Symbol-Level Detection: A New Approach to Silencing Hidden Terminals

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

***Abstract*—Hidden terminals are typical interference sources that can significantly reduce the throughput of a wireless network if it adopts the CSMA/CA MAC protocol. The RTS/CTS mechanism is a well-known solution to this hidden terminal problem. However, it only works well under the assumption that all hidden terminals can decode the CTS packets correctly. In the real world, the CTS packets might not be correctly received all the time due to either the CTS packets are unable to be decoded at remote hidden terminals or the CTS packets are collided with other packets at the hidden terminals. Both of these drawbacks can make the standard RTS/CTS mechanism fail to silence all hidden terminals, and deteriorate the throughput of the wireless network.**

In this paper, we present the RTS/S-CTS mechanism, a novel symbol-level detection mechanism that combats these two drawbacks. The RTS/S-CTS frames make slight changes to the standard RTS/CTS frames, and can be compatible with the standard 802.11 MAC layer. We design the symbol-level detection decoder (SLDD) and NAV decision algorithm that enable the S-CTS frame to be correctly detected from collisions and by remote hidden terminals. We build a testbed of RTS/S-CTS with GNURadio/USRP2 software radio to demonstrate its feasibility and run ns-2 simulations to evaluate its performance. The results show that the RTS/S-CTS can achieve up to 63% throughput improvement in the random topology network scenario compared with the standard RTS/CTS.

***Index Terms*—Cross-layer design, hidden terminal problem, RTS/CTS, signal correlation and detection, wireless networks**

I. Introduction

Hidden terminals are typically considered harmful in wireless networks since the interference from these hidden terminals can significantly reduce the throughput of wireless networks [1]–[6]. Current IEEE 802.11 MAC protocol mainly uses two mechanisms, carrier sense multiple access with collision avoidance (CSMA/CA) and RTS/CTS (virtual carrier sensing), to handle this hidden terminal problem [1], [4], [7]–[9]. In the standard RTS/CTS mechanism, the NAV time field of the RTS/CTS packets plays an important role that the terminals, which are not involved in the RTS/CTS handshake, can decode the NAV time and defer their transmissions for that time duration. Ideally, underlying the assumptions that (1) all hidden terminals are within the data transmission range of a receiver and (2) the CTS packet from the receiver suffers no collisions, the RTS/CTS mechanism is successful in contending for the wireless channel [10], [11]. However, in the real world, these two assumptions do not hold all the time. Consequently, it may cause two problems: (1) Remote hidden terminals that are out of the data transmission range of the receiver may not be able to decode the CTS packet correctly due to its low signal-to-noise-ratio (SNR). (2) The CTS packet may be collided with other concurrently transmitted packets so that hidden terminals cannot successfully decode the collided CTS packet due to its low signal-to-interference-plus-noise-ratio (SINR). Therefore, the hidden terminal problem cannot be fully solved by the standard RTS/CTS mechanism [3], [4], [8].

In this paper, we address the above two problems as the *remote hidden terminal problem* due to the low SNR of the received CTS packet, and the *CTS collision problem* due to the low SINR of the received CTS packet. Both problems make the CTS packet un-decodable at hidden terminals under low SNR/SINR environments. Thus, it is a challenge to convey the desired NAV time information to those hidden terminals. We propose a novel RTS/S-CTS mechanism that uses global-known symbol sequences to carry the NAV time information. We catalogue the NAV time durations and use different symbol sequences, which are called *S-NAV indicators* in this paper, to present different catalogued NAV time durations. These indicators can be detected and identified by a node under low SNR/SINR environments. The RTS/S-CTS frames require no change to the standard RTS/CTS packets at the MAC layer, but just append a new "S-NAV" field to the tail of the CTS frame at the PHY layer. The RTS/S-CTS handshake is same as the standard RTS/CTS handshake except that we devise a *symbol-level detection decoder* (SLDD) at the PHY layer to detect the S-NAV indicator, together with an *NAV decision algorithm* to pass the NAV time information up to the MAC layer. As the S-NAV does not have to be decoded into bits, the RTS/S-CTS can be compatible with current 802.11 MAC protocol.

Compared with the standard RTS/CTS mechanism, the RTS/S-CTS mechanism has following key features:

(1) The S-CTS works at the *symbol level*, i.e., the S-CTS frame's detectable range is enlarged from the data transmission range to the interference range, which is controlled by tuning the detectable threshold β_{S-NAV}. By contrast, the standard CTS works at the *bit level*, i.e., the CTS packet can only be correctly decoded within the data transmission range.

(2) The RTS/S-CTS mechanism can achieve good performance even under low SNR/SINR environments. It uses the

This work was supported in part by grants from Hong Kong RGC (PolyU 521312), Hong Kong PolyU (A-PL84, A-PJ16), and National Natural Science Foundation of China (No. 61272463).

symbol-level correlation method to detect the S-NAV indicator from the S-CTS frame. Moreover, by using a new identification algorithm, called *best candidate algorithm*, it can detect more S-NAV indicators under the same threshold. Therefore, the NAV time can be divided and presented more precisely compared with conventional correlation methods.

The main contributions of this paper are summarized as follows:

- We design, implement, evaluate and analyze the RTS/S-CTS on a 4-node testbed with GNURadio/USRP2 software radio. The results demonstrate the feasibility of the RTS/S-CTS to combat both remote hidden terminal and CTS collision problems.
- To reveal the performance improvement, we conduct simulations for various network topology scenarios in ns-2. The results show that our RTS/S-CTS can achieve more than 63% performance improvement compared with the standard RTS/CTS in the random network topology scenario.
- We propose a new mechanism of delivering the NAV time information using symbol-level information encoding and decoding. We show the feasibility of delivering such information at the symbol level. This also opens up the probability of conveying other control information from the bit level to the symbol level. For example, a similar mechanism is applied in [12] to eliminating the exposed terminal problem as well.

The rest of the paper is organized as follows: The Section II addresses two problems suffered by the standard RTS/CTS. We describe the RTS/S-CTS mechanism in Section III. In Section IV we discuss the hardware implementation and in Section V we conduct the performance evaluation through simulations. Section VI further discusses several issues about the mechanism. We review the related work in Section VII and conclude the paper in Section VIII.

II. Drawbacks of Standard RTS/CTS Mechanism

A. Remote Hidden Terminal Problem

The standard RTS/CTS works when a node's data transmission range is equal to its interference range. However, recent studies [1], [4], [8] have revealed that packets' transmission power can raise the environmental noise of the wireless channel, which causes the node's interference range to be much larger than its data transmission range. To avoid interference, the distance between transmitter-receiver d and the distance between receiver-interferer d_i should satisfy $d_i \geq \sqrt[k]{\beta_{SNR}} \cdot d$, where β_{SNR} is the SNR threshold that the received packet can be correctly decoded into bits and k is the signal decay factor [3]. That means, when $d_i < \sqrt[k]{\beta_{SNR}} \cdot d$, the receiver cannot correctly decode the packet. Note that the impact of the interferer on the receiver's reception of the packet is related to the distance between transmitter and receiver.

Assume all nodes have the same data transmission range T_r. When $d \leq \frac{T_r}{\sqrt[k]{\beta_{SNR}}}$, the interference range I_r satisfies $d_i \leq I_r \leq T_r$, i.e., the interferer is within the data transmission range of the receiver. Since the CTS packet from the receiver can be correctly decoded by the interferer, the RTS/CTS handshake can successfully silence the hidden interferer. In that case, there is no remote hidden terminal problem (i.e., *Area 2* in Fig. 1). However, when $\frac{T_r}{\sqrt[k]{\beta_{SNR}}} < d \leq T_r$, we get $T_r < d_i \leq I_r \leq \sqrt[k]{\beta_{SNR}} \cdot T_r$. As the interferer is a remote terminal that locates outside the data transmission range of the receiver, the SNR of the CTS packet at the receiver is below β_{SNR}. The RTS/CTS handshake fails to silence this remote hidden terminal. Thus, any transmission from this remote hidden node will cause the receiver's packet corruption (i.e., *Area 1* in Fig. 1). In this case, the standard RTS/CTS fails to silence node C's transmission to node D, which in turn causes node A's transmission to be collided at node B. We also call this remote hidden terminal drawback as *Low-SNR-CTS problem*.

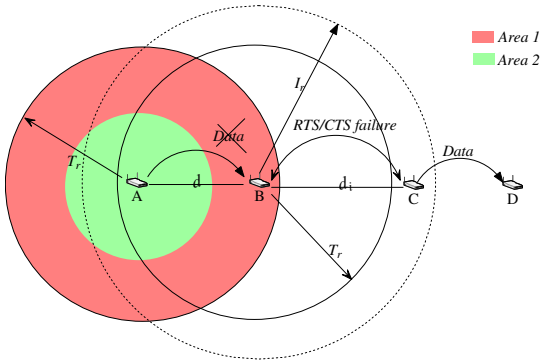


Fig. 1. A remote hidden terminal scenario. Since the CTS packet from node B cannot be decoded by the remote hidden node C, the standard RTS/CTS handshake fails to silence node C. Node C's data transmission causes node A's data transmission corrupted at node B. Here, d and d_i satisfy the relationship $T_r < d_i < \sqrt[k]{\beta_{SNR}} \cdot d$.

B. CTS Collision Problem

The standard RTS/CTS mechanism makes the assumption that RTS/CTS packets would not collide with other packets. This assumption cannot hold when the network's workload becomes high. When multiple nodes concurrently transmit packets, the RTS/CTS packets may be collided with other packets. The corruption of RTS packet does not trouble the system much because, if the RTS is missed, the RTS/CTS handshake will re-initiate after a waiting time. However, the corruption of CTS packet at hidden node(s) can deteriorate the throughput of the system. If the CTS sent by the receiver fails to silence its neighbor(s) for enough NAV time, its data reception may be corrupted by the hidden node(s)' transmissions, which will force the transmitter to retransmit the entire data packet. To emphasize this problem, we also call this CTS collision drawback as *Low-SINR-CTS problem*.

A simple CTS collision scenario is illustrated in Fig. 2: Nodes A~G are wireless stations that form a straight line topology. Fig. 2(a) shows that node B's CTS packet is collided with node D's data packet at node C. Due to the low SINR, the CTS packet cannot be correctly decoded by node C, and node C do not renew its NAV waiting timer. Hence, when node A transmits the data packet to node B, node C may initialize a

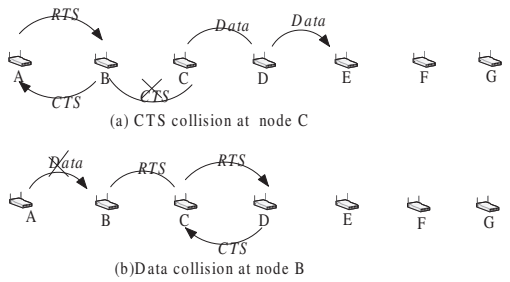


Fig. 2. A CTS collision scenario: (a) Node B's CTS packet is collided with node D's data packet at node C. (b) Because the CTS packet is collided at node C in (a), node C cannot defer its transmission for the NAV time and node C's RTS packet will cause node A's data packet to be corrupted at node B.

new RTS/CTS handshake with node D, as shown in Fig. 2(b). Consequently, node A's data packet is collided at node B. Note that though the RTS/CTS packets may be hardly collided with each other due to their small packet sizes, the probability that the CTS packets are collided with data packets will be high when the network's workload becomes high.

The Low-SNR-CTS and Low-SINR-CTS problems are two main drawbacks that significantly deteriorate the performance of the standard RTS/CTS mechanism. To combat these two drawbacks, we propose a novel RTS/S-CTS mechanism to make the S-CTS frame detectable at the symbol level even under low SNR/SINR environments.

III. RTS/S-CTS Mechanism

Similar to the standard RTS/CTS, the RTS/S-CTS silences the neighboring nodes through the exchange of RTS/S-CTS frames: If a node that is not involved in the transmission can successfully decode or detect the RTS/S-CTS frames, it defers its transmission for the NAV time.

As we have addressed in previous section, because of the Low-SNR/SINR-CTS problems, the RTS/CTS cannot silence all interferers. On the contrary, the RTS/S-CTS can silence these interferers through the symbol-level correlation method. As the low-SNR/SINR-CTS problems only relate to the CTS frame, there is no change to the RTS frame format. To make the S-CTS detectable under low SNR/SINR scenarios, a new "S-NAV" field is appended to the CTS frame at the PHY layer, which makes the S-CTS packet same as the standard CTS packet at the MAC layer (Fig. 3). To specify the format difference of a packet between the MAC and PHY layers, we call it as "packet" at the MAC layer and as "frame" at the PHY layer in this paper.

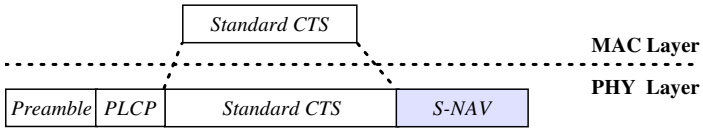
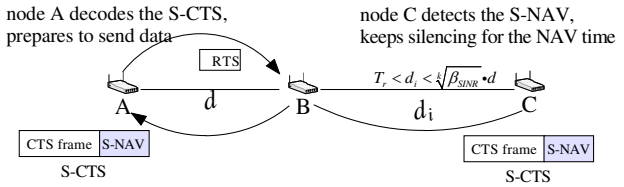
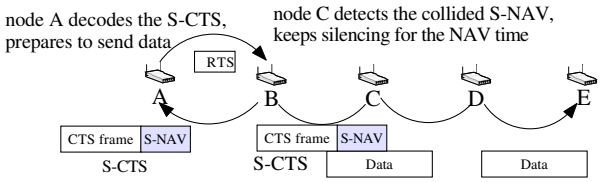


Fig. 3. New S-CTS frame format at the physical layer. We do not change the standard CTS packet at the MAC layer, but only append a S-NAV filed at the PHY layer.



(a) Low-SNR-CTS scenario



(b) Low-SINR-CTS scenario

Fig. 4. The RTS/S-CTS can combat both Low-SNR/SINR-CTS problems.

The new S-CTS frame can combat both Low-SNR/SINR-CTS problems (Fig. 4):

(a) For the Low-SNR-CTS problem case (Fig. 4(a)), because the distance between node B and node C, d_i, satisfies $T_r < d_i < \sqrt[k]{\beta_{SNR}} \cdot T_r$, node C cannot decode node B's S-CTS frame into bits correctly. However, node C can detect the S-NAV field at the symbol level and obtain the NAV time information. Thus, node C can keep silencing for the NAV time.

(b) For the Low-SINR-CTS problem case (Fig. 4(b)), although node B's S-CTS is collided by node D's data transmission at node C, node C can still detect the S-NAV field at the symbol level, obtain the NAV time information, and keep silencing for the NAV time.

A. S-CTS Frame Generation

When a station receives a RTS packet, it checks if it is the designated receiver. If yes, it achieves the NAV time from the RTS's duration field, minus the time that is required to transmit the CTS frame and ACK frame plus three SIFS intervals. That means, only the time for transmitting the pending data is remained.

After the calculation, the receiver encodes the resultant NAV time, called "S-NAV time", into the S-NAV field (see details in Section III-C2), and appends this S-NAV field to a standard CTS frame at the physical layer to build a S-CTS frame. The receiver responses this S-CTS frame to the transmitter.

B. S-CTS Frame Reception

A station normally uses two mechanisms, hard decision decoder (HDD) and soft decision decoder (SDD), to decode the incoming signals at the PHY layer [8], [13]. Either one can deliver the bits up to the MAC layer correctly if received signals meet the SNR/SINR threshold requirement. However, as the S-CTS frame needs to be detected under low SNR/SINR environments, we design a new decoder, called *symbol-level detection decoder* (SLDD), to detect the S-CTS frame at the PHY layer, as shown in Fig. 5:

1) Under High SNR/SINR Environments: When the incoming signal meets the threshold requirement, the correct bits are

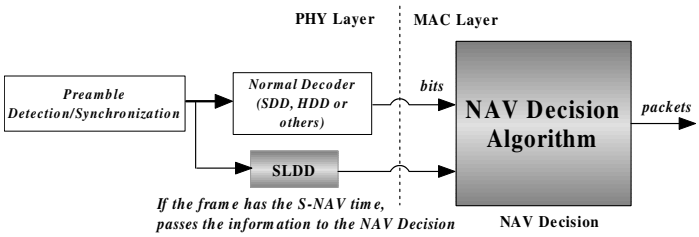


Fig. 5. The receiving process of S-CTS frame. Gray blocks are the new components. The SLDD detects the S-CTS frame's S-NAV field and passes this S-NAV time information to the NAV decision block at the MAC layer.

delivered to the MAC layer. For a received S-CTS frame, as the MAC layer can correctly obtain the standard CTS packet from the PHY layer, both the receiver address *RA* and the NAV time T_{NAV} are derived. In this scenario, the S-NAV time information in the S-NAV field becomes useless.

2) Under Low SNR/SINR Environments: Due to the low SNR/SINR, the hidden terminals cannot correctly decode the CTS packet at the MAC layer. However, for any node not involved in the data transmission, the only useful information carried by the CTS packet is the NAV time. This information is also presented as the S-NAV time information, which is stored in the S-NAV field and can be detected by the SLDD at the PHY layer (see details in Section III-C).

When the NAV decision block only receives the S-NAV time information and no *RA* is obtained from the S-CTS frame, the station knows that it does not involve in the data transmission. Then the NAV decision adds the S-NAV time T_{S-NAV} achieved from the SLDD, together with the time required to transmit one ACK frame plus two SIFS intervals to calculate a new NAV time $T_{NAV-TIME}$. If the new NAV time is larger than the time kept in the NAV-timer $T_{NAV-timer}$, the NAV decision renews the NAV-timer and ceases for the duration. The *NAV decision algorithm* is listed in Algorithm 1:

Algorithm 1 NAV Decision Algorithm

Input: The digital bits from the normal HDD/SDD decoders; the S-NAV time from the SLDD decoder.

1: **if** the packet is correctly decoded **then**
2: Extract *RA* and T_{NAV} from the digital bits; set $T_{NAV-TIME}$ to T_{NAV}.
3: **else**
4: Set *RA* to *NULL*; obtain T_{S-NAV} from the S-NAV field; set $T_{NAV-TIME}$ to $T_{S-NAV} + T_{ACK} + 2 \cdot T_{SIFS}$.
5: **if** *RA* is the station's address **then**
6: Prepare for sending the data packet.
7: **else**
8: Renew the NAV-timer if $T_{NAV-TIME} > T_{NAV-timer}$; cease for $T_{NAV-timer}$.

C. S-NAV Detection and Identification

S-CTS's two main challenges pertain to detecting the S-NAV field and discerning the different NAV time information, both of which depend on how a known-sequence signal is detected. Similar to that in [14]–[16], we accomplish this by using cross-correlation between the incoming signal and the known-sequence signal. Say that if the known-sequence signal has L samples, the receiver aligns the first L incoming signal samples with the known L samples, computes the correlation value, then, shifts the incoming samples by one, and re-computes the correlation value. The correlation value reaches the peak when the incoming samples are perfectly matched to the known samples. Thus, a station can verify the presence of a known-sequence signal even if it is submerged in a high noise environment. We call this "signal correlation and detection".

1) Signal Correlation and Detection: When a packet transmits over the wireless channel, the transmitter needs to map the digital bits over an analog passband symbols in the digital modulation process. Mathematically, analog passband symbols are represented as a stream of discrete complex samples. At the receiver side, the RF down-converter samples the incoming signal, and derives a stream of discrete complex samples. However, those received samples differ from transmitted samples. Assume an original transmitted sample is $A[n]e^{j\phi[n]}$, where $A[n]$ refers to the amplitude of the n^{th} sample and $\phi[n]$ refers to its phase, the received sample $y[n]$ can be represented as:

$$y[n] = HA[n]e^{j\phi[n]} + w[n] \tag{1}$$

Here, H is also a complex number representing the channel coefficient between transmitter and receiver, $w[n]$ is the Gaussian noise at n^{th} sample.

In practice, samples are actually distorted due to hardware constraint and wireless channel effect: frequency offset, sampling offset, and inter-symbol interference [2], [15]. The hardware implementation has to estimate and compensate the three offsets so as to decode the incoming signal samples. For example, considering the frequency offset between transmitter and receiver Δf over time Δt, Eq. (1) can be represented as:

$$y[n] = HA[n]e^{j\phi[n]} \cdot e^{j2\pi\Delta f\Delta t} + w[n] \tag{2}$$

Assume $x_{kn}[i]$ is the i^{th} complex sample of the known sequence, i.e., $x_{kn}[i] = A_{kn}[i]e^{j\phi_{kn}[i]}$. $y_{tr}[i]$ is the incoming signal sample from the transmitter, $y_{int}[i]$ is the incoming signal sample from the interferer, L is the length of the known sequence and $0 \le i \le L-1$. The signal correlation value, E_{kn}, can be calculated as:

$$E_{kn} = |\sum_{i=0}^{L-1} \bar{x}_{kn}[i](y_{tr}[i] + y_{int}[i])| \tag{3}$$

Here, $\bar{x}_{kn}[i]$ represents the complex conjugate of $x_{kn}[i]$.

However, Eq. (3) cannot compute the correlation spike because the frequency offset can destroy the correlation. The receiver needs to compensate the frequency difference Δf and shift each sample by $-2\pi\Delta f\Delta t$. Since the known sequence is independent of the signal from y_{int} and the noise, as long as L is large enough, $\sum_{i=0}^{L-1} \bar{x}_{kn}[i] \cdot y_{int}[i] \cdot e^{-j2\pi\Delta f\Delta t}$ and $\sum_{i=0}^{L-1} \bar{x}_{kn}[i] \cdot w[i] \cdot e^{-j2\pi\Delta f\Delta t}$ would be close to 0. On contrast, when the known sequence is presented in the incoming signal samples, the correlation value E_{kn} will reach a spike:

$$
\begin{aligned}
E_{kn} &= |\sum_{i=0}^{L-1} \bar{x}_{kn}[i](y_{tr}[i] + y_{int}[i]) \cdot e^{-j2\pi\Delta f\Delta t}| \\
&= |\sum_{i=0}^{L-1} \bar{x}_{kn}[i] \cdot H_{tr} x_{kn}[i] e^{j2\pi\Delta f\Delta t} \cdot e^{-j2\pi\Delta f\Delta t} \\
&\quad + \sum_{i=0}^{L-1} \bar{x}_{kn}[i] \cdot y_{int}[i] \cdot e^{-j2\pi\Delta f\Delta t} \\
&\quad + \sum_{i=0}^{L-1} \bar{x}_{kn}[i] \cdot w[i] \cdot e^{-j2\pi\Delta f\Delta t}| \\
&\approx |H|| \sum_{i=0}^{L-1} A_{kn}[i] e^{j\phi_{kn}[i]}|^2 \qquad (4)
\end{aligned}
$$

To detect the known sequence, a threshold β_{kn} is introduced to compare with E_{kn}: If $E_{kn} \geq \beta_{kn}$, the receiver detects the presentence of the sequence; otherwise, the sequence is considered absent.

Normally, threshold β_{kn} can be defined as $\beta_{kn} = \psi \cdot L \cdot RSSI_{kn}$ [15], [16], where ψ is a constant and $RSSI_{kn}$ is the received signal strength indicator of the known-sequence signal. Thus, the comparison inequality can be changed to:

$$
\frac{E_{kn}}{L \cdot RSSI_{kn}} \geq \psi. \qquad (5)
$$

We call $\frac{E_{kn}}{L \cdot RSSI_{kn}}$ as "normalized correlation value".

2) S-NAV Detection and Identification: As we know, the S-NAV field plays a critical role in defeating the Low-SNR/SINR-CTS drawbacks. When receiving the S-CTS frame, a hidden terminal can achieve the S-NAV time information from the S-NAV field and keep silencing for the requested time. There are two issues for the S-NAV detection and identification process: (1) How to provide enough global-known sequences, which are called *S-NAV indicators*, to carry different S-NAV time information? (2) How to identify those S-NAV indicators under low SNR/SINR environments? To solve these problems, we propose both *indicators mapping function* and *best candidate algorithm* to present and identify different NAV time information.

In the IEEE 802.11 standard, the size of MAC service data unit (MSDU) is limited to 2272 bytes. However, in the IPv4 and Ethernet standard (Version 2), the maximum transmission unit (MTU) cannot exceed 1500 bytes. Since the IEEE 802.11 MAC still uses IPv4 as its upper layer, the MTU is set as 1500 bytes. Consequently, the data transmission time cannot exceed a maximum transmission time T_{max}. The indicator mapping function divides all possible data transmission time into N catalogues and maps each catalogued time T^i_{cata} to a global-unique S-NAV indicators. T^i_{cata} is calculated as follows:

$$
T^i_{cata} = \frac{T_{max}}{N} \cdot i, \qquad (6)
$$

where i is the catalogued index of the S-NAV indicator and $1 \leq i \leq N$. Note that the indicators mapping function could keep the station(s) waiting longer than actual data transmission time, which is called "catalogue overhead" in this paper.

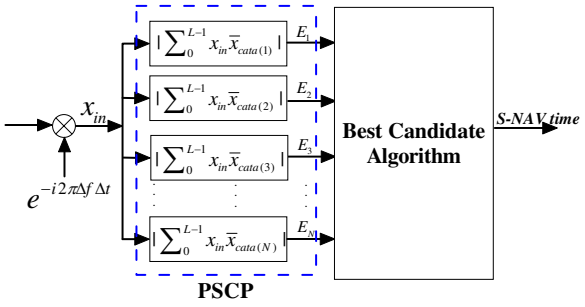


Fig. 6. The structure of SLDD.

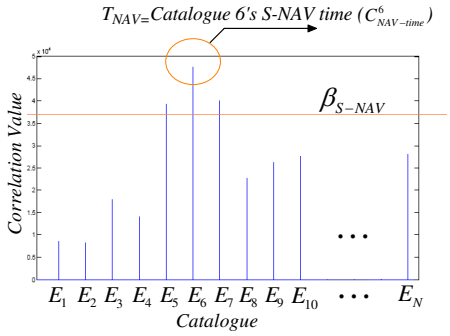


Fig. 7. N-catalogued S-NAV indicators scheme. The example shows that catalogue 6 is the best candidate to be selected.

When the station prepares for the S-CTS frame, it obtains the NAV time from the RTS frame, calculates the data transmission time, finds the catalogued time just longer than the data transmission time, maps the catalogued time to the corresponding S-NAV indicator, and stores that S-NAV indicator in the S-NAV field of the S-CTS frame at the PHY layer.

To decode the S-NAV time from the S-CTS frame, the SLDD resorts to a *parallel signal correlation process* (PSCP) (Fig. 6): The SLDD correlates the incoming samples with N different S-NAV indicators and picks up the results that exceed β_{S-NAV}, which are called the *candidates*. The SLDD compares those candidates to find the one with maximum value E_i, which is the *best candidate*. The SLDD gets the index i and sets the S-NAV time T_{S-NAV} as the catalogued time T^i_{cata}. The best candidate algorithm is listed as Algorithm 2:

Algorithm 2 Best Candidate Algorithm

Input: Incoming symbol samples from the RF down-converter.
Output: The S-NAV time T_{S-NAV}.

1: Correlate the incoming samples with N different S-NAV indicators; pick up the correlation values that exceed β_{S-NAV} as the candidates.
2: **if** the number of candidates > 0 **then**
3: Compare those candidates and find out the one with maximum value E_i as the best candidate.
4: Get the catalogued index i of the best candidate; set T_{S-NAV} to the catalogued time T^i_{cata}; .
5: **else**
6: Do nothing.

Fig. 7 illustrates the decoding procedure of SLDD with N

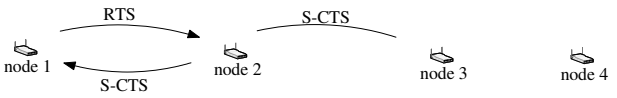


Fig. 8. The network topology of the 4-node testbed.

S-NAV indicators. Different from the conventional correlation detections which only allow one correlation value to exceed the threshold, the SLDD allows several candidates to exceed the threshold β_{S-NAV}, and the best candidate algorithm can pick up the best candidate to output the S-NAV time.

IV. Hardware Experiments

In this section, we reveal the hardware implementation and experimental methodology. Also, we discuss some practical issues about S-NAV detection and identification.

A. Hardware Implementation and Experimental Methodology

1) Hardware Implementation: We implemented the RTS/S-CTS mechanism on a 4-node GNURadio/USRP2 testbed. Each node is a commodity PC connected to a Universal Software Radio Peripheral 2 (USRP2) [17] with RFX2400 daughterboard. The RFX2400 operates at the 2.4GHz frequency range. All PCs are installed Ubuntu 10.04 and GNURadio [18].

The RTS/S-CTS uses the BPSK modulation/demodulation module, which is commonly used in the 802.11 standard. We used the default GNURadio configuration for the communications, i.e., on the transmitter side, the DAC rate is 400e6 samples/s, the interpolation rate is 200 (4 interpolation rate in the DAC chip itself and 50 interpolation rate controlled by GNURadio), and the number of samples per symbol is 2; on the receiver side, the ADC rate is 100e6 samples/s and the decimation rate is 50. Given the above parameters and a BPSK modulation, the resulting bit rate is 1Mbps.

2) Experimental Methodology: USRP2 has hardware delays in transmitting samples from the RF front-end to its connected commodity PC, also GNURadio incurs artificial software delay to process these samples. Thus, it is difficult to conduct a real time evaluation of the RTS/S-CTS in high bit rates. Hence, we resorted to the trace-based evaluation that is also used in [15], [16]. Each node saves all the outgoing and incoming samples for off-line processing.

We set up the 4-node GNURadio/USRP2 testbed shown as Fig. 8: (1) To evaluate the remote hidden terminal problem, we increased the distance between nodes 2 and 3 to make them not communicate with each other. We made nodes 1 and 2 exchange RTS/S-CTS frames to set up link 1. Similarly, nodes 3 and 4 set up link 2. Here, node 3 is a remote hidden node of node 2. (2) To evaluate the CTS collision, we set the distance between nodes 2 and 3 less than the transmission range. Nodes 1 and 2 exchange RTS/S-CTS frames to set up a link, node 3 is a hidden node of node 1, and node 4 broadcasts some random data as an interferer, causing the CTS collision at node 3.

The design of S-NAV plays a crucial role in the proposed RTS/S-CTS mechanism. There are three factors that affect the design of S-NAV indicators: the length of S-NAV indicators, the detectable threshold β_{S-NAV} of S-NAV indicators, and the minimum Hamming distance among S-NAV indicators. We conducted hardware experiments to study how these factors affect the S-NAV detection and identification under low SNR/SINR environments.

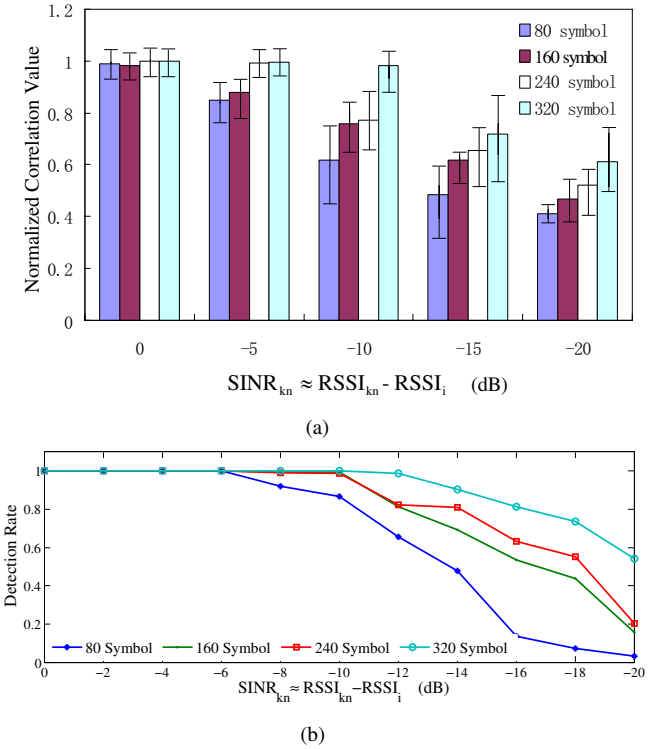


Fig. 9. (a) Normalized correlation value with various indicator lengths under different SINRs. (b) Detection rate with various indicator lengths under different SINRs.

B. Length of S-NAV Indicators

A pertinent concern is how long the S-NAV indicator should be. Evident from Fig. 9(a), we can see that longer indicators can make higher normalized correlation values, which also make the indicators easily detected under lower SINR environments (Fig. 9(b)).

However, a longer indicator also means more channel occupation time. Table I gives the channel occupation time overhead with various indicator lengths.

Standards \ Indicator Length	80	160	240	320
802.11a/g	6.67	13.3	20	26.6
802.11b	7.27	14.55	21.82	29.09
	Time (Microsecond)			

TABLE I
Channel Occupation Time Overhead.

To make a trade-off between SINR and channel occupation overhead, a metric *indicator length utilization* U_L is defined as:

$$U_L = |\frac{SINR_{min}}{L}|, \quad (7)$$

where $SINR_{min}$ is the lowest SINR that, given length L, the indicator can be detected (with detection rate $\geq 95\%$, Fig. 9(b)).

Indicator Length	80	160	240	320
Utilization	0.075	0.069	0.046	0.041

TABLE II
INDICATOR LENGTH UTILIZATION.

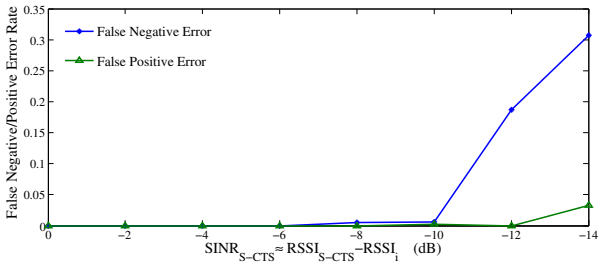


Fig. 10. False negative/positive error rate with 160 symbols. ($\psi_{S-NAV} = 0.55$)

Table II gives the various indicator length utilization. By further considering the available number of S-NAV indicators, in our testbed, we set the length of S-NAV indicator as 160 symbols.

C. Threshold β_{S-NAV}

Another key factor that affects the performance of S-NAV detection (false negative/positive error) is the threshold β_{S-NAV}. According to Eq. (5), we can get

$$\frac{E_{S-NAV}}{L \cdot RSSI_{S-NAV}} \geq \psi_{S-NAV}. \quad (8)$$

ψ_{S-NAV} is closely related to both false negative error and false positive error. Fig. 10 shows the two error rates with 160 symbols. Obviously, the false negative error dominates the detection's performance within $-14dB$. From Fig. 9(b), we can see that the false negative error is more related to the SINR and indicator length. The false positive error rate is mainly due to that the correlation value of S-NAV indicator and data is larger than β_{S-NAV}, which can be defeated by a large Hamming distance between the indicator and data.

Table III shows the decreasing trend as the Hamming distance increases. When the Hamming distance becomes 52, the false positive error rate is 0.2%.

Hamming Distance	34	40	46	52
FPE Rate	0.170	0.047	0.008	0.002

TABLE III
THE RELATIONSHIP BETWEEN HAMMING DISTANCE AND FALSE POSITIVE ERROR (FPE) RATE. (SINR = $-10dB$, $\psi_{S-NAV} = 0.55$)

Clearly, Eq. (8) mainly deals with the Low-SINR-CTS problem. When a station is out of the data transmission range, $RSSI_{S-NAV} \leq \beta_{SNR} + RSSI_{no}$ where β_{SNR} is the SNR threshold for correctly decoding packets and $RSSI_{no}$ is the environment noise (typically, -98∼-95dBm). To balance the Low-SNR-CTS problem and exposed terminal problem [6], in this case, $\beta_{SNR} + RSSI_{no}$ is used instead of $RSSI_{S-NAV}$. Thus, Eq. (8) changes to

$$\frac{E_{S-NAV}}{L \cdot (\beta_{SNR} + RSSI_{no})} \geq \psi_{S-NAV}. \quad (9)$$

Fig. 11 shows the hardware result of the detection rate under different SNRs.

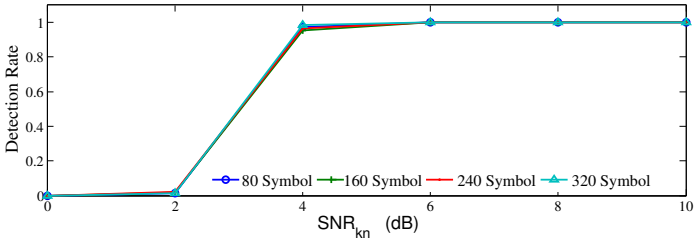


Fig. 11. Detection rate with various indicator lengths under different SNRs.

D. Minimum Hamming Distance among S-NAV Indicators

The minimum Hamming distance between any pair of S-NAV indicators will affect the performance of S-NAV identification. Similar to the false positive error rate, the best candidate algorithm may also choose a wrong candidate as the S-NAV indicator, causing the S-NAV time information to be decoded incorrectly. We call this error as "indicator decoding error". The indicator decoding error rate has close relationship with the minimum Hamming distance between any pair of S-NAV indicators. From Table IV, we can see that, by increasing the Humming distance between two S-NAV indicators, the indicator decoding error rate can be less than 0.1% when the Humming distance is 22. Therefore, the indicator decoding error rate can be minimized by enlarging the Humming distance between any pair of S-NAV indicators. However, enlarging the Hamming distance would reduce the available number of S-NAV indicators that the system can use.

Hamming Distance	6	10	14	18	22
PDE Rate	0.0570	0.0200	0.0079	0.0038	0.0010

TABLE IV
THE RELATIONSHIP BETWEEN HAMMING DISTANCE AND INDICATOR DECODING ERROR (PDE) RATE. (SINR = -10dB)

Compared with the conventional correlation detection [13], [15], [16] that only allows one candidate, which requires the minimum Humming distance to be 52, the best candidate algorithm reduces the required minimum Humming distance to be 22, which means that we can design more S-NAV indicators to alleviate the catalogue overhead. In our USRP2 experiment, the Humming distance of the 160 symbol-length indicators is set as 22. We can design more than 150 different S-NAV indicators, consequently, the catalogue overhead is below 13.3μs in 802.11a.

V. PERFORMANCE EVALUATION

In this section, we give ns-2 simulation results that show the effectiveness of our RTS/S-CTS mechanism to solve the Low-SNR/SINR-CTS problems. We simulated the RTS/S-CTS under different network scenarios: a 4-node line topology for the Low-SNR-CTS problem scenario (Fig. 1), a 7-node line topology for the Low-SINR-CTS problem scenario (Fig. 2), and a 16-node random network topology for general scenario (Fig. 14). We compared the performance of the RTS/S-CTS with standard CSMA/CA and RTS/CTS protocols. We modified ns-2's source code at the physical layer to support the S-CTS's symbol-level detection by using the hardware experiment results. We also considered the catalogue overhead of the RTS/S-CTS. To

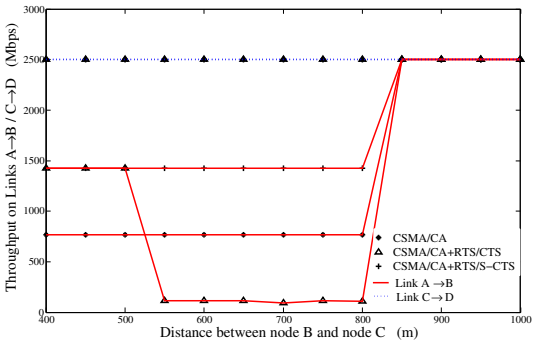


Fig. 12. The throughput in the Low-SNR-CTS scenario.

evaluate the performance, we employed two common metrics, throughput and packet delivery rate. Table V lists the parameter configurations used in our simulation.

Parameter	Value	Parameter	Value
Transmission range	500m	Preamble	16μs
Carrier sensing range	870m	SIFS	16μs
S-NAV	13.3μs	DIFS	34μs
Catalogue overhead	13.3μs	CWmax	1023μs
Link capacity	6Mbps	CWmin	15μs
Packet size	700~1500 bytes	Time slot	9μs

TABLE V
PARAMETER CONFIGURATIONS FOR NS-2 SIMULATION.

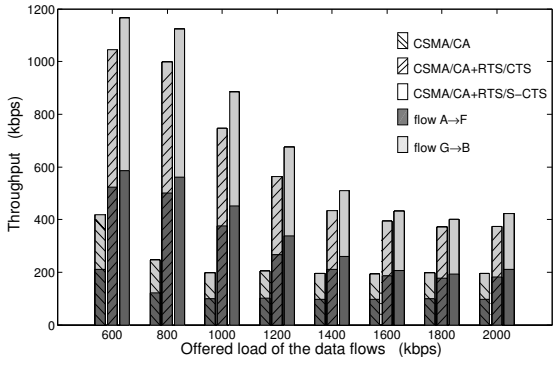
A. Low-SNR-CTS Problem Scenario

To evaluate the RTS/S-CTS's performance in the Low-SNR-CTS environment, we built up two directional links (A→B and C→D) as Fig. 1. The distance between A→B (and C→D) is 480m. We set data flow on each link as 2.5Mbps. We varied the distance between B and C (denoted as d_i) from 400m to 1000m to study the changes of throughput on both links. The simulation result (Fig. 12) shows that RTS/S-CTS can completely solve the remote hidden terminal problem (when d_i ranges from 500m to 800m) while the other two schemes cannot compete under this circumstance. More interestingly, we observed that the RTS/CTS suffers more severely than the CSMA/CA when the remote hidden terminal problem occurs. The reason is that node A's RTS/data packet can be corrupted by node C's RTS/data packet with a high probability.

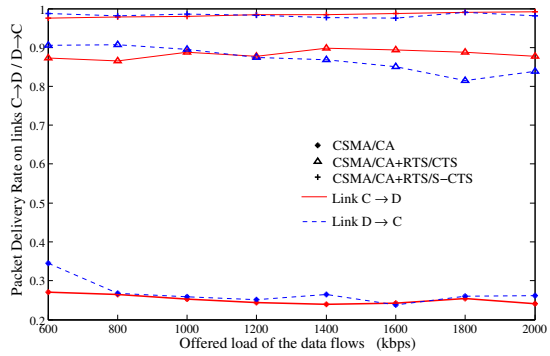
B. Low-SINR-CTS Problem Scenario

To simulate the low SINR scenario, we built up two unidirectional data flows (A→F and G→B) as Fig. 2. The distance between each pair of adjacent nodes is 480m. We injected two data flows concurrently at nodes A and G. We varied the data flow from 600Kbps to 2Mbps to study the throughput and packet delivery rate on each link.

The simulation reveals that: (1) For unidirectional flows, the RTS/S-CTS achieves 8~15% improvement compared with standard RTS/CTS, and 200~250% improvement compared with traditional CSMA/CA in each data flow throughput and total end-to-end network throughput (Fig. 13(a)). 2) By solving the CTS collision problem, the RTS/S-CTS can safeguard each



(a)



(b)

Fig. 13. Results in the Low-SINR-CTS scenario: (a) The throughput. (b) Packet delivery rate on link C→D/D→C.

link's packet delivery rate to be above 97%. More importantly, this guarantee would not be affected by the offered load of the data flows (Fig. 13(b)), and can significantly save the retransmission energy cost.

C. Random Network Topology Scenario

To evaluate the RTS/S-CTS's scalability and generality, we generated a random network topology with 16 nodes and randomly set up 6 links (Fig. 14). We varied the data flow on each link from 1Mbps to 4.5Mbps. For each data flow, we varied the packet size randomly. We run CSMA/CA, standard RTS/CTS and our RTS/S-CTS respectively. We measured the throughput of each link and summarized the average throughput of the six links in Fig. 15. The result shows that, when the data flow on each link exceeds 4Mbps, the RTS/S-CTS can improve the throughput of those affected links for more than 63% compared with CSMA/CA and standard RTS/CTS.

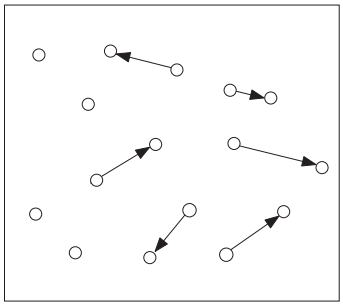


Fig. 14. The random network topology.

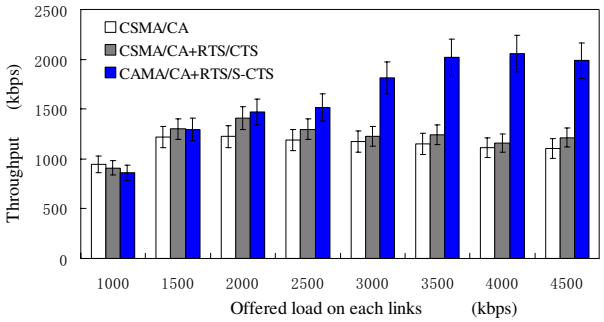


Fig. 15. The throughput in the random network topology scenario.

However, when the link's data flow is low (<1.5Mbps), the RTS/S-CTS becomes useless (Fig. 15). Furthermore, because of the overhead of the S-NAV field and the catalogue overhead, the throughput of RTS/S-CTS is even slightly below the standard RTS/CTS.

As a summary, we can see from the improvement of throughput that the RTS/S-CTS outperforms CSMA/CA and RTS/CTS. Moreover, Fig. 15 explains why standard RTS/CTS is just a "backup and supplement" mechanism to CSMA/CA [7], [19]. With high link workload, the standard RTS/CTS cannot protect the receiver from hidden terminals due to the CTS collision (Low-SINR-CTS problem). The RTS/S-CTS defeats those drawbacks by designing a new method to solve the hidden terminal problem.

VI. Discussions

We further discuss some issues arisen from the RTS/S-CTS mechanism that remain unaddressed in this paper:

(1) *Complexity:* Although using a larger number of different S-NAV indicators can reduce the catalogue overhead, it may also introduce computation overhead to conduct signal correlation of the incoming signal sample by sample. Fortunately, we use the preamble detection and synchronization (Fig. 5) to activate the SLDD. As the S-NAV field in the S-CTS frame has a constant size, instead of correlating all the incoming samples, we can just *cut* the appropriate S-NAV samples and do the correlation operation *only* for that set of S-NAV samples. Thus, the computation complexity of the PSCP is $\theta(cN)$, where N is the total number of S-NAV indicators adopted and c is a constant cost for conducting one signal correlation. Note that this computation complexity is also a constant cost when N is fixed. Comparing with the preamble detection that needs to correlate all the incoming samples with the preamble, this constant cost is significantly less than the preamble detection cost.

(2) *Self-Test and Cancelation:* The RTS/S-CTS is a cross-layer mechanism. This feature makes it easily compatible with old protocols, but this may bring new issues. Generally, each packet needs to pass the CRC self-test and will be abandoned when it fails the CRC checking. When the RTS/S-CTS is under a very low SINR scenario ($SINR < -20dB$), the S-CTS's false positive error rate becomes remarkably high (Fig. 16).

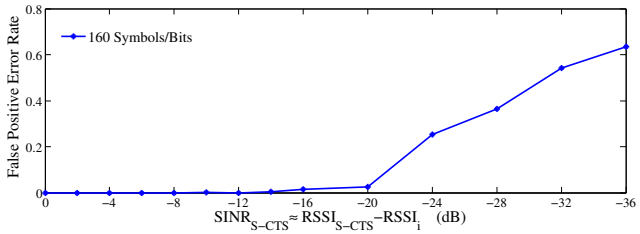


Fig. 16. False positive error rate with 160 symbol length.

Recall that the false positive error incurs because the station erroneously detects a S-NAV indicator from data. This will cause unnecessary time waiting. We propose a *self-test and cancelation mechanism* for the RTS/S-CTS to eliminate this problem: When the parameters of S-NAV indicators, such as length and minimum Hamming distance, have been settled, the maximal number of candidates that exceed the threshold β_{S-NAV} in the best candidate algorithm has also been fixed. We fix this number N_{S-NAV} for the SLDD (e.g., in Fig. 7, N_{S-NAV} is set to 3). If the PSCP generates more than N_{S-NAV} candidates, the SLDD would cease delivering the S-NAV time information to the MAC layer.

(3) *Impact of Packets Size:* The RTS/S-CTS mechanism is similar to the standard RTS/CTS in the MAC layer. This brings us another concern: the RTS/CTS packets may waste the wireless channel when the size of data transmission packets is small. In fact, the RTS/CTS mechanism is not activated unless the size of data packets exceeds a threshold. The standard RTS/CTS mechanism does not specify the value for this threshold, since it relates to many network parameters, such as network topology and network traffic patterns, which have been well studied [7], [19]. In this paper, we do not give detailed discussion on this threshold neither.

VII. Related Work

In this section we briefly review some prior related work. The hidden terminal problem has been well-studied in the past two decades. Most solutions to the hidden terminal problem work at the MAC layer. MACA [11] proposes a mechanism using the RTS/CTS exchange without carrier sense to reserve the wireless channel. MACAW [10] revises the MACA and uses the ACK packet to acknowledge the successful reception of data transmission. However, both of them assume the RTS/CTS exchange can perfectly received by hidden nodes, which is not likely the case in the wireless network for most of the time. Moreover, it introduces extra control overhead when the size of data packets is relatively small. Fullmer and Garcia-Luna-Aceves further proposed FAMA family MAC protocols (FAMA [20], FAMA-PJ [5] and FAMA-NCS/NPS [1]), which require the length of the RTS/CTS packets to be larger than a fixed size due to the awareness of RTS/CTS packet collisions. This partially solves the RTS/CTS collision. These MAC protocols relies on the virtual carrier sensing. However, they all suffer the Low-SNR/SINR-CTS drawbacks, which motivates us to resort to new solutions to the hidden terminal problem.

Xu et al. [3] revealed the remote hidden terminal problem and proposed two solutions, selective response to RTS request and directional antenna. The former solution requires that a CTS can be granted only if the RTS's energy level is higher than a threshold. As a result, this solution reduces more than half of the effective data transmission range, which sacrifices the network connectivity. Moreover, it cannot defeat the CTS collision problem at the same time. The directional antenna could be a solution to both remote hidden terminal and CTS collision problems. However, because the beam-width of directional antenna is narrow, it requires 5∼9 times more CTS retransmissions to cover the whole region, and may cause the jamming problem if they work with omni-antennas [21]. Also, it is costly to equip a directional antenna into wireless devices.

Recent studies exploited a new form of interference cancelation strategy [13], [15], [22]–[25]. Instead of avoiding collision, the new strategy tries to reconstruct the collided packets at the PHY layer by using some known symbol level information. Jamieson and Balakrishnan [13] proposed a partial packet recovery mechanism to recover the whole packet via SoftPHY. The SoftPHY interface collects bits information and requires the transmitter only to retransmit corrupted bits for saving wireless network bandwidth. ANC [25] provides an algorithm for canonical 2-way relay transmission; it doubles the capacity of typical 2-way network by designing an analog sample coding algorithm. But, it is based on the assumption that the receiver has already known one of the collision packets, and not suitable for the random network. ZigZag [15] works under 802.11 protocol scenarios and deals with general collisions. However, it can only perform well in the AP-Station mode and the collided packets require retransmitting multiple times.

Busy tone has been proposed in [26], [27] to silence the hidden nodes. In [26], the busy tone message is transmitted to hidden nodes through a separated control channel, which wastes the wireless spectrum. In [27], to improve spectrum efficiency, the wireless device sets up a full-duplex channel using two antennas. However, the transmitter and receiver have to stay within a short distance so as to decode the data signal correctly.

More recently, a new form of collision avoidance called CSMA/CN has been proposed in [14], [16]. Instead of collision avoidance, it uses a collision notification packet to send out the packet collision information, so that the transmitter can stop transmitting the data immediately. While it implements a kind of CSMA/CD mechanisms in wireless environments, it does not alleviate the hidden node problem, and would still be interfered by hidden nodes.

Comparing to the previous works, the proposed RTS/S-CTS requires no changes to the standard 802.11 MAC, has no constraint on the transmitter-receiver distance, and does not need expensive hardware such as directional antenna or full-duplex transceiver. Therefore, it is a practical solution that can solve the Low-SNR/SINR-CTS problems with low cost.

VIII. Conclusions

Comparing to the RTS/CTS mechanism, the RTS/S-CTS is more effective in silencing hidden terminals. We show its feasibility and performance improvement through both hardware implementation and software simulation. Furthermore, we analysis the complexity of RTS/S-CTS and propose a self-test/cancelation mechanism to defeat detection errors. All these efforts make the RTS/S-CTS more practical to real world network scenarios.

References

[1] C. L. Fullmer and J. J. Garcia-Luna-Aceves, "Solutions to Hidden Terminal Problems in Wireless Networks," in *ACM SIGCOMM*, 1997.

[2] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.

[3] K. Xu, M. Gerla, and S. Bae, "Effectiveness of RTS/CTS Handshake in IEEE 802.11 Based Ad Hoc Networks," in *IEEE Ad Hoc Networks*, 2003.

[4] P. C. Ng, S. C. Liew, K. C. Sha, and W. T. To, "Experimental Study of Hidden Node Problem in IEEE 802.11 Wireless Networks," in *ACM SIGCOMM* Poster, 2005.

[5] C. L. Fullmer and J. J. Garcia-Luna-Aceves, "FAMA-PJ: A Channel Access Protocol for Wireless LANs," in *ACM MobiCom*, 1995.

[6] L. B. Jiang and S. C. Liew, "Improving Throughput and Fairness by Reducing Exposed and Hidden Nodes in 802.11 Networks," *IEEE Trans. on Mobile Computing*, vol. 7, no. 1, pp. 34–49, 2008.

[7] M. Gast, *802.11 Wireless Networks: The Definitive Guide*. O'Reilly Media, 2002.

[8] K. A. Jamieson, "The SoftPHY Abstraction: From Packets to Symbols in Wireless Network Design," Ph.D. dissertation, MIT, 2008.

[9] E. A. Lee and D. G. Messerschmitt, *Digital Communication*. Kluwer Academic, 1993.

[10] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A Media Access Protocol for Wireless LAN's," in *ACM SIGCOMM*, 1994.

[11] P. Karn, "MACA-A New Channel Access Method for Packet Radio," in *the 9th ARRL Computer Networking*, 1990.

[12] J. Yao, T. Xiong, and W. Lou, "Elimination of Exposed Terminal Problem Using Signature Detection," in *IEEE SECON*, 2012.

[13] K. Jamieson and H. Balakrishnan, "PPR: Partial Packet Recovery for Wireless Networks," in *ACM SIGCOMM*, 2007.

[14] S. Sen, N. Santhapuri, R. Choudhury, and S. Nelakuditi, "Moving Away from Collision Avoidance: Towards Collision Detection in Wireless Networks," in *ACM HotNets*, 2009.

[15] S. Gollakota and D. Katabi, "Zigzag Decoding: Combating Hidden Terminals in Wireless Networks," in *ACM SIGCOMM*, 2008.

[16] S. Sen, R. R. Choudhury, and S. Nelakuditi, "CSMA/CN: Carrier Sense Multiple Access with Collision Notification," in *ACM MobiCom*, 2010.

[17] Ettus Inc, "Universal Software Radio Peripheral," http://ettus.com.

[18] GNU Radio, "GNU Radio - The Open Source Software Radio Project," http://gnuradio.squarespace.com/.

[19] IEEE Computer Society. 802.11, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," 2007.

[20] C. L. Fullmer and J. J. Garcia-Luna-Aceves, "Floor Acquisition Multiple Access (FAMA) for Packet-Radio Networks," in *ACM SIGCOMM*, 1995.

[21] Cisco, "Omni Antenna vs. Directional Antenna," in *Cisco Document*, 2007.

[22] D. Halperin, T. Anderson, and D. Wetherall, "Taking the Sting out of Carrier Sense: Interference Cancellation for Wireless LANs," in *ACM MobiCom*, 2008.

[23] S. Katti, D. Katabi, H. Balakrishnan, and M. Mdard, "Symbol-Level Network Coding for Wireless Mesh Networks," in *ACM SIGCOMM*, 2008.

[24] S. Zhang, S. C. Liew, and P. P. Lam, "Hot Topic: Physical-Layer Network Coding," in *ACM MobiCom*, 2006.

[25] S. Katti, S. Gollakota, and D. Katabi, "Embracing Wireless Interference: Analog Network Coding," in *ACM SIGCOMM*, 2007.

[26] Z. J. Haas and J. Deng, "Dual Busy Tone Multiple Access (DBTMA): A Multiple Access Control Scheme for Ad Hoc Networks Communications," *IEEE Trans. on Communications*, vol. 50, no. 6, pp. 975–985, 2002.

[27] N. Singh, D. Gunawardena, A. Proutiere, B. Radunovic, H. V. Balan, and P. Key, "Efficient and Fair MAC for Wireless Networks with Self-interference Cancellation," in *IEEE WiOpt*, 2011.