Protecting Wireless Networks against a Denial of Service Attack Based on Virtual Jamming

Dazhi Chen, Jing Deng, and Pramod K. Varshney
EECS Dept., Syracuse University, Syracuse, NY 13244
Email: {dchen02, jdeng01, varshney}@ecs.syr.edu

I. Introduction

In the IEEE 802.11 MAC protocol [1], virtual carrier-sense and physical carrier-sense functions are used to determine the availability of the shared medium. The medium is considered idle only when both of these two functions indicate that the medium is idle. While the physical carrier-sense function uses the physical layer to sense the carrier, the virtual carrier-sense function is based on the Network Allocation Vector (NAV). Most IEEE 802.11 frames carry a duration field, which is used to reserve the medium for a fixed time period. The NAV is a timer that indicates the amount of time for which the medium has been reserved. Transmitting nodes set the NAV to the time for which they expect to use the medium, including the transmission time of all the frames in a sequence. Other nodes set up a process to count down the NAV. When the NAV is greater than zero, the virtual carrier-sense function indicates that the medium is busy. When the NAV reaches zero, the medium is reported to be idle. This mechanism, combined with the RTS/CTS exchange, is designed to reduce frame collisions and prevent the hidden terminal problem.

However, when nodes set up the NAV values, they do not know whether the expected frame exchange will actually take place. Furthermore, they do not verify if the NAV value has reserved a time that is indeed necessary for the current operation. These are vulnerabilities that a misbehaving node may exploit to block neighboring nodes from accessing the shared medium for an extended period of time. In this work, we investigate these vulnerabilities and potential *virtual jamming* attacks made possible by them. We also propose a backward-compatible solution to overcome these vulnerabilities.

II. The virtual jamming problem

Physical jamming may prohibit victim nodes from accessing the shared channel by the use of interfering signals. Similarly, due to the use of the virtual carrier-sense function in the IEEE 802.11 MAC layer, well-behaved nodes may be misled by misbehaving nodes to update their NAVs in such a way that they cannot access the shared channel. We term such an attack as *virtual jamming*. Since *virtual jamming* consumes much smaller amount of energy compared to physical jamming, it is more viable or efficient for misbehaving nodes or malicious nodes to launch such an attack. Virtual jamming can be realized by two possible types of attacks that exploit the vulnerabilities of the virtual carrier-sense mechanism, i. e. ,the *Spurious RTS/CTS attack* and the *NAV attacks*.

A. Spurious RTS/CTS attack

We define that a node is *falsely blocked* if it is prohibited from transmitting at some time instant without a genuine cause. Since a node that receives an RTS/CTS frame must defer its transmission, it is *falsely blocked* if the transmission of the corresponding data frame does not occur even though the medium is available. Because of this deferral mechanism, a misbehaving node may launch a Denial of Service (DoS) attack on a wireless network with the use of spurious RTS or CTS frames. That is, a misbehaving node may randomly send out a large number of spurious RTS or CTS frames addressed to a possibly non-existing node in order to *falsely block* other well-behaved nodes in the neighborhood, thereby successfully realizing virtual jamming. We term such an attack as a *spurious RTS/CTS attack*. Intuitively, the attack is more effective when the misbehaving node is mobile since it can *falsely block* more well-behaved nodes. In addition, we notice that *false blocking* may propagate to some nodes which are a limited number of hops away from a misbehaving node in the network. We term this phenomenon as *limited false blocking propagation*. Limited false blocking propagation may occur more frequently when a spurious CTS attack is launched on a network with high traffic load. Fig. 1 illustrates this scenario. In fact, in a wireless network without any misbehaving nodes, the *false blocking problem* induced by a failed RTS/CTS exchange may occur at some nodes so that the performance of a network degrades [2]. However, in addition to the method proposed in [2], this problem can be handled by a provision in the IEEE 802.11 specification [1], which allows nodes not receiving a frame header within a specified period to reset the NAV counter. Note that this problem will be much more serious if misbehaving nodes exploit this vulnerability and actively initiate an attack. The solutions in [2] and in [1] cannot solve the problem completely, especially when a spurious CTS attack is launched.

B. NAV attacks

The second vulnerability results from the fact that a node does not check the validity or correctness of the duration field on received frames. Let R_{RTS}, R_{CTS}, R_{DATA}, and R_{ACK} denote the requested reservation times corresponding to RTS, CTS, DATA, and ACK frames, respectively, that should be set to block neighboring nodes in order to complete the current operation. We note that the values of R_{DATA} and R_{ACK} are always constant, thus it can be easily detected if these

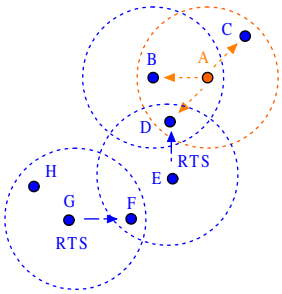

Fig. 1. Nodes B, C, and D overhear spurious RTS/CTS frames initiated by the misbehaving node A, and are thereby falsely blocked. Then Node F is falsely blocked by node E's RTS intended for node D. It is possible for node H to be falsely blocked by node G's RTS if node G sends an RTS frame intended for node F at the time before falsely blocked node F resets its NAV and so on. This demonstrates spurious RTS/CTS attack and limited false blocking propagation.

values are tampered with. However, misbehaving nodes can arbitrarily set the value of R_{RTS} and/or R_{CTS} to reserve the channel for an additional time due to the variable size of a data frame, thus *falsely blocking* those neighboring nodes by means of this type of virtual jamming. The maximum time that the channel can be reserved for in a single frame is limited by the size of the duration field, i. e. , $32767\mu s$ [1]. This is sufficiently large for misbehaving nodes to disrupt the network. We define these attacks as *NAV attacks*.

To eliminate these vulnerabilities and defend against virtual jamming, we propose a solution to allow well-behaved nodes to correctly update their NAV values with the help of the incoming frames. It is interesting to notice that traditional security methods such as key-based cryptographic solutions cannot solve this problem.

III. Protection from Virtual Jamming: The NAV Validation Scheme

Our protection scheme, termed *NAV Validation*, verifies the duration field of every overheard frame at the MAC layer. Obviously, an overheard DATA or ACK frame should be discarded if the reservation time R_{DATA} or R_{ACK} is not equal to the fixed constant.

More specifically, we set two MAC layer timers, the RTS timer and the CTS timer, to track the RTS-DATA and the CTS-ACK sequence. At any time, there exists at most one such timer at a node since it only verifies the largest channel reservation time requested by its neighbors. If a node does not receive the header of the corresponding DATA frame within an $RTS\_DATAHEAD\_Time$ specified by the RTS timer, the NAV will be reset to zero. Otherwise, the node should wait until the complete DATA frame is received, then compare the previous R_{RTS} value with the recalculated NAV value according to the actual size of the received data frame. The NAV counter will be updated to the correctly reserved channel time if they are not equal. Similarly, within a CTS_ACK_Time for a CTS timer, if no expected ACK frame is received, the NAV will be reset; otherwise, R_{CTS} will be verified and the NAV will be updated to the correctly reserved channel time accordingly.

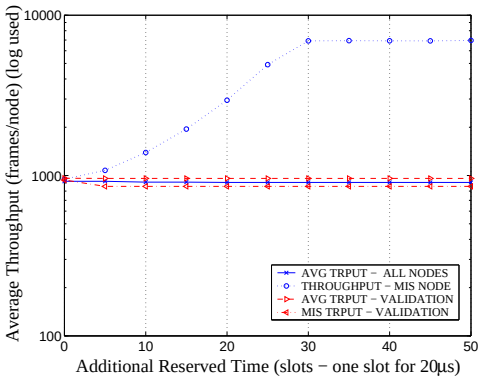

Fig. 2. Performance evaluation of a Wireless LAN under NAV attacks

In addition, a behavior table is utilized at every node to record the behavior of each of its neighboring nodes, and the node employs appropriate strategies on well-behaved and misbehaving nodes. One advantage of our proposed solution is that it is backward-compatible since it does not require any change in the frame format or frame exchange sequence.

We have performed a preliminary evaluation of the proposed protection scheme using the $NS2$ simulator. An IEEE 802.11-based wireless LAN with a base station as the common receiver and eight terminals as senders is simulated. One of the terminals is randomly selected as the misbehaving sender. All the senders are assumed to be backlogged and they are in range of each other. The misbehaving sender tries to *falsely block* other seven well-behaved nodes and monopolize the shared channel with NAV attacks. The simulation results are averaged over 8 runs and each simulation run is fed with a different seed. As shown in Fig. 2, our solution successfully prevents the misbehaving sender from monopolizing the shared channel and restricts its throughput to the estimated fair share. Note that additional reserved time in the figure denotes the redundant or extra channel reserved time a misbehaving node may add to the duration field of a MAC frame besides the reserved time that is genuinely required.

IV. Conclusions

The IEEE 802.11 MAC scheme may suffer from a number of vulnerabilities due to the use of the virtual carrier-sense function. In this work, we have proposed a NAV validation scheme to eliminate such vulnerabilities and to successfully defend against a Denial of Service(DoS) attack based on virtual jamming. Our preliminary results are quite encouraging. Our ongoing work includes the investigation of other consequences of these vulnerabilities and of our proposed solution through analytical and simulation methods.

References

[1] "IEEE Standard for Wireless LAN - Medium Access Control and Physical Layer Specification, P802.11," 1999.

[2] S. Ray, J. B. Carruthers, and D. Starobinski, "RTS/CTS-Induced Congestion in Ad Hoc Wireless LANs," in *WCNC*, 2003.