Hidden problems with the hidden node problem

Ashikur Rahman
Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada, T6G 2E8
Email: ashikur@cs.ualberta.ca

Pawel Gburzynski
Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada, T6G 2E8
Email: pawel@cs.ualberta.ca

***Abstract*— We discuss a few problems introduced by the RTS/CTS mechanism of collision avoidance and focus on the *virtual jamming* problem, which allows a malicious node to effectively jam a large fragment of a wireless network at a minimum expense of power. We propose a solution to this problem and provide experimental data illustrating the impact of virtual jamming and the effectiveness of our proposed solution.**

***Keywords:* Ad-hoc Networks, Broadcast, Medium Access Control, Collision Avoidance, IEEE 802.11.**

I. The hidden node problem

The notorious hidden node problem deals with a configuration of three nodes, like A, B, and C in Figure 1, whereby B is within the transmission range of A and C, while C is outside the range of A. In a situation like this, C will not be able to detect the ongoing transmission of A to B by carrier sensing and, consequently, it can inadvertently interfere with B's reception of A's packet. Two known solutions to this problem involve *physical carrier sensing* and *virtual carrier sensing*.

A. Physical carrier sensing

The *transmission range* of a node A is defined as the area inside which other nodes are able to correctly receive A's packets. On the other hand, the *carrier sense range* of A is the area encompassing those nodes whose transmission A can perceive (carrier sense) while not necessarily being able to receive the transmitted packets. Generally, it is unreasonable to assume that the two areas are always the same, e.g., the carrier sense range can be twice the transmission range [7].

Suppose that every node in Figure 1 has the same transmission range (represented by a solid circle). Node C is out of the transmission range of node A and thus would appear as a hidden node to B. However, if the carrier sense range of B is larger than the transmission range of A (see the dashed circle), B is hidden no more because it can sense the transmission of A and thus avoid interfering with it. This mechanism for eliminating the hidden node problem was proposed in [7].

Carrier detection is usually controlled by *thresholds* applied to the level of (actualy or apparently) perceived signal. Low thresholds tend to be sensitive to many factors involving more than the distance between the nodes, e.g., the natural noise level in the neighborhood. While formally, increasing the carrier sense range is possible, low thresholds may trigger many false indications, which will result in unnecessary backoffs and reduced throughput, possibly below one that could be achieved by simply ignoring the hidden node problem altogether.

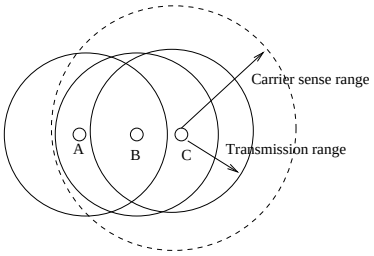


Fig. 1. A hidden node scenario.

B. Virtual carrier sensing

To alleviate the hidden node problem, Karn [5] proposed a two-way handshake involving short packets whose exchange should precede the actual transmission. The sender starts by transmitting a Request-To-Send (RTS) packet. After receiving RTS, the intended recipient sends a Clear-To-Send (CTS) packet to the sender. Both packets specify the length of time needed to transmit the actual data packet. Any third party node receiving any of the two packets will know for how long it should refrain from transmission as to avoid interfering with the exchange in progress. This protocol has been standardized into the popular IEEE 802.11 family of access schemes. The complete exchange involves four packets: RTS/CTS/DATA/ACK, with the first pair taking care of the hidden nodes, and the final ACK providing for reliable delivery (triggering retransmissions on failures). A node may use both the physical and virtual carrier sense mechanisms to determine whether it is allowed to transmit.

II. Problems with the RTS-CTS handshake

Consider Figure 2. Suppose that node A is the sender and node B is the receiver. Let R_A and R_B denote the respective transmission ranges of A and B, and P_A and P_B denote the corresponding *transmission areas*, i.e., the sets of points of the rectangle U covered by the circles with the radii R_A and R_B drawn around the nodes. Assume that any node v in U is

painted with one of the following colors:

green	if $v \in P_A - P_B$
red	if $v \in P_B - P_A$
yellow	if $v \in P_A \cap P_B$
white	if $v \in (P_A \cup P_B)'$

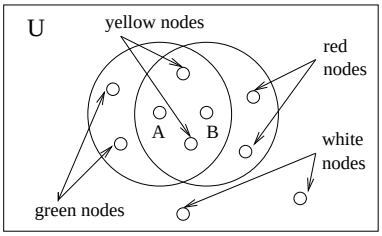


Fig. 2. Different areas of perception for a transmission from A to B.

A. Inhibiting non-interfering parallel transmissions

Suppose that A is transmitting to B (Figure 2). If a green node, (being outside the range of B) decides to transmit to a white node, its transmission will cause no damage at B; however, the green node is within the range of A, so it won't be able to receive anything while A is transmitting. Similarly, a red node (located within the range of B but outside the range of A) is technically able to receive (from the white nodes) while B is receiving from A. Only the yellow nodes are truly restricted: they must not transmit, and they are also unable to receive.

A node can determine its color with respect to an ongoing transmission as the involved nodes go through the RTS/CTS handshake. In our sample scenario (Figure 2), when A sends the RTS packet, it will be received by all green and yellow nodes. When B responds with CTS, all red and yellow nodes will be able to hear it. Thus, a node recognizing the RTS packet will paint itself green, a node recognizing the CTS packet will paint itself red, and a node that overhears them both will be painted yellow.[1]

In summary, the RTS/CTS mechanism, aimed at the elimination of hidden nodes, introduces a new problem: it hinders some non-interfering transmission that could be carried out in parallel. While eliminating hidden nodes will reduce collisions, thereby positively impacting the throughput, the elimination of some legitimate transmission will have the opposite effect. Although a scheme has been proposed in [2] to admit some parallel transmissions while avoiding the hidden node problem, it requires a significant modification of the IEEE 802.11's RTS/CTS mechanism.

B. False blocking

Consider the scenario shown in Figure 3(a). While A is transmitting to B, all green, yellow and red nodes are temporarily blocked. However, white nodes, being outside the range of both A and B, are free to transmit and receive. Suppose that a white node D is trying to transmit to a yellow node C. D will initiate the handshake with an RTS packet addressed to C (Figure 3(b)), but C (being blocked) will fail to respond with a CTS. D will assume that C is busy and will try later. The problem is that the ineffective RTS packet will paint all white nodes within D's transmission range green, and they will remain blocked for the entire time of the non-existent transmission, as announced by D. Notably, this *false blocking* [6] will propagate if some other white node tries to send something to any of the newly-painted green nodes. This is illustrated in Figure 3(c), with node E trying to reach node F.

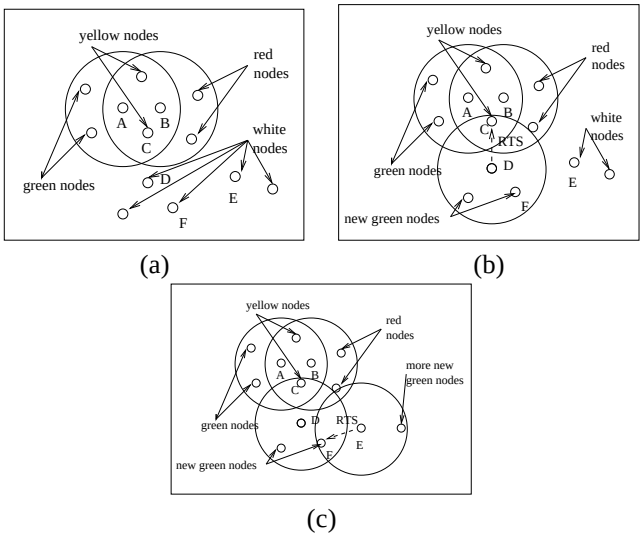


Fig. 3. False blocking.

C. Virtual jamming

False blocking is a prerequisite to a class of denial of service attacks against networks using the four-way handshake for collision avoidance. A malicious node can deliberately send (short) RTS packets at some intervals announcing long transmissions never to occur. This way the node will be able to effectively jam a possibly large segment of the network with a trivially small expenditure of power. The problem is not in the fact that a node can jam a wireless network (which can hardly be doubted), but that the amount of power needed to carry out this kind of attack can be trivially small [3], [4]. This *virtual jamming* is illustrated in Figure 4, where node M sends false RTS packets to node R with a large legitimate value in the duration field. When nodes G and H receive such a packet, they will both become blocked for the amount of time requested by M.

With one solution to this problem, suggested in [4] and dubbed *RTS validation*, a node receiving an RTS packet views it with a limited trust. The node only remains blocked until the time when, based on the expected timing of the CTS packet to arrive from the intended recipient, the sender should commence the transmission of the data packet. If the transmission does not happen, then the node unblocks.

[1]Note that in the algebra of colors yellow = green + red.

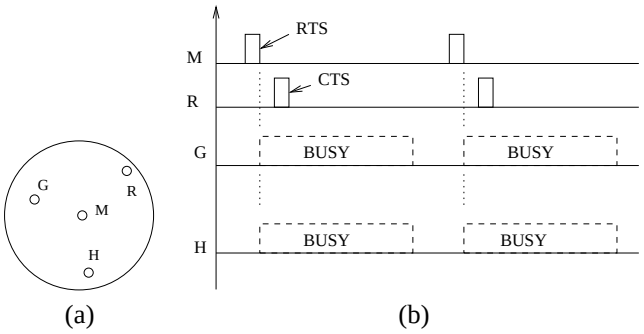


Fig. 4. Virtual jamming.

Otherwise, it will continue waiting for the remainder of the transmission time announced in the RTS packet by the sender.

Unfortunately, this solution brings only partial and illusory help. An attacker, being aware of this mechanism, can make sure to follow the RTS packet (after a pertinent delay) with a short dummy data packet. This way it will still trick the nodes into blocking at only minimally increased expense of power.

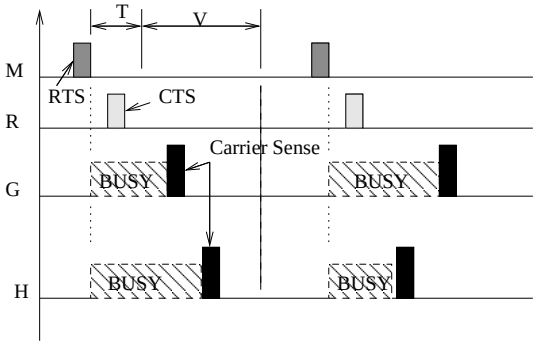


Fig. 5. Random RTS validation.

To combat this "advanced" variant of virtual jamming, we propose to include a random component in the RTS validation scheme. Specifically, instead of checking for the presence of transmission after a fixed interval following the reception of RTS, the node will do it at a random instant falling within the announced transmission window. Suppose that the RTS packet received at time t announces a transmission lasting V seconds. The data transmission is expected to start at $t + T$, where T covers the time after which the recipient should respond with CTS, and should continue until $t + T + V$. The node will partition the interval V into a number of equal size slots and pick one of those slots at random for verification. If the medium is busy, then the exchange is assumed valid; otherwise, the node will exit its blocking state. This mechanism is illustrated in Figure 5 (for the scenario shown in Figure 4). We call it *random RTS validation*.

III. Experimental results

To study the effect of (advanced) virtual jamming, we implemented an NS-2 model [1] of the two-dimensional static scenario shown in Figure 6. In this scenario, the inner nodes 1–4 form a clique. Each of the outer nodes 5–8 is only reachable from exactly one of the inner nodes. The traffic consists of three CBR flows (1024-byte packets): $2 \rightarrow 6$, $3 \rightarrow 7$ and $4 \rightarrow 8$. The transmission rate (sufficient to keep the network saturated) is 1Mbps for all three sources.

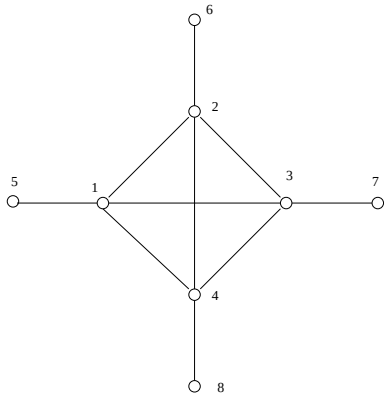


Fig. 6. The network topology of the simulation model.

Figure 7 shows the aggregate throughput achieved by all three streams without jamming. The aggregate throughput is almost steady and constant for the total duration of the experiment, except at the beginning, before the network has reached the steady state.

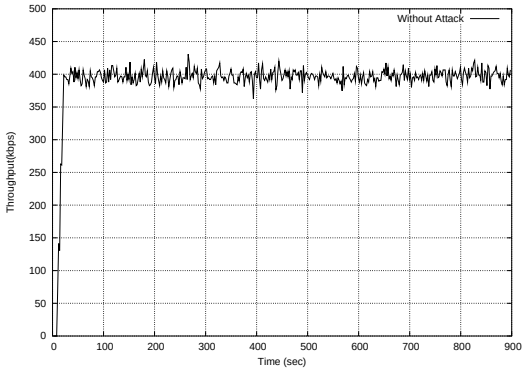
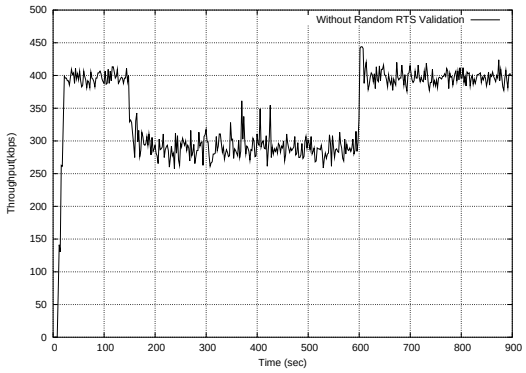


Fig. 7. Aggregate throughput (no attack).

In the second scenario (Figure 8), node 1 (the attacker) sends dummy RTS packets to node 5 at the frequency of a regular traffic source. Those RTS packets are received by nodes 2, 3 and 4 and cause them to become blocked. The attack begins at second 150 and continues until second 600, and the aggregate throughput drops by 25% during the attack.

Figure 9 refers to the third scenario. Node 1 launches the same attack as before (seconds 150 through 600), while the other nodes follow the *random RTS validation* procedure. Although the attack is perceptible in the aggregate throughput, the drop is almost insignificant.

In the last experiment, we vary the offered load to determine the impact of the attack, and the proposed countermeasure, on



(b)

Fig. 8. Virtual jamming attack launched by node 1.

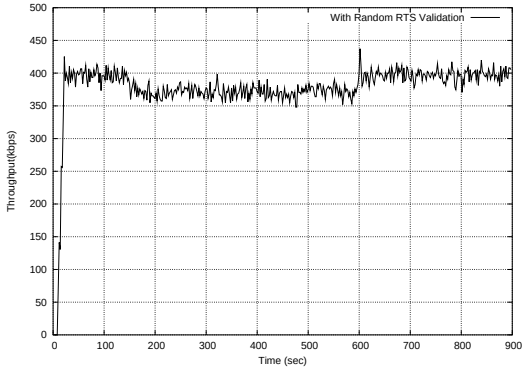
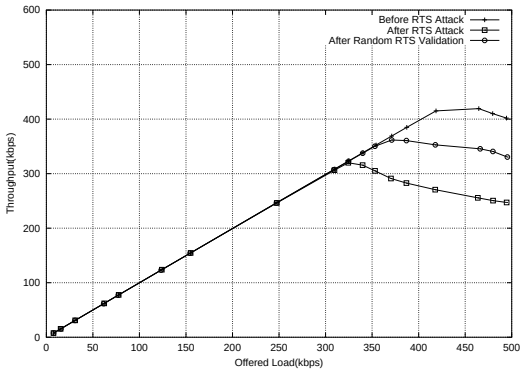
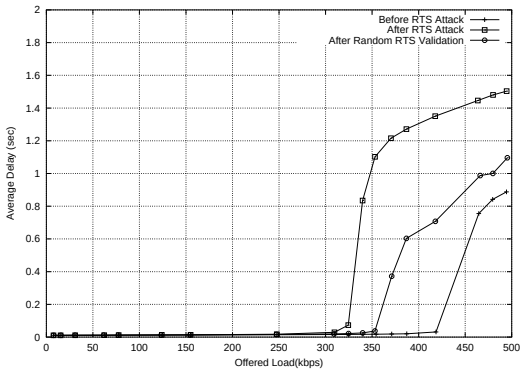


Fig. 9. Virtual jamming attack + random RTS validation.



(a)



(b)

Fig. 10. Virtual jamming attack under varying load.

the maximum throughput achievable by our network as well as on the delays. The results are shown in Figure 10. Under normal (no attack) conditions, the peak throughput is roughly 415 Kbps achieved at the offered load of approximately 450 Kbps (it drops slightly when the network becomes saturated). Under the attack (with no countermeasures), the maximum throughput drops to about 315 Kbps and then falls further as the load becomes heavier. With random RTS validation in place, we do see some drop in the maximum throughput, but not as large as in the previous case. A similar behavior is noted for the delay.

IV. Conclusion

The RTS/CTS handshake does not solve all problems related to medium access in the wireless environment: it trades some problems (like the hidden node problem) for others (inhibition of parallel transmissions and exposure to virtual jamming attacks). While elimination of the interference caused by hidden nodes does have a positive impact on the network performance, the problems introduced by the RTS/CTS mechanism will tend to counterbalance those benefits.

In this paper, we have discussed some of the known problems and added one more to the list. The "advanced" variant of virtual jamming allows an attacker to effectively block large regions of the network with a trivially small expense of power, despite some natural countermeasures taken by the cognizant nodes. To combat this kind of attacks, we have proposed a scheme called random RTS verification, which significantly reduces their impact.

One should note that a CTS attack is also possible, pretty much to the same effect as the RTS attack, except for a slightly worse power budget of the attacker. This is because the blocking time achieved with a false RTS packet is longer than that caused by a false CTS.

References

[1] The Network Simulator: NS-2: notes and documentation. http://www.isi.edu/nsnam/ns/.

[2] A. Acharya, A. Misra, and S. Bansal. MACA-P: a MAC for concurrent transmissions in multi-hop wireless networks. In *First IEEE International Conference on Pervasive Computing and Communications PERCOM*, 2003.

[3] J. Bellardo and S. Savage. 802.11 denial-of-service attacks: Real vulnerabilities and practical solutions. In *Proceedings of the USENIX Security Symposium*, August 2003.

[4] D. Chen, J. Deng, and P. K. Varshney. Protecting wireless networks against a denial of service attack based on virtual jamming. In *The Ninth ACM Annual International Conference on Mobile Computing and Networking (MobiCom) Poster*, September 2003.

[5] P. Karn. MACA–a new channel access methodfor packet radio. In *9th Computer Networking Conference on ARRL/CRRL Amateur Radio*, pages 134–140, September 1990.

[6] S. Ray, J. B. Carruthers, and D. Starobinski. RTS/CTS-induced congestion in ad hoc wireless LANs. In *WCNC*, 2003.

[7] K. Xu, M. Gerla, and S. Bae. How effective is the IEEE 802.11 RTS/CTS handshake in ad hoc networks? In *IEEE GlOBECOM*, volume 1, pages 17–21, November 2002.