M. Gerla, "Interference aware (IA) MAC: An enhancement to IEEE 802.11b DCF," in *Proc. IEEE VTC*, 2003, pp. 2799–2803.

[26] K. Mittal and E. Belding, "RTSS/CTSS: Mitigation of exposed terminals in static 802.11-based mesh networks," in *Proc. IEEE WiMesh*, 2006, pp. 3–12.

[27] A. Acharya, A. Misra, and S. Bansal, "Design and analysis of a cooperative medium access scheme for wireless mesh networks," in *Proc. ICST BROADNETS*, 2004, pp. 621–631.

[28] K. Jamieson and H. Balakrishnan, "PPR: Partial packet recovery for wireless networks," in *Proc. ACM SIGCOMM*, 2007, pp. 409–420.

[29] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "XORs in the air: Practical wireless network coding," in *Proc. ACM SIGCOMM*, 2006, pp. 243–254.

[30] S. Katti, D. Katabi, H. Balakrishnan, and M. Medard, "Symbol-level network coding for wireless mesh networks," in *Proc. ACM SIGCOMM*, 2008, pp. 401–412.

[31] C. Qin, N. Santhapuri, S. Sen, and S. Nelakuditi, "Known interference cancellation: Resolving collisions due to repeated transmissions," in *Proc. IEEE WiMesh*, 2010, pp. 1–6.

[32] L. Li, K. Tan, H. Viswanathan, Y. Xu, and Y. Yang, "Retransmission ≠ Repeat: Simple retransmission permutation can resolve overlapping channel collisions," in *Proc. ACM MobiCom*, 2010, pp. 281–292.

[33] T. Li, M. K. Han, A. Bhartia, L. Qiu, and E. Rozner, "CRMA: Collision-resistant multiple access," in *Proc. ACM MobiCom*, 2011, pp. 61–72.

[34] N. Singh, D. Gunawardena, A. Proutiere, B. Radunovie, H. V. Balan, and P. Key, "Efficient and fair MAC for wireless networks with self-interference cancellation," in *Proc. IEEE WiOpt*, 2011, pp. 94–101.

[35] S. Sen, N. Santhapuri, R. Choudhury, and S. Nelakuditi, "Successive interference cancellation: A back-of-the-envelope perspective," in *Proc. ACM HotNets*, 2010.