

Self-learning Collision Avoidance for Wireless Networks

Chun-cheng Chen, Eunsoo Seo, Hwangnam Kim, and Haiyun Luo

Dept. of Computer Science, UIUC

Email: {chen35,eseo2,hkim27}@uiuc.edu, haiyun@cs.uiuc.edu

Abstract—The limited number of orthogonal channels and the autonomous installations of hotspots and home wireless networks often leave neighboring 802.11 basic service sets (BSS's) operating on the same or overlapping channels, therefore interfering with each other. However, the 802.11 MAC does not work well in resolving inter-BSS interferences due to the well-known hidden/exposed receiver problem, which has been haunting in the research community for more than a decade. In this paper we propose SELECT, an effective and efficient self-learning collision avoidance strategy to address the open hidden/exposed receiver problem in wireless networks. SELECT is based on the observation that carrier sense with received signal strength (RSS) measurements at the sender and the receiver are strongly correlated. A SELECT-enabled sender exploits such correlation using automated on-line learning algorithm, and makes informed judgment of the channel availability at the intended receiver. SELECT achieves collision avoidance at packet-level time granularity, involves zero communication overhead, requires no hardware support beyond what is available in off-the-shelf 802.11 devices, and easily integrates with the 802.11 DCF. Our evaluation in both analysis and simulations show that SELECT addresses the hidden/exposed receiver problem well. In typical hidden/exposed receiver scenarios SELECT improves the throughput by up to 140% and channel access success ratio by up to 302%, while almost completely eliminating contention-induced data packet drops.

I. INTRODUCTION

802.11¹ wireless LANs usually rely on careful channel assignment to avoid the interferences between neighboring basic service sets (BSS's). However, because there is only a very limited number of orthogonal channels (e.g., 3 for 802.11b/g and 12 for 802.11a in US) and because the interference range of an 802.11 transceiver is often long compared with the communication range, the clients and access point of a BSS are often interfered by the clients and access points in neighboring BSS's operating on the same or overlapping channels. This problem is further aggravated by the widespread, autonomous installations of 802.11 home networks and hotspots. Recently published data on metropolitan area 802.11 coverage [1], [2] show that more than 40% of the access points are operating on channel 6. In Boston, a maximum number of 85 access points were detected in interference range [2], [3], which leads to at least 28 access points directly interfering with each other given that more than 90% access points are 802.11b/g. In April 2005, a wardriving survey of one of the authors' residential subdivision, where 87 single houses are located, showed that

¹We abuse "802.11" to denote the IEEE 802.11 family.

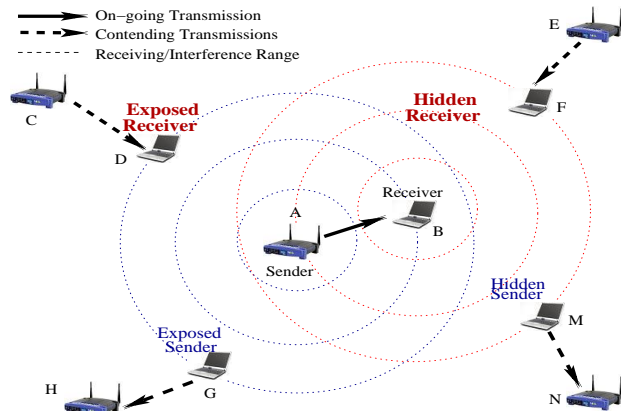


Fig. 1. Hidden/exposed terminal problem in 802.11 networks. Sender G and receiver D are *exposed* in A's on-going transmission, while sender M and receiver F are *hidden* from A's transmission.

an average number of 5.6 access points per house are detected active on channel 6.

As a result of the inter-BSS interferences, clients located around the boundary of a BSS may suffer from the well-known open hidden/exposed terminal problem [4], [5], [6]. We illustrate the problem in Figure 1². With 802.11 DCF (Distributed Coordination Function), sender A and receiver B exchange Request-to-Send (RTS) and Clear-to-Send (CTS) control messages to notify potential competitors and protect the following data packet transmission. As a result, exposed sender G will defer its transmission on reception of A's RTS message and/or its physical carrier sense of A→B transmission. Hidden sender M will also defer its transmission on reception of receiver B's CTS message³. Therefore, 802.11's RTS/CTS handshake handles the *hidden/exposed sender problem* well.

However, neither 802.11 DCF nor any other existing solution handles the *hidden/exposed receiver problem* within the 802.11's framework of single-channel operation. In Figure 1, receiver D is exposed while receiver F is hidden from the active sender A. They have to remain idle until receiver B receives the data packet and sender A receives the acknowledgment. However, neither sender C nor E is aware of the

²We use circle to represent the receiving/interference range in Figure 1. However, we do not assume circular receiving/interference area. It can be of any dynamic shape in reality.

³FAMA [7] proposes an elegant improvement with *long dominating CTS* to handle hidden senders that miss the receiver's CTS and later interfere the data packet reception.

on-going A→B transmission. Hence C or E may initiate RTS requests to their intended receiver D or F in the middle of an A→B transmission. Sender C’s RTS message will collide with the sender A’s signal at exposed receiver D⁴. Hidden receiver F cannot respond to E’s RTS, because F has received a CTS from receiver B and must remain idle until the A→B transmission finishes - a mechanism named “virtual carrier sense” in 802.11. After the timer for CTS expires, sender C (or E) doubles its contention window size and engages in another round of random backoff before it tries to send the RTS again. Following the same reasoning the situation will be worse if RTS/CTS are disabled and two-way DATA/ACK handshake is employed, which is common in 802.11 WLANs.

The consequences of the hidden/exposed receiver problem are severe. First, after a number of retries, the sender drops the head-of-line data packet, resulting in contention-induced packet loss. Second, unsuccessful RTS attempts might mislead the sender to the conclusion that the intended receiver is unavailable or the channel quality at the receiver side is low. In the former case a false link breakage is triggered, resulting in routing instability. In the latter case the sender reduces the data rate which aggravates the channel contention. Third, unsuccessful RTS attempts inflate the sender’s 802.11 contention window quickly according to the exponential backoff algorithm and causes unfair channel access. Fourth, repeated RTS attempts prevent the sender’s neighbors from transmitting, lowering the shared channel utilization. Finally, if the hidden/exposed receiver problem happens in wireless LANs, it will persist until the clients move and the contention relation changes.

Although the general hidden/exposed terminal problem has attracted a lot of attention for more than a decade, the vast majority of existing work has been devoted to the hidden/exposed sender problem only⁵. While many research efforts have been invested in mitigating the effects of hidden/exposed receiver (see Section II for a comprehensive review), the problem itself remains open. *The fundamental challenge lies in the lack of effective and efficient mechanisms to exchange channel availability information between the sender and the receiver, at packet level time granularity, before a channel access attempt is made.*

In this paper we propose SELECT, a self-learning collision avoidance mechanism, to address hidden/exposed receiver problem in wireless networks. SELECT is based on the observation that *carrier sense with received signal strength (RSS) measurements at the sender and the receiver are strongly correlated*. This correlation, once established, could be used to provide the sender with information regarding the receiver’s channel status and *vice versa*. Once a hidden/exposed receiver is detected the sender can employ appropriate mechanisms to eliminate the negative impacts.

We first use real-radio measurement data to demonstrate the correlation between sender and receiver carrier sense RSS

in a multihop wireless network, which serve as our motivation for this work (Section III). We then describe SELECT which exploits such correlation and *directly* characterizes the relationship between a sender’s RSS and its channel access success ratios. Practical issues such as computation constraint, storage constraint, RSS measurement noise, temporal dynamics of wireless signal propagation and closed-loop system ossification are also addressed through careful SELECT design (Section IV). We then analyze SELECT-enhanced 802.11 DCF in Section V, and evaluate the performance in Section VI through intensive simulations in terms of throughput, channel access success ratio, packet losses and fairness. Our results show that in typical hidden/exposed receiver scenarios SELECT significantly increases throughput by up to 140% and channel access success ratio by up to 302%. It almost completely eliminates contention-induced packet drops. We finally conclude with future work in Section VII.

SELECT is a sender-side only collision avoidance mechanism. It addresses hidden/exposed receiver problem at the packet-level time granularity and involves zero communication overhead. It does not rely on special hardware support (e.g., multiple channel communication capability as proposed in BAPU [8]). Instead, SELECT only uses instantaneous RSS measurement, a standardized sensory function built-in in most wireless transceivers for carrier sense (e.g., off-the-shelf 802.11 wireless devices). Furthermore, SELECT does not rest on any analytical model of wireless signal propagation, which is affected by many factors and very difficult to analytically appraise [9], [10]. Finally, SELECT complies with 802.11’s PHY and MAC specifications and is completely compatible with other non-SELECT 802.11 devices. Only small non-disruptive enhancements to collision avoidance are required to incorporate SELECT into 802.11 DCF, as we will further elaborate in Section IV.

II. RELATED WORK

Medium access control can be either contention based or schedule based. Contention based schemes are usually preferred in data networks because they achieve higher statistical gain, are easier to implement, and are robust to synchronization errors. Collisions have to be resolved in contention based medium access control. Different from widely adopted collision detection in wired network (e.g., IEEE 802.3 Ethernet), collision avoidance is usually adopted for wireless medium access control since it simplifies the wireless transceiver.

IEEE 802.11 [11] medium access control, predominantly Distributed Coordination Function DCF, is probably the most popular CSMA/CA MAC. 802.11 was designed for infrastructure mode, where nodes in a Basic Service Set (cell) can be *at most two hops* away from each other and *communicate only with the centralized access point*. 802.11 DCF handles hidden/exposed senders well but does not address the hidden/exposed receiver problem, since the latter usually does *not* exist in a network operating in infrastructure mode. However, as shown in [4], [5], [6] the hidden/exposed receiver problem manifests itself in multihop 802.11 networks with nodes

⁴Unless the signal from sender C is sufficiently strong to capture.

⁵Except for BAPU [8] using *dual-channel* collision avoidance.

distributed three or more hops from each other. MACAW [12] and FAMA [7] are early proposals on CSMA/CA wireless MAC. They handle hidden/exposed sender problem even better than 802.11, but leave the hidden/exposed receiver problem open.

BAPU [8] addresses the hidden/exposed receiver problem, but it requires two channels and uses a dedicated control channel for signaling. Recently several multi-channel variations of 802.11 medium access control are also proposed, e.g., SSCH [13] and MMAC [14], but their goal is to increase network capacity, not to handle hidden/exposed receiver. The design of single-transceiver multiple channel involves extra latency for channel synchronization and requires time-synchronization hardware [13], or introduces significant modification on 802.11 [14]. Furthermore, multiple unlicensed 802.11 channels are not always available, as the cases in Japan and India [15]. SELECT addresses the hidden/exposed receiver problem using only one single shared channel. In fact, it does not even involve any communication overhead.

Another interesting option is to use receiver-initiated collision avoidance [16], [17]. It solves the literal hidden/exposed “receiver” problem, since the receiver initiates channel access. However, the hidden/exposed sender problem emerges since a hidden/exposed sender may not be able to respond to the receiver’s poll. SELECT keeps the existing collision avoidance mechanisms that are mature in dealing with hidden/exposed sender, and designs additional mechanism to handle hidden/exposed receiver.

Optimal MAC carrier sense threshold has been studied to maximize the channel reuse [18], [19]. These analysis, however, do not help address hidden/exposed receiver problem. Furthermore, as we will show in Section III-C, simply comparing RSS with a single “adaptive” carrier sense threshold loses a lot of information about the relationship between RSS and the success ratio of channel access, leading to either conservative channel reuse or collisions due to hidden/exposed receivers.

Recent measurements [10] using existing 802.11 devices showed that signal-to-interference-noise ratio (SINR), *measured as an average over many packets*, may not be a good predictive tool for the successful delivery of a packet. It seems to be contradictory to our approach at the first glance. Note that we do not use SINR of a packet transmission to predict whether the packet will be successfully received. Instead, we seek to correlate the RSS, *measured before a sender transmits a packet*, with the packet delivery success ratio. In spirit our approach is in line with the *physical carrier sense* defined as a mandatory mechanism in the 802.11 standard (as well as other CSMA-based MAC designs), which has been implemented in all 802.11 interfaces and proven useful.

III. MOTIVATION

SELECT is grounded on the observation that the correlation between sender-side and receiver-side carrier sense RSS can be exploited for collision avoidance. In this section, we use measurement data from our multihop wireless testbed to verify

the correlation and motivate two design options for further examination in the next section.

A. Testbed setup

We choose Crossbow MICA2 sensor motes with ChipCon model CC1000 [20] single-chip RF transceivers to form our wireless network testbed. Each MICA2 mote is equipped with an 8-bit 4MHz microcontroller running a microthread operating system, called TinyOS [21], from its internal flash memory. The memory size available at each node is limited: 128KB of program memory and 4KB of data memory. In our testbed CC1000 radio works in the 433MHz frequency band, and achieves a maximum data rate of 19.2kbps. Depending on the power setting the communication range varies from 1 to 20 meters in our lab.

The primary reason we use MICA2 motes in our experiments is the programmability of its CC1000 radio. The TinyOS radio stack for CC1000 transceiver is open-source, and allows byte-level transmission/reception control. Therefore, we can implement our SELECT collision avoidance (as an ongoing work), and test their performances. Off-the-shelf 802.11 devices do not expose the interface for carrier sense. They usually have collision avoidance built in the firmware, and the source codes are proprietary. Although CC1000 is far different from 802.11 PHY specifications, their core functions for the implementation of 802.11’s DCF MAC protocol are similar (e.g., single-channel asynchronous communication and carrier sense with RSS). Besides, the computation and storage constraints of the MICA2 platform are comparable to the constraints (e.g., memory and computation constraints) when implementing SELECT in 802.11 devices in the future.

B. Sender/receiver RSS correlation

Every mote in our testbed runs a stripped version of 802.11 DCF, derived from [22], with RTS-CTS-DATA-ACK, physical and virtual carrier sense, and random backoff. Physical carrier sense is implemented as a sampling of the ADC port. The ADC reads the received signal from CC1000’s analog pin and converts it into a 10-bit voltage reading. The voltage reading can be further mapped into RSS in dBm according to [20].

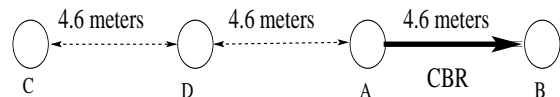


Fig. 2. Mote D is an exposed terminal.

We first study a simple 3-hop topology with 4 motes, as shown in Figure 2. All 4 motes are placed 0.5 meter above the floor. Under the specific power setting mote A and B cannot communicate beyond 4.6 meters. We configure mote A to transmit 65-byte packets as fast as the channel allows, and log the RSS at motes C and D as fast as possible over a 5-second interval. Note that mote D is potentially an exposed receiver if mote C initiates transmission request. Our purpose in this experiment is to construct a scenario of exposed receiver, similar to the one shown in Figure 1, and study the relationship

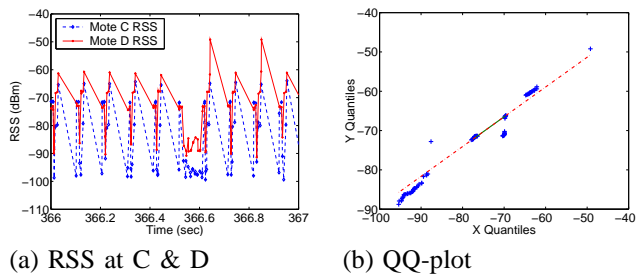


Fig. 3. RSS at mote C and D while A is transmitting to B

between the RSS at a potential sender (C) and the RSS at an exposed receiver (D).

Figure 3(a) shows the RSS samples over a 1-second period. The calculated correlation coefficient between these two sequences of RSS samples is as high as 0.878. We also show the QQ-plot of these samples over the a 5-second interval in Figure 3(b). From the QQ-plot, we can see an excellent fit of the straight line, indicating that the two sequences of RSS samples at motes C and D comply with the same distribution. Similar results are also obtained in the scenario of hidden receiver, as we reverse the flow direction (i.e., B→A).

These results are not surprising. Although the wireless signal attenuates quickly as it travels in the air, the hidden/exposed receivers and their senders must locate within an area that is close enough to an on-going transmission. Note that this area could be of any shape and could change over time due to multipath fading. Although the on-going transmission (e.g., A→B or B→A) may not be strong enough to be decoded at all nodes in the area (e.g, node C), it is strong enough to dominate their RSS's in the presence of environmental noises and other interferences caused by transmissions further away. Therefore the RSS measurements in the interference area will exhibit strong correlations, reflecting their receptions of the wireless signals from the same sources of interferences.

C. RSS v.s. success ratio

To facilitate our discussions of the SELECT design in the next section, we perform another experiment with 8 MICA2 motes placed in the topology shown in Figure 4. We configure four transmissions: A→B, C→D, E→F, and G→H, where senders transmit 65-byte packets as fast as the channel allows. Note that receiver B is an exposed receiver when C→D is active, and a hidden receiver when G→H is active. Transmission E→F serves as additional interferences. We log the carrier sense RSS at sender A before it initiates channel access (with RTS), and the success (reception of CTS) or failure (CTS timeout) of the attempt. We define the success ratio as the number of CTS messages received by a sender over the total number of RTS attempts. Figure 5 shows the RSS (aggregated over 0.5dBm intervals) versus the success ratio at mote A during a measurement time period of 75 seconds. We show the stability of such a mapping at two specific RSS readings during an experiment that lasted for 2000 seconds in Figure 6. The variation of the success ratio is below 20% with a sampling interval of 50 seconds. This results holds as long

as the interference signal dominates the RSS measurements, which is true in the cases of hidden/exposed terminals.

Figure 5 and 6 clearly reveal the impacts of hidden/exposed receiver on the success ratio of the sender's channel access. First, a carrier sense with low RSS at the sender (A) does not necessarily mean the channel is available at the receiver (B). In fact, all RSS's in Figure 5 are below the default carrier sense threshold, but the success ratio of RTS attempts can be as low as 14.3% when sender A's RSS is around -93.68dBm, a scenario when the intended receiver B is likely hidden/exposed. Second, the relationship between the sender's RSS and the corresponding channel access success ratio is not monotonic. If a sender only contends for the channel when its RSS is lower than a carrier sense threshold, then depending on the threshold setting, the sender will either suffer from serious channel access failure even though the RSS is relatively low (-93.68dBm), or lose the opportunity to successfully grab the channel when the carrier sense RSS is relatively large (-91.37dBm) but the success ratio is actually high. Finally, the mapping between the RSS and the channel access success ratio can help to significantly improve the channel access success ratio, if a sender deliberately chooses not to contend when the current RSS and past history suggest a low success probability.

IV. SELF-LEARNING COLLISION AVOIDANCE

The analysis of the experiment data presented in the above section also shed light on potential solutions. Intuitively one could leverage the strong correlation between the sender's RSS and the receiver's RSS, as shown in section III-B, and have the sender estimate the RSS and channel availability at the receiver. We did not take this approach, however, for the following two reasons. First, establishing the RSS correlation requires the receiver to feedback its RSS in a timely manner. This feedback will inevitably involve some signaling between the sender and the receiver, which complicates the MAC and/or PHY layers. Second, estimating receiver's RSS only detects an exposed receiver. The RSS at a hidden receiver could actually be low.

A receiver fails to respond to a sender's RTS for two reasons. At an exposed receiver the RTS collides with an on-going transmission, while at a hidden receiver the channel is already reserved by the CTS of the on-going transmission. The consequences of these two scenarios are the same from the sender's perspective, and it does not have to differentiate these two. Therefore, exact estimate of receiver's RSS is neither sufficient (when the receiver is hidden) nor necessary (since the sender only needs to know that the RSS at the receiver is low enough). Instead, the sender can *directly* establish the mapping between its RSS and the success ratio of its channel access attempts, as suggested in section III-C and Figure 5. Since the RTS success is signified by the reception of the CTS while the RTS failure is signified by the CTS timeout, no additional signaling between the sender and the receiver is necessary.

In the rest of this section we first present the details of maintaining the mapping between the RSS and the channel

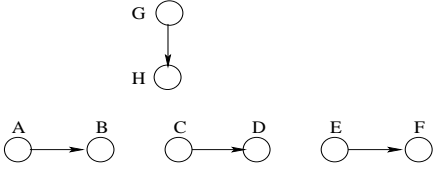


Fig. 4. Mote B is a hidden/exposed receiver

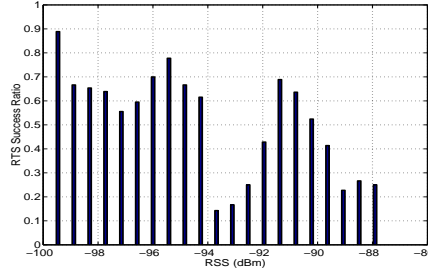


Fig. 5. RSS v.s. RTS success ratio at mote A

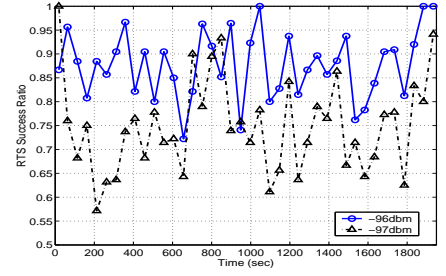


Fig. 6. RSS v.s. RTS success ratio over time

access success ratio (section IV-A and IV-B). We then study the appropriate integration of SELECT collision avoidance with 802.11 DCF (section IV-C).

A. RSS-SR mapping maintenance

Hidden/exposed receivers complicate the relationship between the sender's RSS and the channel access success ratio. This complexity and the requirement for adaptability invalidate our first thought of representing the mapping (e.g., Figure 5) in analytical forms (e.g., least squares fitting with a high-degree polynomial). Given the fact that most 802.11 NICs have at least 128KB SRAM embedded, we choose to maintain the histogram directly, trading-off a relatively small storage (in hundreds of bytes) for lower design complexity and computation overhead.

Specifically, we divide the range of the RSS, $[RSS^{min}, CS^{thred}]$, into N intervals $[RSS_i^{min}, RSS_i^{max}]$ ($i = 1, \dots, N$), where RSS^{min} is set to the estimated noise level or measured minimum RSS (e.g., -100dBm in section III-C) and CS^{thred} is the default carrier sense threshold. We consider $RSS < CS^{thred}$ only since no channel access attempt will be made if the channel is not even available at the sender. We divide intervals evenly for efficient lookup. For each RSS interval I_i three state variables are maintained: the number of successful channel access attempts S_i , the number of failed channel access attempts F_i , and the timestamp T_i^{upd} as the last time S_i and F_i are updated. Essentially the histogram is maintained as a simple one-dimensional array with N elements of $I_i = \langle S_i, F_i, T_i^{upd} \rangle$ tuples. The size of the array N can be based on the available memory. Since close RSS measurements are aggregated and represented by a single entry I_i , it also helps suppress RSS measurement errors. We use $N = 300$ in our simulations.

Figure 7 shows the pseudo-codes for a sender to update the mapping when a new channel access attempt is made and the success/failure is determined. For the mapping to adapt to the current operating environment, outdated records have to be removed. Therefore, both S_i and F_i have to be constrained as the number of successful/failed channel access attempts within a recent time window T_{win} . Note that the proper setting of T_{win} depends on the dynamics of the environments, such as the traffic pattern, network topology, and signal propagation. If we denote the new record as $\langle rss, sf \rangle$ where $sf = 1$ when the channel access attempt succeeds and $sf = 0$ when

```

// Input - rss: carrier sense RSS
// - sf: 1 if succeed, 0 if fail, -1 if no new record
Upd_RSS_SR(rss, sf)
1.  $i = \lfloor (rss - RSS^{min}) / I_{width} \rfloor$ ; // Locate element  $I_i$ 
   //  $I_{width} = (CS^{thred} - RSS^{min}) / N$ 
2.  $\alpha = 1 - (t - T_i^{upd}) / T_{win}$ ; // Adaptive aging factor
3. if ( $\alpha < 0$ ) then  $\alpha = 0$ ;
   // if ( $t - T_i^{upd} > T_{win}$ ) clear  $S_i$  and  $F_i$ 
4. if ( $sf == 1$ )
   then  $S = 1, F = 0$ ; // channel access succeeds
   else if ( $sf == 0$ )
   then  $S = 0, F = 1$ ; // channel access fails
   else  $S = 0, F = 0$ ; // no new record
5.  $S_i = \alpha \cdot S_i + S$ ; // Update # of successful attempts
6.  $F_i = \alpha \cdot F_i + F$ ; // Update # of failed attempts
7.  $T_i^{upd} = t$ ; // Update timestamp

```

Fig. 7. A sender update the mapping with new record $\{rss, sf\}$. $O(1)$ computation overhead

the channel access attempt fails, the standard approximation to the windowed sum is to apply an aging factor α on S_i and F_i periodically: $S_i = \alpha S_i + S$ and $F_i = \alpha F_i + F$, where $S = sf$ and $F = \sim S$. The update period is set to $T_{period} \ll T_{win}$ and $\alpha = 1 - T_{period} / T_{win}$. However, senders obtain records aperiodically. Moreover, updating all N S_i 's and F_i 's leads to $O(N)$ per-update overhead. We address these two problems by adapting the aging factor α based on the time from last update T_i^{upd} :

$$\alpha = \begin{cases} 1 - \frac{t - T_i^{upd}}{T_{win}} & \text{if } t - T_i^{upd} < T_{win} \\ 0 & \text{Otherwise} \end{cases}$$

The above results can be derived by simulating a periodic update process with infinitely small period. With the dynamic aging factor the mapping will only be updated *on-demand* after the sender obtains a new record or before the sender queries the mapping for the channel access success ratio (see section IV-B). Per-update complexity is also reduced to $O(1)$ to guarantee quick return. *The cost we pay is the maintenance of the per-entry timestamp T_i^{upd} of the last update, another tradeoff of storage for complexity.*

B. RSS-SR mapping lookup

With the RSS-SR mapping maintained, a sender could query the mapping to obtain the historical success ratio of channel

access attempts under certain carrier sense RSS. Figure 8 shows the pseudo-codes. When a lookup request is received the mapping is firstly updated to remove outdated records (those records that fall out of the T_{win} window). We then locate the newly updated interval I_i corresponding to the given RSS, and examine if there are enough records established. The historical success ratio is returned if the total number of success and failures is above certain threshold. Otherwise a 100% success ratio is returned. The reason is that by default an RSS smaller than the carrier sense threshold CS^{thred} is interpreted as idle channel unless there are enough negative results to overturn this conclusion.

```

// Input - rss: carrier sense RSS
// Output: historical channel access success ratio
RSS_SR_LookUp(rss)
1. if ( $rss \geq CS^{thred}$ ) return 0%;
   // Channel is busy at sender
2. Upd_RSS_SR(rss, -1); // Remove outdated records
3.  $i = \lfloor (rss - RSS^{min}) / I_{width} \rfloor$ ; // Locate element  $I_i$ 
   //  $I_{width} = (CS^{thred} - RSS^{min}) / N$ 
4. if ( $S_i + F_i > Min\_Num\_Rec$ ) then // Enough records
   return  $S_i / (S_i + F_i)$ ; // Success ratio
   else // No enough records
   return 100%; // Channel is idle by default

```

Fig. 8. A sender queries for historical success ratio of channel access under certain rss . $O(1)$ computation overhead

Applying a sliding time window on the histogram to remove out-dated records is critical for the system adaptability. In open-loop, a MAC module outputs records of successes and failures to drive the SELECT module to generate an accurate RSS-SR mapping. When the loop is closed, however, the MAC module will generally avoid attempting to access the channel under those RSS's classified by SELECT as signs of low channel access success ratio. Therefore, no further records will be obtained once an RSS interval is mapped to low channel access success ratio, and the closed-loop system stagnates at those RSS intervals regardless of potential operating scenario changes. By limiting the valid records within a recent time window and removing the impact of old records, the closed-loop system can gradually “recover” so that the MAC module will gradually start to try channel access under those RSS's that are previously believed to lead to failures.

C. Integration with 802.11 DCF

We have instantiated two interfaces in SELECT for interactions with medium access control: `Upd_RSS_SR` and `RSS_SR_LookUp`. It is straightforward for the use of the first interface: once a MAC module accesses the channel and the result is determined, it calls `Upd_RSS_SR` to update the RSS-SR mapping. In this section, we study in the specific context of 802.11 DCF how a sender can take advantage of better estimation of channel access success ratio provided by `RSS_SR_LookUp`. Before we discuss design options we first briefly review the current collision avoidance in 802.11 DCF.

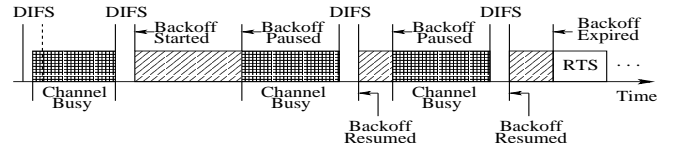


Fig. 9. 802.11 DCF collision avoidance

As shown in Figure 9, an 802.11 DCF sender with a data packet to transmit first monitors the channel for a time period called DIFS: DCF interframe space. If the channel is idle for DIFS without interruption the sender chooses a random number R uniformly from the interval $(0, CW)$, and starts a backoff timer that expires after $(R \cdot aSlotTime)$. CW is the current contention window size, and $aSlotTime$ is one slot time, a constant defined by the 802.11 PHY layer [11]. The backoff timer will be paused when the channel becomes busy, and will be resumed after the channel has been idle for DIFS without interruption. The sender will contend for the wireless channel (with RTS) when the backoff timer expires. If the channel access attempt fails (CTS timeout), the sender doubles its backoff time window CW (a.k.a. exponential backoff), waits for another DIFS, and starts another round of random backoff.

One way to incorporate SELECT into the above process of collision avoidance is to measure the RSS and call `RSS_SR_LookUp` after the backoff timer expires and before the sender sends out the RTS (with four-way handshake) or the DATA (with two-way handshake). If `RSS_SR_LookUp` returns a high success ratio, say above 50%, the sender proceeds to contend for channel. Otherwise the sender skips the RTS, and executes the 802.11 DCF protocol as if the channel access had failed (doubling CW , waiting for DIFS, and starting a new round of random backoff). This way, it saves the sender the transmission of the RTS and the timeout for the CTS. It also solves the problem of *false blocking*, as the neighbors of the sender will not have to remain idle and wait for a data transmission that actually does not exist.

However, our simulation results show that above approach only improves the performance marginally ($\sim 10\%$). The reason is that above approach does not correct the flaws in applying 802.11 DCF collision avoidance to resolve contentions from other BSS's. The exponential random backoff was designed to de-synchronize the channel access among multiple senders within interference range of each other. Collisions happen *only* as a result of two or more senders choosing the same random number, suggesting that the current contention window is not large enough. Therefore doubling the contention window size on collision is appropriate to accommodate the increased tension among competing senders. This mechanism works fine in a single BSS. However, in case of hidden/exposed receiver due to interferences from other BSS's, the collisions may result from the sender's lack of information on channel status at the intended receiver, not the increasing number of competing senders. Doubling contention window size in these scenarios is inappropriate and only

causes unfair channel access or even starved flows.

SELECT's `RSS_SR_LookUp` can help the sender eliminate collisions due to the lack of information regarding receiver's channel status. Specifically, by querying `RSS_SR_LookUp` a sender can *suspend its backoff timer whenever its current RSS suggests low channel access success ratio, and resume the backoff timer whenever its RSS stays at those levels suggesting high channel access success ratio* for DIFS without interruption. By the time the backoff timer expires the channel must be available at both the sender and receiver, with high probability⁶. Collisions can only be caused by two senders picking up the same random backoff, in which case the original exponential backoff algorithm fits in perfectly.

By suspending the backoff timer whenever the channel access success ratio is low, the definition of "Channel Busy", as shown in Figure 9, is extended to representing busy channel at either the sender or the receiver. The channel is now considered "Busy" if either sender's channel is unavailable (due to physical or virtual carrier sense failure) or the channel at receiver is unlikely to be available (signified by low channel access success ratio). With this simple extension of "Channel Busy" the original state flow (Figure 9) still applies. Therefore the change to the 802.11 state machine is minimal since no new state is introduced.

V. MODELING AND ANALYSIS

In this section, we develop two analytical models. One describes the throughput of 802.11 DCF MAC that is vulnerable to the hidden/exposed receiver problem, and the other describes the throughput of SELECT-enhanced 802.11 DCF.

A. Assumptions

We first enumerate the following two assumptions behind our models. (1) We assume that the channel availability at the receiver is only affected by other interfering data transmissions, not by noises, fading, and other wireless dynamics. Therefore receiver side channel unavailability period (i.e., interference period), I_r , follows the distribution of one successful transmission time, T_{succ} . We denote the probability of hidden/exposed receiver problem as P_r , i.e., the probability that when the channel is available at sender but not available at the receiver. (2) Following the convention of existing 802.11 DCF analysis, we assume the future behavior at each sender is independent of the past. Thus, all the idle periods, $idle_i$, consisting of consecutive idle slots between two consecutive collisions or before a successful transmission, are independent of each other. Furthermore, all collisions and successful transmissions are independent of each other. We use 802.11b DCF parameters in our evaluation, as specified in Table I.

B. Throughput models

In this section, we first analyze how the hidden/exposed receiver problem affects the throughput in 802.11 DCF-based wireless networks, and then describe to what extent the throughput in the networks can be enhanced with SELECT.

⁶Considering the sliding window mechanism introduced in section IV-B for closed-loop system adaptability.

TABLE I
IEEE 802.11b DCF PARAMETERS

Basic Bit Rate	1 Mbps	MAC data header	224 bits
Channel Bit Rate	11 Mbps	ACK size	112 bits
Slot Time (T_{slot})	20 μ sec	RTS size	160 bits
SIFS	10 μ sec	CTS size	112 bits
DIFS	50 μ sec		
CW_{min}	32	Phy preamble	144 bits
CW_{max}	1024	Phy header	48 bits

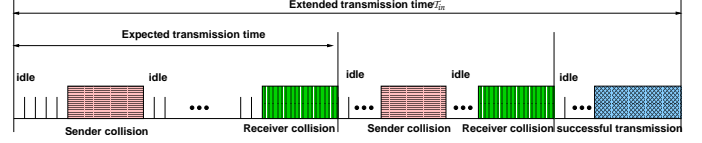


Fig. 10. Transmission time affected by interference

1) *802.11 DCF throughput with interference at receiver:* Figure 10 illustrates the time to successfully transmit one data frame and how interferences at the receiver extends the transmission time. We use the following notations for our analysis:

- μ_{in} is the normalized throughput with interference at the receiver;
- T_{in} is the transmission time taken to successfully transmit one data frame with interference at receiver;
- m is the payload size of 802.11 DCF MAC data frame;
- Nt is the number of collisions due to interferences at sender;
- Nr is the number of collisions due to interferences at receiver;
- $idle_i$ is the sequence of consecutive idle slots before i -th collisions;
- $idle_{succ}$ is the sequence of consecutive idle slots just prior to the successful frame transmission;
- I_r is the interference period at receiver;
- T_{coll_i} is i -th collision period;
- T_{succ} is the successful frame transmission time;
- M is the number of sender's neighboring nodes that cause collisions at sender;
- τ is the transmission probability at sender;

The network throughput, μ_{in} , with the interferences at a hidden/exposed receiver can be formulated as:

$$\mu_{in} = \frac{\bar{m}}{\bar{T}_{in}},$$

where \bar{m} is the average payload length and \bar{T}_{in} is the average transmission time.

According to Figure 10, T_{in} is constructed in the following way;

$$T_{in} = \sum_{k=0}^{Nr} \left\{ \sum_{i=0}^{Nt} [idle_i * T_{slot} + T_{coll_i}] + I_r \right\} + idle_{succ} * T_{slot} + T_{succ}.$$

The expectation of T_{in} is therefore $E(T_{in}) =$

$$E \left[\sum_{k=0}^{N_r} \left\{ \sum_{i=0}^{N_t} [idle_i * T_{slot} + T_{coll_i}] + I_r \right\} + idle_{succ} * T_{slot} + T_{succ} \right]$$

Based on independence assumptions the expected transmission time is simplified as:

$$E[T_{in}] = E[N_r] * (E[N_t] * E[idle] * T_{slot} + E[N_t] * E[T_{coll}] + E[I_r]) + E[idle] * T_{slot} + E[T_{succ}]$$

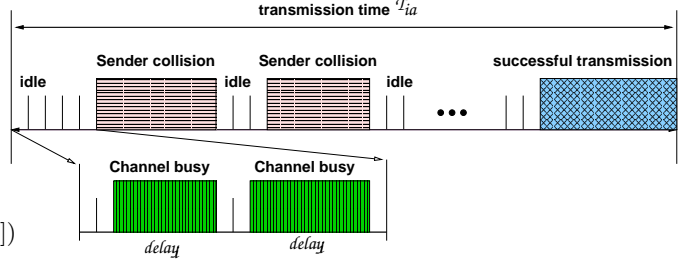


Fig. 11. Transmission time

In the rest of this subsection we determine all terms in above expression.

(i) Since the probabilities for collision and successful transmission are:

$$P_{coll} = \frac{1 - (1 - \tau)^M - M \cdot \tau(1 - \tau)^{M-1}}{1 - (1 - \tau)^M} \text{ and } (1)$$

$$P_{succ} = \frac{M \cdot \tau(1 - \tau)^{M-1}}{1 - (1 - \tau)^M}, (2)$$

$E[N_t]$ is

$$E[N_t] = \sum_{i=0}^{\infty} i \cdot P_{coll}^i \cdot P_{succ} = \frac{1 - (1 - \tau)^M}{M\tau(1 - \tau)^{M-1}} - 1 (3)$$

(ii) Since the probabilities for a slot to be idle or busy are:

$$P_{idle} = (1 - \tau)^M \text{ and } (4)$$

$$P_{busy} = 1 - (1 - \tau)^M, (5)$$

$E[idle]$ is

$$E[idle] = \sum_{i=0}^{\infty} i \cdot P_{idle}^i \cdot P_{busy} = \frac{(1 - \tau)^M}{1 - (1 - \tau)^M} (6)$$

(iii) Since the collision at the receiver side is caused by interfering data transmissions, $E[N_r]$ can be determined as $E[N_r] = \sum_{i=0}^{\infty} i \cdot P_r^i = \frac{1}{1 - P_r}$

$$(iv) \quad E[T_{coll}] = RTS + EIFS (7)$$

where $EIFS = SIFS + ACK + DIFS$.

$$(v) \quad E[T_{succ}] = RTS + CTS + \bar{m} + ACK + 3 * SIFS + DIFS (8)$$

(vi) Since we assume that I_r follows the same distribution as that of successful transmission time, T_{succ} :

$$E[I_r] = E[T_{succ}]. (9)$$

2) *SELECT-enhanced 802.11 throughput with interference at the receiver*: Figure 11 depicts the transmission time it takes to successfully transmit one data frame when we adopt SELECT (see Section IV-C) to address the interferences at the receiver. We add the following two notations for our analysis.

- μ_{ia} is the normalized throughput when the interference at the receiver is addressed with SELECT;
- T_{ia} is the time to successfully transmit one MAC frame with SELECT.

Similarly, the network throughput $\mu_{ia} = \frac{\bar{m}}{\bar{T}_{ia}}$, where \bar{m} is the average payload length and \bar{T}_{ia} is the average transmission time. Based on Figure 11, T_{ia} can be formulated as:

$$T_{ia} = \sum_{i=0}^{N_t} [delay_i + T_{coll_i}] + delay_{succ} + T_{succ}.$$

Since a SELECT sender recognizes the channel availability at the intended receiver, the idle (backoff) period at the sender is expanded to include the duration when the channel is unavailable at the receiver. Therefore $idle_i$ in the previous analysis is changed to $delay_i$:

$$delay_i = \sum_{j=0}^{idle_i} [(1 - P_r) * T_{slot} + P_r * (T_{slot} + I_r)],$$

$$delay_{succ} = \sum_{k=0}^{idle_{succ}} [(1 - P_r) * T_{slot} + P_r * (T_{slot} + I_r)].$$

Since

$$E[T_{ia}] = E \left[\sum_{i=0}^{N_t} [delay_i + T_{coll_i}] + delay_{succ} + T_{succ} \right],$$

and $E[delay] = E[idle] * (T_{slot} + P_r * E[I_r])$ and $E[delay_{succ}] = E[delay]$, $E[T_{ia}]$ can be determined as

$$E[T_{ia}] = (E[N_t] + 1) \cdot (E[idle] * (T_{slot} + P_r * E[I_r])) + E[N_t] \cdot E[T_{coll}] + E[T_{succ}].$$

We can determine the terms in above expression in a similar way to Section V-B.1. Specifically, (i) $E[N_t]$ is determined as in Eq. (3) with P_{idle} and P_{busy} in Eqs (1)-(2); (ii) $E[idle]$ is defined as in Eq. (6) with P_{idle} and P_{busy} in Eqs (4)-(5); (iii) $E[T_{coll}]$ is defined in Eq. (7), and $E[T_{succ}]$ is in Eq. (8); (iv) $E[I_r]$ is in Eq. (9);

3) *Derivation of interference probability, P_r* : The probability of the channel unavailability at receiver, P_r , depends on interfering transmissions. Note that for simplicity, we assume that only successful transmissions in the neighborhood of the receiver interfere with the transmission at the sender. We again add the following notations for our derivation.

- m_r is the payload size of the interfering transmissions at the receiver's neighborhood, assumed to be equal to m ;

- T_r is the transmission time taken to successfully transmit one frame at the receiver's neighborhood;
- Nt_r , $T_{coll,i,r}$, $T_{succ,r}$ are defined at the receiver, corresponding to Nt , $T_{coll,i}$, T_{succ} at the sender;
- N is the number of neighboring nodes at the receiver.

Let N be the number of neighboring nodes in the proximity of the receiver. The expected P_r can be $\overline{P}_r = \frac{\overline{m}_r}{\overline{T}_r}$ where \overline{m}_r is the average payload size and \overline{T}_r is the average transmission time at the receiver. Based on Figure 10 or 11 without receiver collisions, T_r can be formulated as follows:

$$T_r = \sum_{i=0}^{Nt_r} [idle_i * T_{slot} + T_{coll,i,r}] + idle_{succ} * T_{slot} + T_{succ,r}.$$

Similar to previous analysis, the expectation of T_r is:

$$E[T_r] = E \left[\sum_{i=0}^{Nt_r} [idle_i * T_{slot} + T_{coll,i,r}] + idle_{succ} * T_{slot} + T_{succ,r} \right]$$

Based on the independence assumption:

$$E[T_r] = E[Nt_r] \cdot E[idle] * T_{slot} + E[Nt_r] \cdot E[T_{coll,r}] + E[idle] * T_{slot} + E[T_{succ,r}]$$

The terms in the above expression are determined as follows; (i) Since collision and successful transmission probabilities can be decided the same way as in Eqs. (1)–(2) where τ_r (the transmission probability at the receiver) and N (the number of neighboring nodes at the receiver side) are used instead of τ and M , $E[Nt_r]$ is therefore:

$$E[Nt_r] = \sum_{i=0}^{\infty} i \cdot P_{coll,r}^i \cdot P_{succ,r} = \frac{1 - (1 - \tau_r)^N}{N\tau_r(1 - \tau_r)^{(N-1)}} - 1.$$

(ii) Given idle and busy probabilities for each slot in Eqs. (4) and (5):

$$E[idle] = \sum_{i=0}^{\infty} i \cdot P_{idle,r}^i \cdot P_{busy,r} = \frac{(1 - \tau_r)^N}{1 - (1 - \tau_r)^N}.$$

(iii) $E[T_{coll,r}]$ and $E[T_{succ,r}]$ are determined by Eq. (7).

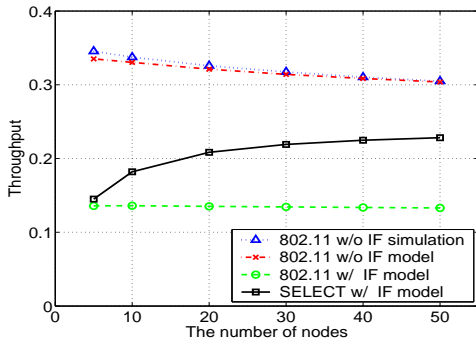


Fig. 12. Throughput v.s. the number of competing nodes

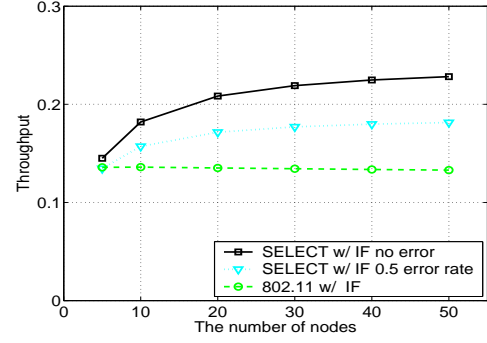


Fig. 13. Throughput comparison with 0.5 SELECT prediction errors

4) *Transmission probability, τ (τ_r):* The transmission probability, τ (or τ_r), is the probability with which each node attempts to transmit, and its computation basically follows [23]. τ is simply $1/(E[B] + 1)$, where $E[B]$ is the average backoff window size, and $E[B] = (E[CW] - 1)/2$, where $E[CW]$ is the average contention window size.

In order to determine $E[CW]$, we employ the iterative algorithm coarsely described in [23], which produces a sequence of $\{E[CW^{(n)}], n = 0, 1, 2, \dots\}$. The sequence is converged to $E[CW]$, such that $|E[CW^{(i)}] - E[CW^{(i-1)}]| < \epsilon$ where ϵ is given. At each iteration, we firstly determine the collision probability, $p_{coll}^{(i)}$ in what follows, since the contention window size is subject to collision probability:

$$p_{coll}^{(i)} = 1 - (1 - p_{coll}^{(i-1)})^{M-1}, \quad (10)$$

where M is the number of nodes. With the collision probability, we compute contention window size $E[CW^{(i)}]$ by using Lemma 4 in [23], and then we can compute $\tau^{(i)} = 2/(E[CW^{(i)}] + 1)$. Finally, we decide $\hat{\tau}$ when the iterative algorithm terminates, which is incorporated into the analytical models in Section V-B.

C. Analytical results

We present our analysis results in Figure 12 and the impact of SELECT prediction errors in Figure 13. Figure 12 depicts the throughput versus the total number of neighboring nodes, M (N), at the the sender (receiver) with the RTS/CTS enabled. The data packet size is set to 1000B. To simplify the presentation we compute the channel unavailability probability, P_r , under the assumption that the numbers of competing neighbors are the same at the sender and receiver: $M = N$ and $\overline{m} = \overline{m}_r$.

We first verify the accuracy of our modeling in scenarios where no interference at receiver is introduced. We compare the total throughput of our model with the output of *ns-2* simulations under different number of senders (i.e., different levels of contentions), marked as “802.11 w/o IF model” and “802.11 w/o IF simulation” in Figure 12 respectively. Note that the throughput of our model is within 2% of that in *ns-2* simulated throughput.

We then show the throughput generated by 802.11 DCF model and SELECT-enhanced 802.11 DCF model (as derived above). From Figure 12 we can clearly see that SELECT

consistently improve the throughput up to 72%, and the improvement increases as the number of interfering nodes increases. The reasons are as follows. In the presence of hidden/exposed receiver problem, 802.11 DCF encounters additional collisions at receiver and incurs more collisions before a data packet can be transmitted. In addition, as the number of collisions increases, a sender's contention window size stays large, which causes long idle backoff periods. SELECT-enhanced 802.11 DCF reduces the chances of collisions by taking into account potential interferences at receiver. It saves collisions and controls the contention window size at a lower level to keep the unproductive idle backoff period short.

We finally study the impact of SELECT prediction errors on the throughput in Figure 13. In the analysis we set a 50% error ratio for SELECT's estimate of the channel availability at receiver. In another word, half of the time the estimate for receiver's channel availability or unavailability is wrong. As shown in the figure the throughput gain decreases gracefully compared with what a perfect SELECT could achieve, but is still up to 37% higher than the original 802.11 DCF. This result suggests that under heavy interferences due to hidden/exposed receiver 802.11 DCF's clear channel assessment (CCA) performs even worse than a random guess.

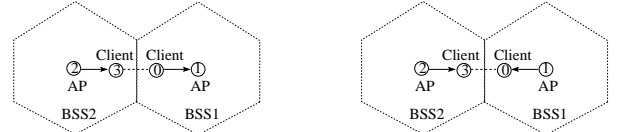
VI. PERFORMANCE EVALUATION

We implement SELECT in *ns-2* simulator version 2.28. For the current *ns-2* 802.11 implementation, nodes receive packets only when the RSS from the sender is greater than the receiving threshold, and the impact of any signal with RSS less than the carrier sense threshold is completely ignored - no matter how many those signals are. This is obviously an over-simplification of the reality. We replace this part of 802.11 functions with the ones developed in [24], so that all signals from the simulation are taken into account at receiver, and the combined signal to interference-noise (SINR) ratio is used to determine if an incoming signal can interfere or be received/captured. We use Two-Ray Ground radio propagation model, and the transmitting power is set so that the communication range is 115m, and the carrier sense threshold is set so that the default interference range is 200m. We use 2Mbps basic rate and 11Mbps data rate based on IEEE 802.11b. Each simulation runs for 45 seconds unless otherwise stated.

In all our SELECT simulations presented in this section we set $RSS^{min} = -100\text{dBm}$, consistent with our measurements shown in Figure 3 and 5, and CS^{thred} equal to the default carrier sense threshold. We require at least 10 samples in the current time window to make a success ratio prediction (the default is 100% for RSS's below the carrier sense threshold). We also add -100dBm additive white Gaussian noise to all RSS measurements, and set the sliding time window to 2 seconds. Manual static routing is used in the simulation scenario with CBR/UDP traffic.

We use three metrics to evaluate the performance. **Success ratio** is the total number of successful data messages received over the total number of RTS transmitted. **Data packet drops** denotes the total number of DATA packet drops at MAC layer

due to contention, normalized over the simulation duration. Both success ratio and number of packet drops per second can be viewed as metrics for the effectiveness of the collision avoidance. Finally **throughput** at transport layer serves as the metric for protocol efficiency and fairness in channel sharing.



(a) Exposed receiver problem (b) Hidden receiver problem

Fig. 14. Hidden/exposed receiver problem in 802.11 WLANs.

A. Hidden/exposed receiver problem

We first study the well-known exposed receiver problem as shown in Figure 14(a) where sender 0 and 2 are outside the interference range. Client 3 is an exposed receiver since it is placed in the communication/interference range of client 0, which is associated with another access point (node 1) in a neighboring BSS. Notice that in this configuration flow 0→1 will always succeed in the channel contention because its receiver (client 1) is not interfered by flow 2→3. We therefore vary the offered load (CBR/UDP rate) of flow 0→1, while keeping flow 2→3 always backlogged (with a 4Mbps CBR) to see how effectively SELECT can avoid collisions for flow 2→3 and keep the channel utilization high.

Figure 15-18 show the number of packet drops due to contention, the throughput gain for flow 2→3 compared with 802.11, channel access success ratio, and throughput respectively, when the exposed receiver (client 3) is out of the communication range of the interfering sender (client 0). Since sender 0 cannot receive receiver 3's CTS message, RTS/CTS handshake will not help flow 2→3 avoid collision but increase the overhead. We therefore use two-way handshake (i.e., without RTS/CTS) in 802.11 (as well as SELECT) for the best throughput. As we can see from these figures SELECT significantly reduces the number of packet drops due to contention by as much as 81.8% and increases the channel access success more than three-folds when flow 0→1 overloads the channel (with an offered load of 3.4Mbps). As a result, SELECT improves exposed receiver's throughput (flow 2→3) by as high as 140% (Figure 16), while increases the channel utilization from 69.8% to 86.1% (Figure 18).

We then move the exposed receiver 3 closer to sender 0 so that they are within communication range. That is, the dominating sender 0 will be able to yield to the exposed receiver (node 3) if its CTS is received. We therefore enable four-way handshake (i.e., with RTS/CTS) for 802.11 (and SELECT). The results are presented in Figure 19-22. Again SELECT consistently out-performs 802.11 in terms of reduced packet drop, increased channel access success ratio, and improved throughput. Moreover, with the help of RTS/CTS, it achieves almost optimal channel utilization as shown in Figure 22. Notice that as shown in Figure 20 and 22 SELECT achieves

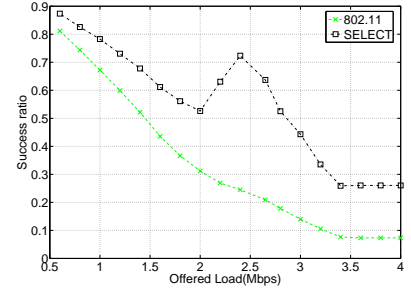
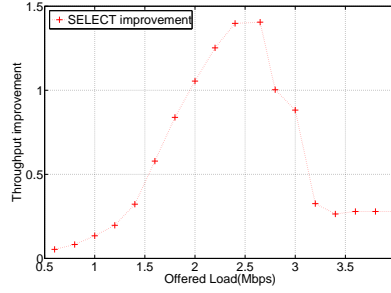
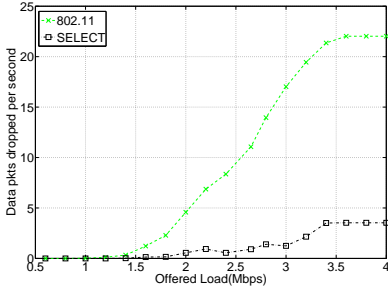


Fig. 15. Data packet drop at node 2 (w/o RTS/CTS) Fig. 16. Throughput gain at node 2 (w/o RTS/CTS) Fig. 17. Success ratio at node 2 (w/o RTS/CTS)

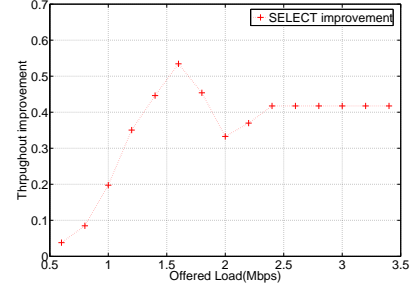
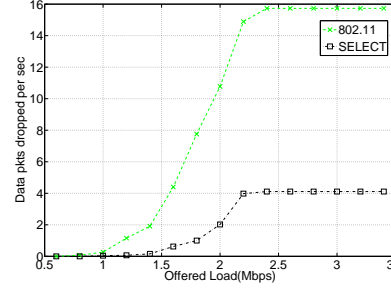
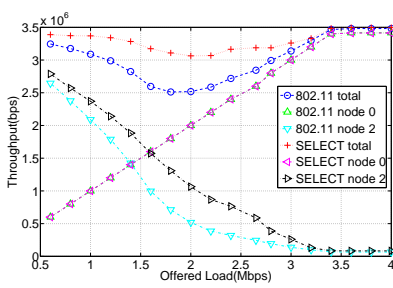


Fig. 18. Throughput profile (w/o RTS/CTS) Fig. 19. Data Packet drop at node 2 (w/ RTS/CTS) Fig. 20. Throughput gain at node 2 (w/ RTS/CTS)

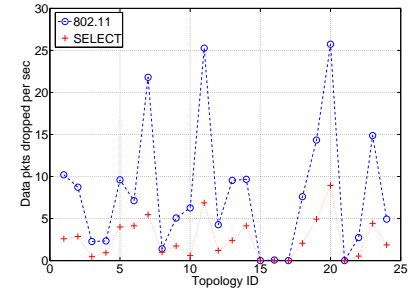
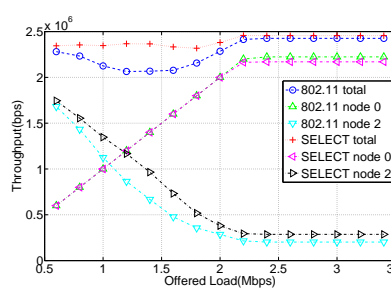
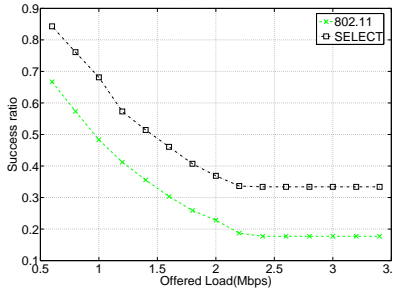


Fig. 21. Success ratio at node 2 (w/ RTS/CTS) Fig. 22. Throughput profile (w/ RTS/CTS) Fig. 23. Data packet drops: random topologies

the highest throughput gain for the exposed receiver when the throughputs of the two flows are close (i.e., when the offered load of flow $0 \rightarrow 1$ is around half the channel capacity) and the contention between these two flows is the maximum.

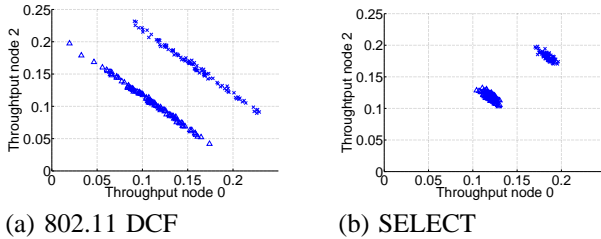


Fig. 24. Normalized throughputs (\times : w/ RTS/CTS \triangle : w/o RTS/CTS)

Finally we study the hidden receiver problem in Figure 14(b). Again sender 1 and 2 are outside the interference range from each other, and one of the receiver 0 and 3 will be

the hidden receiver when the other one is actively receiving. Note that in this scenario the topology is symmetric, and 802.11 DCF is able to achieve long-term fairness among those two competing flows. While SELECT continues to reduce the number of contention-induced packet drops by more than three folds (we omit the results due to lack of space), our simulations show that SELECT significantly improves the short-term fairness, as defined in [25]. In our simulations we keep both flows backlogged with 4Mbps CBR/UDP. The normalized throughputs of both flows in consecutive 0.4 second intervals for 802.11 and SELECT are shown in Figure 24. It is clear that SELECT helps solve the short-term unfairness problem because both flows intelligently predict when the channel is busy and behave socially to avoid unnecessary collisions.

B. Large random topologies

We finally study the performance of SELECT in large random 802.11b/g WLAN topologies. We study a 5x5 hexagon

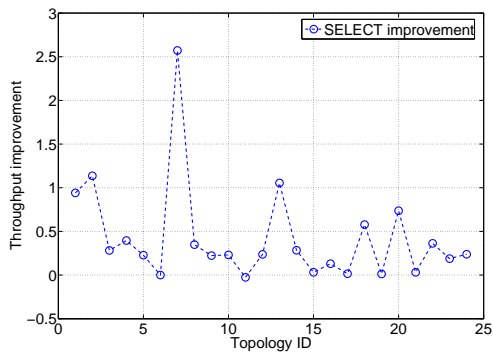


Fig. 25. Throughput gains in random topologies

BSS layout in a two-dimension space. The size of each hexagon BSS is set to the communication range. We emulate the automatic channel assignment strategy as implemented in most 802.11 access points. That is, we randomly go through all BSS's one by one, and assign one of the 3 non-overlapping channels that is least used in the BSS's six neighboring BSS's. We then randomly place a total number of 50 clients in the network resulting in an average of 2 clients per BSS. Each client associates with its nearest access points on the channel that is assigned to the BSS. The client also establish a CBR/UDP connection with its access point with the flow direction randomly determined.

We found out that automatic channel assignment always results in two or more neighboring BSS's on the same channel, consistent with the published measurement data [2], [1]. With an average of 2 clients per BSS, 60% of our simulated topologies suffer from hidden/exposed terminal problem. Figure 23 and 25 show the numbers of packet drops and throughputs for those hidden/exposed terminals in 24 random topologies. In summary SELECT reduces the number of packet drops by 59.8% (mean) with a standard deviation of 27% (Figure 23), and improves the throughputs of those hidden/exposed terminals by 42.6% (mean) with a standard deviation of 56.4%.

VII. CONCLUSION

Collision avoidance in wireless networks is complex and the correct perception of the channel availability is affected by a large number of factors. Many factors are dynamic in either temporal or spatial domains or both, and very difficult to model analytically or appraise at packet-level fine time granularity. Yet effective collision avoidance is a basic requirement for a wireless network. In this paper we propose SELECT, an effective and efficient self-learning collision avoidance to tackle the long-haunting hidden/exposed receiver problem. The SELECT design principle lies on the general observation that wireless system is complex and dynamic. White-box design methods based on analytical modeling or explicit, detailed reasoning are cumbersome and expensive. Instead, treating the entire system as a black box and directly reacting to the observable input-output could greatly reduce the complexity of the design and implementation, while achieving substantial

performance gain. It is our belief that this methodology could be applied to other open challenging problems in wireless networking.

REFERENCES

- [1] "WiFi Maps," <http://www.wifi-maps.com/>.
- [2] "Place Lab," <http://www.placelab.org/>.
- [3] A. Akella, G. Judd, P. Steenkiste, and S. Seshan, "Self management in chaotic wireless deployments," in *Proceedings of ACM MobiCom*, 2005.
- [4] S. Xu and T. Saadawi, "Does the IEEE 802.11 MAC protocol work well in multihop wireless ad hoc networks?" *IEEE Communication Magazine*, vol. 39, no. 6, pp. 130–137, June 2001.
- [5] Z. Fu, H. Luo, P. Zerfos, S. Lu, L. Zhang, and M. Gerla, "The impact of multihop wireless channel on TCP performance," *IEEE Transactions on Mobile Computing*, vol. 4, no. 2, pp. 209–221, March/April 2005.
- [6] D. Berger, Z. Ye, P. Sinha, S. Krishnamurthy, M. Faloutsos, and S. K. Tripathi, "TCP-friendly medium access control for ad-hoc wireless networks: Alleviating self-contention," in *Proceedings of IEEE MASS*, 2004.
- [7] C. L. Fullmer and J. Garcia-Luna-Aceves, "Solutions to hidden terminal problems in wireless networks," in *Proceedings of ACM SIGCOMM*, 1997.
- [8] V. Bharghavan, "Performance evaluation of algorithms for wireless medium access," in *Proceedings of IEEE Performance and Dependability Symposium*, 1998.
- [9] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic, "Impact of radio irregularity on wireless sensor networks," in *Proceedings of ACM MobiSys*, 2004.
- [10] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, "Link-level measurements from an 802.11b mesh network," in *Proceedings of ACM SIGCOMM*, 2004.
- [11] IEEE Computer Society, "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," IEEE standard 802.11, 1999.
- [12] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A medium access protocol for wireless LANs," in *Proceedings of ACM SIGCOMM*, 1994.
- [13] P. Bahl, R. Chandra, and J. Dunagan, "SSCH: Slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks," in *Proceedings of ACM MobiCom*, 2004.
- [14] J. So and N. H. Vaidya, "Multi-channel MAC for ad hoc networks: Handling multi-channel hidden terminals using a single transceiver," in *Proceedings of ACM MobiHoc*, 2004.
- [15] B. Raman and K. Chebrolu, "Design and evaluation of a new MAC protocol for long-distance 802.11 mesh networks," in *Proceedings of ACM MobiCom*, 2005.
- [16] F. Talucci, M. Gerla, and L. Fratta, "MACA-BI (MACA by invitation) a receiver oriented access protocol for wireless multihop networks," in *Proceedings of IEEE PIMRC*, 1997.
- [17] J. Garcia-Luna-Aceves and A. Tzamaloukas, "Receiver-initiated collision-avoidance in wireless networks," *ACM Wireless Networks, Special Issue on Selected Papers from Mobicom 99*, vol. 8, no. 2/3, pp. 249–263, 2002.
- [18] J. Zhu, X. Guo, L. L. Yang, and W. S. Conner, "Leveraging spatial reuse in 802.11 mesh networks with enhanced physical carrier sensing," in *Proceedings of IEEE ICC*, 2004.
- [19] X. Yang and N. H. Vaidya, "On the physical carrier sense in wireless ad-hoc networks," in *Proceedings of IEEE INFOCOM*, 2005.
- [20] "CC1000 single chip very low power RF transceiver," http://www.chipcon.com/files/CC1000_Data_Sheet_2_2.pdf.
- [21] "TinyOS," <http://www.tinyos.net/>.
- [22] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of IEEE INFOCOM*, 2002.
- [23] F. Cali, M. Conti, and E. Gregori, "Tuning of the IEEE 802.11 protocol to achieve a theoretical throughput limit," *IEEE/ACM Transactions on Networking*, vol. 8, no. 6, December 2000.
- [24] C. Hu and J. C. Hou, "A reactive channel model for expediting wireless network simulation," in *ACM SIGMETRICS Poster*, 2005.
- [25] M. Garetto, J. Shi, and E. Knightly, "Modeling media access in embedded two-flow topologies of multi-hop wireless networks," in *Proceedings of ACM MobiCom*, 2005.