Bankrupting the Jammer

Farhana Ashraf, Yih-Chun Hu and Robin H. Kravets
University of Illinois at Urbana-Champaign
Email: {fashraf2, yihchun, rhk}@illinois.edu

***Abstract*— The high vulnerability of nodes in a WSN to jamming arises from the low resilience to jamming signals, easy differentiability of packet types and high predictability of wakeup schedules. In this paper, we propose `Jam-Buster` — a jam-resistant solution for a single channel WSN that increases resilience by using multi-block payload, eliminates differentiation by using equal size packets and reduces predictability by randomizing the wakeup times of the sensors. While each of these individual components is quite simple, the combination of the three components results in a jam-resilient system that forces the jammer to spend more energy to be effective and so reduce its own lifetime.**

I. Introduction

Due to the broadcast nature of the wireless medium, communication in any wireless network is inherently susceptible to radio interference — intentional or unintentional. Malicious attackers can take advantage of this fact and transmit jamming signals to interfere with ongoing transmissions. This results in an attack on the nodes' energy, wasting both the sender's and the receiver's energy and resulting in shorter network lifetime. While energy efficiency is a major concern in sensor networks, most existing protocols are not designed with consideration of the presence of jamming attacks. This omission leads to significant asymmetry of cost. Essentially, jamming is very cheap (i.e., requires very little energy investment from the jammer to disrupt an ongoing transmission) [**?**]. As a result, a jammer can waste significant energy of the sensors while spending only little energy of its own, enabling hammers with even limited energy to launch successful attacks against a wireless sensor network.

Several factors contribute to the effectiveness of such *cheap jamming* in sensor networks. First, due to the wireless channel, packets have very low *resilience* to interference. Essentially, a jammer only needs to disrupt a few bytes inside a packet to effectively jam the entire packet. The cost for transmitting these few bytes by the jammer are several orders of magnitude lower than the cost of the sender transmitting and the receiver receiving the entire packet. Second, the easy *differentiability* of data and control packets facilitates selective jamming. A jammer does not need to jam every packet to disrupt an ongoing transmission. Rather, a jammer only needs to jam a few well-selected packets, and so incur low transmission cost, and yet achieve the same effectiveness of jamming almost all packets. Third, with current periodic MAC protocols, a jammer can even *predict* future data transmission times. Thus, the jammer no longer needs to continuously listen to the channel to detect packets. By synchronizing its wakeup times with this pattern, a jammer can essentially duty cycle and still detect and jam all packets while incurring minimal idle listening costs.

Current solutions to defend WSNs against such cheap jammers are all based on hiding an ongoing packet transmission from reactive jammers. If a reactive jammer cannot detect a packet, the jammer cannot jam that packet. To reduce packet detection, the sender can use either a randomized Start-of-Frame Delimiter (SFD) [2] for each packet, channel surfing [3] or a combination of both [2]. However, to use both of these techniques, senders and receivers must be tightly synchronized to successfully transmit packets, which results in high synchronization overhead. Moreover, if the jammer has multiple radios and can listen to multiple channels at the same time, or if there are multiple jammers in the neighborhood, each listening to a different channel, channel surfing can no longer hide packets from the jammer. Overall, while these techniques may be able to hide some packets from the jammer with similar hardware capabilities, these solutions are quite expensive and have no defense against cheap jamming when the packets are detected.

Instead of trying to outsmart the jammer, we take a completely different approach that attacks the jammer and makes jamming more expensive. The contribution of our research is `Jam-Buster` — a low overhead jam-resistant framework that attacks the three factors that contribute to cheap jamming — resilience, differentiability and predictability.

II. Multi-Block Payload: Defense against Low Resilience

To enable low energy communication, packets in sensor networks are typically only protected by a CRC at the end of the packet. However, this leaves the packet susceptible to very short jamming signals that cause the entire packet to be discarded. The goal is to find a level of protection that is of low cost to the sender but of high cost to the jammer to jam. While there is no need for something as aggressive as error correcting codes [4], [5], additional checks can be added to the packets to help the receiver determine if there is any valid data in the packet. To this end, `Jam-Buster` is designed to use *multi-block payloads* within each packet, where each block is protected by its own CRC.

The multiple CRCs enable the receiver to independently verify each block for corruption. The receiver marks a block to be *jammed* only if the CRC for that particular block fails, and marks the block as *unjammed* otherwise. Essentially, the CRC for a block fails at the receiver if the jamming signal overlaps with any of the ℓ_b data bytes inside that block, or if

978-1-4577-0093-4/11/$26.00 ©2011 IEEE

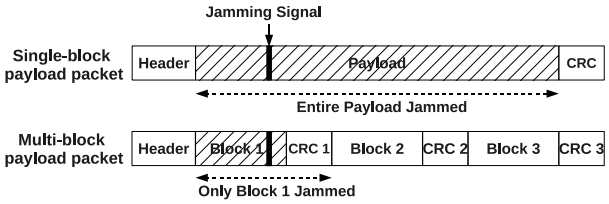

Fig. 1. Multi-block payload packet limits the jammer's effectiveness

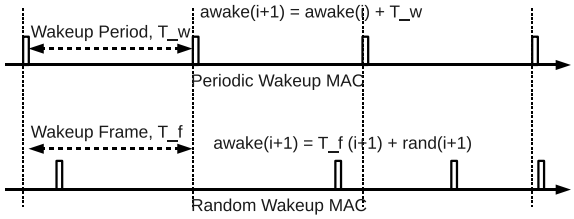

Fig. 2. Random Wakeup

the signal overlaps with any of the ℓ_c bytes of the CRC for that particular block (see Figure 1). Thus, an ℓ_j-byte jamming signal can disrupt only n_j blocks out of the k blocks, on an average:

$$n_j = \begin{cases} 0 & \text{if } \ell_j = 0 \\ \texttt{min}(\lceil \frac{\ell_j}{\ell_b+\ell_c} \rceil + 1, k) & \text{otherwise} \end{cases}$$

The CRCs for the remaining $(k - n_j)$ blocks succeed.

The total unjammed data bytes recovered from a packet, U_{mbp} depends on both the number of unjammed blocks inside a packet and the length of each block:

$$U_{mbp} = (k - n_j).\ell_b.$$

As k increases, the receiver can achieve more fine-grained recovery (i.e., more unjammed blocks) from a jamming signal. However, with a higher k, the total CRC overhead increases. As a result, the fixed sized packet can hold less data ($L - k.\ell_c$ bytes). Due to the negative effect k has on these two parameters, it is important for the sender to choose an optimal k to maximize U_{mbp} at the receiver. The main challenge lies in that U_{mbp} also depends on the length of the jamming signal. This type of approach naturally leads to a competition between the senders and the jammer, which we plan to model using game theory.

III. Look-alike Packets: Defense against Differentiability

Packet differentiation allows the jammer to make well-informed decisions about which packets to jam based on the packet type and the actual MAC protocol being used. To eliminate such differentiation, `Jam-Buster` uses two techniques to make all packets *look-alike*. First, all packets are the same size, regardless of the amount of data or the data type. Second, inter-frame spacing is randomized to prevent the jammers from using communication patterns to determine packet type. To complete the solution, all packets are encrypted to hide the type information inside the packet header.

With *look-alike packets*, a jammer cannot attack the operation of the MAC protocol by targeting only one type of packet. Essentially, `Jam-Buster` causes the jammer to jam all packets. However, this is very expensive and will quickly deplete the jammers battery.

IV. Random Wakeup: Defense against High Predictability

Duty-cycling is extremely effective at reducing energy consumption in sensor networks. However, the same techniques that make duty-cycled MAC protocols energy-efficient make them vulnerable to jammers. Essentially, if the receivers know when the senders are transmitting, so do the jammers. To break this correlation without reducing energy efficiency, `Jam-Buster` randomizes the wakeup times of the senders, which in turn randomizes the data transmission times.

Similar to other duty-cycled MAC protocols, each sensor divides time into slots called *wakeup frames*. All sensors have equal sized wakeup frames (T_f time units long). Unlike periodic MAC protocols, where each node wakes up at the beginning of its wakeup frames, a `Jam-Buster` node selects a random time within a wakeup frame and wakes up at that time (see Figure 2). For each wakeup frame, a sensor chooses a different wakeup time. A sensor wakes up at time a_n during its n-th wakeup frame. a_n is determined by adding a random offset, r_n, to the time at which the frame begins, $(n-1).T_f$:

$$a_n = (n-1).T_f + r_n.$$

The r_ns are linearly distributed random variables with values in the interval $[0, T_f]$, generated from a pseudo-random sequence. Each node initializes its pseudo-random generator with a random seed so that the r_ns for two nodes are not correlated.

This random wakeup scheme enables `Jam-Buster` to prevent a jammer from predicting the wakeup times of the nodes. Due to the random r_n sequence, it is not possible to predict the wakeup times based on the correlation between successive observed wake-ups of a node. Moreover, the different seed used in each node prevents the use of statistical analysis to infer the future transmission times by observing the behavior of multiple nodes. Since the jammer can no longer synchronize its wakeup times with the awake intervals of the sensors, any transmission that occurs when the jammer is asleep remains undetected by the jammer. Thus, the only way for a reactive jammer to detect and jam all packets is to remain awake all of the time. However, such a jammer incurs very high idle listening cost and thus has a very short lifetime.

V. Experimental Evaluation

The goal of our evaluation is to show that `Jam-Buster` can successfully improve the jam resilience of wireless sensor networks by forcing the jammer to spend more energy. Since the jammer's objective is to disrupt as much data as possible, and the objective of the sensors is to recover as much data bytes as possible from jamming, we evaluate `Jam-Buster`'s effectiveness by analyzing the total number of *unjammed*

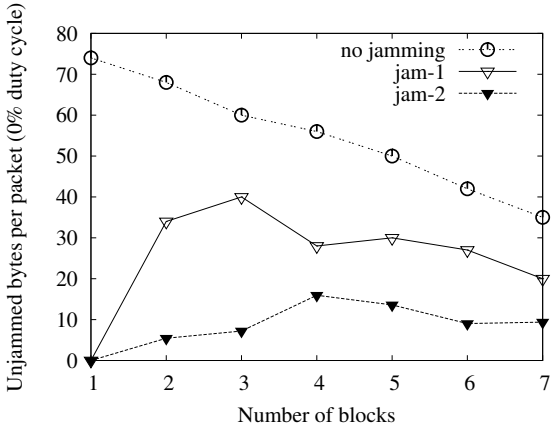

Fig. 3. Unjammed bytes per packet with multi-block payload

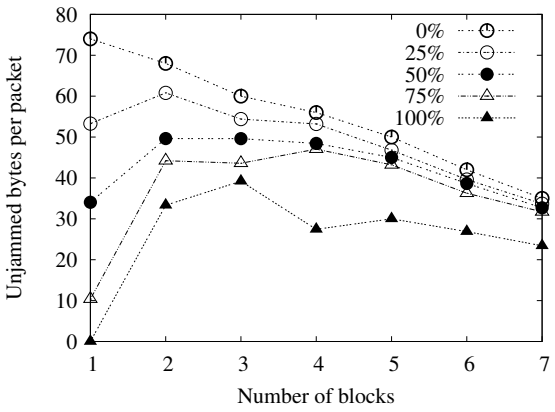

Fig. 4. Unjammed bytes with duty cycling `jam-1`

bytes. While a high number of unjammed bytes shows the effectiveness of `Jam-Buster`, a low number of unjammed bytes shows the effectiveness of the jammer.

We implemented `Jam-Buster` in TinyOS, and ran the experiments on `Tmote Sky` motes in a single hop network. For all evaluations, we assume that the jammer has the same power and energy limitation as the sensors in the network.

The sender generates CBR data, with inter-arrival time varying from 1 `sec` to 5 `sec` to 120 `sec`. All sensors (except the jammer) wakeup randomly maintaining a 1% duty cycle. Each sensor wakes up randomly for 5 `msec` within a 500 `msec` wakeup frame. In our experimental setup, we varied the jammer's duty cycle, d_j from 100% to 0% in steps of 25%. For the sender, we varied the number of blocks inside the packet from 1 to 7. The jammer in our experiments starts transmitting a jamming signal as soon as it detects a packet by listening for SFD in the channel. While `jam-1` uses 1 jamming strobe to jam a packet, `jam-2` transmits 2 jamming strobes to disrupt more blocks. During our experiments, we observed 11 bytes of unjammed data between each strobe due to hardware constraints.

For each experimental setup, we ran our experiment for 30 minutes, averaged the results over 5 runs.

Results

Without multi-block payload packet, i.e., with $k = 1$, the packet is extremely vulnerable to jamming. At this point, any jamming signal, irrespective of its length, can disrupt the entire packet (see Figure 3). As the senders increase k, the resilience to jamming improves. When the jammer is transmitting a single jamming pulse, the sender can retrieve 43% of its data by using just 1 more block. When the jammer transmits 2 jamming signals, the sender can still retrieve 6% of the data. The lower percentage, however, is due to more blocks being jammed by the additional jamming pulse. As the sender further increases k, it starts receiving more bytes due to more unjammed blocks. The total unjammed bytes increases even more as the jammer starts duty cycling to prolong its own lifetime and thus detects fewer packets (see Figure 4). However, as expected, for all cases, after the number of blocks crosses a certain threshold (for 1 jamming pulse, the threshold is 3 blocks, for 2 jamming pulses, the threshold is 4), the number of unjammed bytes per packet starts to decrease due to the reduced amount of data in each packet. This is due to the fixed overhead per block due to the CRC. Our evaluations show that the overhead is minimal. By paying a 16 byte overhead for two CRCs, the sensor can save 50% of its data when the jammer is transmitting 1 jamming pulse per packet, thus proving the effectiveness of `Jam-Buster`.

VI. Conclusion

In this paper, we propose a novel approach to defend against cheap jamming in WSN. Instead of proposing new anti-jamming solutions, our approach is based on attacking the jammer and forcing it to spend more energy to achieve effective jamming. Our multi-block payload forces the jammer to jam more bytes inside a packet. The look-alike packets forces the jammer to jam more packets. Finally, the random wakeup forces the jammer to remain awake longer. We have implemented `Jam-Buster` in TinyOS. Preliminary results from our testbed shows feasibility as well as effectiveness of our solution.

As the future work, we plan to model our system using game theoretic approach to analyze and determine the optimal number of blocks. We will also perform extensive experiments to evaluate `Jam-Buster`'s performance in different network and traffic conditions.

References

[1] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," in *MobiHoc*, 2005.
[2] A. Wood, J. Stankovic, and G. Zhou, "DEEJAM: Defeating energy-efficient jamming in IEEE 802.15. 4-based wireless networks," in *SECON*, 2007.
[3] W. Xu, W. Trappe, and Y. Zhang, "Channel surfing: defending wireless sensor networks from interference," in *IPSN*, 2007.
[4] J. Castura and Y. Mao, "Rateless coding over fading channels," *IEEE Communications Letters*, 2006.
[5] S. Howard, C. Schlegel, and K. Iniewski, "Error control coding in low-power wireless sensor networks: When is ECC energy-efficient?" *EURASIP Journal on Wireless Communications and Networking*, 2006.