Droplet: A New Denial-of-Service Attack on Low Power Wireless Sensor Networks

Zhitao He
SICS Swedish ICT
Box 1263
164 29 Kista, Sweden
Email: zhitao@sics.se

Thiemo Voigt
Uppsala University
and SICS Swedish ICT
Sweden
Email: thiemo@sics.se

***Abstract*—In this paper we present a new kind of Denial-of-Service attack against the PHY layer of low power wireless sensor networks. Overcoming the very limited range of jamming-based attacks, this attack can penetrate deep into a target network with high power efficiency. We term this the Droplet attack, as it attains enormous disruption by dropping small, payload-less frame headers to its victim's radio receiver, depriving the latter of bandwidth and sleep time. We demonstrate the Droplet attack's high damage rate to full duty-cycle receivers, and further show that a high frequency version of Droplet can even force nodes running on very low duty-cycle MAC protocols to drop most of their packets.**

I. Introduction

Wireless sensor networks are known to be vulnerable to jamming, a form of Denial-of-Service (DoS) attack that works by sending an interference signal to cause bit errors at the target node's PHY layer [1]. An implicit assumption about the high effectiveness of jamming is that a single jammer can put multiple sensor nodes within its transmission range out of service, implying that a small number of jammers are sufficient to disrupt an entire network consisting of many densely located nodes, causing a disproportional level of damage [2].

On the other hand, for most commercial and industrial WSN applications, any jamming attacker tends to be always limited to a certain level of transmission power, and might have limited battery capacity as well, thus leaving room for the target network to embrace power-efficient anti-jamming techniques to mitigate the damage. Researchers have focused on finding defense mechanisms for a single physical link, which can be evaluated by a set of metrics about the relative costs of the attacker and the pair of defenders [3]. The essential strategy of such anti-jamming techniques is bolstering nodes with extra data redundancy and incorporating more protocol intelligence while still keeping the average radio duty-cycle sufficiently low, in order to contain and outlast a potential jammer. This can trigger an arms race, as the attacker seeks to adjust its own radio duty cycle to gain relative advantage over its target, which in turn justifies an increase in the defense budget of the defending network [4].

The motivation for a high defense level against jamming is based on the aforementioned assumption that the damage level is multiplied when a single jammer affects a large area of densely deployed nodes. This assumption is, however, invalid in most practical scenarios. First, an intrusive, high power radio transmitter is hard to escape detection if it is planted in the central area of a network, e.g., inside a household or a factory. Another reason is that even a subverted node located in the heart of the deployment area tends to have a very limited *effective jamming range*, much smaller and fragmented than its transmission range. A jammer converted from a low power sensor node, such as one based on a 2.4 GHz IEEE 802.15.4 radio transceiver, can block a bidirectional link only when it is carefully placed between the communicating pair. Its effect on nodes further away usually amounts to a harmless rise in the noise floor.

Is it possible to launch a PHY layer DoS attack from a low power sensor radio at the peripheral of a network, which can disrupt the entire area within the radio's transmission range?

In pursuit of an answer, we discover a common vulnerability at a sensor node's radio receiver, whose built-in synchronization mechanism is easy to trigger using fake data frames. We construct a malicious attacker using a sensor node, who generates a short data frame to trick the targeted node's receiver into starting header synchronization and frame decoding. Furthermore, the receiver hardware's incapability to distinguish an authentic PHY payload size header from a forged one allows our attacker to use an empty payload to prolong the receiver's decoding process to a full-size frame length period, until the empty frame is eventually dropped due to a CRC error. Meanwhile legitimate data frames destined to the victim are dropped, and the attacker spends a large proportion of its time resting its radio in sleep mode, conserving energy for the next attack. Because the synchronization header is a shared PHY layer parameter across a network, a single attack frame can affect all nodes in idle listening mode within the attacker's transmission range, equivalent to an extended-range jam burst. Repeated transmission of such short attack frames are analogous to sporadic rain droplets falling on a house roof that deprives an inhabitant of sleep. We therefore call this new type of DoS attack the *Droplet attack*. Because Droplet attacks target the radio's idle listening mode rather than the active receiving mode, it disrupts communication without the need for high transmission power, unlike a jammer who must roar like a thunder storm in order to overwhelm the victim's received signal power. Furthermore, Droplet attacks cannot be mitigated by increasing node density in the target network.

A Droplet attack can strike a heavy blow to nodes operating in full duty-cycles, such as a high traffic gateway node or a normal sensor node scanning for neighbor announcements

or time synchronization beacons. Intermittent transmissions of the Droplet frames at a frequency of 225 Hz are sufficient to reduce the available bandwidth of such a node by over 90%. Once a node enters a low duty-cycle mode, e.g., periodic on-off cycles, however, it has a good chance to avoid the attack because it wakes up often during the relatively long interval between two intermittent Droplet frames. To increase Droplet attack's hit rate of such short wake-up periods, we boost the attack frequency to over 5000 Hz by packing more than 40 attack frames into a single full size frame and then transmitting it in a tight sequence, using a special cyclic transmission mode of the radio. The result is a dense falling of droplets, a.k.a the *Drizzle attack*, that forms a "rain curtain" too thick for most legitimate frames to get through. We show that the Drizzle attack is still more efficient than jamming because of its large range advantage.

Our contributions are the following. We construct both our attackers and target applications on a 2.4 GHz IEEE 802.15.4 radio. We show an ultra low power Droplet attacker operating at 5% radio duty cycle can incur 93% packet errors to full-duty cycle receivers. In contrast, it takes a continuous jammer over 100 times higher transmission power to attain a similar effect. We also evaluate the Droplet and Drizzle attacks against the highly robust, low-duty cycle ContikiMAC. While Droplet's damage effect is reduced to 10%, Drizzle can cause 90% packet errors.

This paper continues with an overview of related work in Section II. We resume our discussion about jamming's range limits in Section III, which leads to the design principles and implementation of Droplet and Drizzle attacks in Section IV and Section V. The effect of these attacks is summarized in Section VI. We draw our conclusions in Section VII.

II. Related Work

A taxonomy of DoS attacks in WSNs by Wood and Stankovic identifies 12 generic types of attacks across the PHY, datalink and network layers [5]. Three of those DoS attacks, namely *Flooding*, *Collisions* and *Jamming* are related to our work. Droplet attack's principle is conceptually similar to a TCP SYN flood [6], which forges intermittent light-weight communication requests that consume a disproportionally large amount of target resources. A TCP SYN flood causes DoS by the depletion of the memory pool holding particular protocol data structures; Droplet causes DoS by depletion of available channel bandwidth. In additional to bandwidth consumption, Droplet also exhausts the target's stored energy, mainly by deprivation of the target radio's sleep cycles. In contrast, a jammed victim normally displays an increase of active time, spent on excessive link and network layer retransmissions. Therefore jamming and Droplet are two PHY layer attacks of rather contrasting nature. Jamming works by "robbing" the receiver of signal quality, whereas Droplet works by "stealing" the receiver's bandwidth when there is no signal.

A survey from 2009 covers many WSN jamming techniques [1]. Our work is most related to those based on low power, duty cycled radios. Law et al.'s statistical jammer collects temporal patterns of the target's communication, which is dominated by MAC parameters, and then schedules bursty jams to collide with the target's transmissions [3]. Their invention of three jammer performance metrics, i.e. *Censorship rate* (percentage of message blocked), *Attrition rate* (percentage of extra energy cost) and *Lifetime advantage* have been adopted by others. We believe this metric set is incomplete without consideration of the effective jamming range, which determines the multiplier of a single jammer's damage effect to surrounding nodes. A small change in this range may tip the delicate power balance between a small number of attackers and the collective effort of many defenders.

Wood et al.'s DEEJAM studies four increasingly sophisticated jamming patterns using normal IEEE 802.15.4 radios [4]. They are the first to use empty-payload PHY headers to launch jam bursts. Inspired by their work, we further investigate the effect of header synchronizations in the process, and show that this effect can reach a range much larger than the interference effect of the headers.

Channel hopping, e.g., Wispernet [7] and packet fragmentation, e.g., DEEJAM and Jam-Buster [8] are among the most common techniques proposed for jamming mitigation. These proactive counter-measures all incur computation and communication overheads, which can be justified only in cases where the assumption about a large jamming range holds true. We are going to re-examine this assumption in the next section.

III. Range Limitations of Jamming

In the section we analyze two main factors that limit the effective range of jamming, namely near-far effect and hidden transmitter effect. This prompts us to look for an alternative attack that is unconstrained by such factors.

A. The 1-to-N Jamming Attack Model

Early studies of WSN jamming attacks assume that a malicious radio transmitter emitting high power, wide-band interferences is highly destructive to sensor nodes connected by low power links [2] [9]. Superior transmission power enables a single jammer to disrupt N surrounding nodes concurrently, the closer nodes suffering more severe disruptions. This intuitive 1-to-N attack model forms the theoretical basis for mitigative route-around measures, which leverage the redundant routing resources of a dense WSN to maintain network connectivity [10]. Researchers later shift their interest to defenses against jamming launched from common low power radio hardware, and implicitly inherit the same attack model, assuming a large ratio between the number of affected nodes and the jammer. They focus much on the economics of the war between two opposing sides, with the attacker rationing its transmission duty cycles to conserve limited energy capacity and the many defenders spending extra computation and bandwidth resources to bolster link robustness. Duty-cycled jammers include fixed duty-cycle burst transmitters such as DEEJAM's *pulse jamming* as well as MAC protocol-aware, speculative packet colliders such as Law et al.'s statistical jammers against S-MAC, LMAC and B-MAC [3] [11]. A relatively high unit cost for link strengthening is justified, based on the reasoning that the attacker is affecting many links in its range at the same time, thus posing a grave threat to rudimentary network services, such as route maintenance and time synchronization, in additional to lowering data throughput.

Counter-intuitively, a dense network with many nodes deployed at close ranges within each other tends to be more robust against jamming than a sparse network. This is because of the *near-far effect*, which dictates that a transmitter can capture a receiver at close range at presence of an interference from another concurrent transmitter afar [12]. Therefore, even if an attacker constructed by low power radios might be able to eavesdrop and inject messages from and to its many targets, it has very limited jam capability to cause extensive communication disruptions. We have been able to verify the near-far effect in a simple scenario consisting of three Zolertia Z1 nodes [13]. Under the attack of a continuous jammer [14] located only three meters away, the receiver switches into and out of the jamming range as another close-by transmitter adjusts its transmission power relative to the jammer. Attempts to jam a commercial or industrial WSN are likely to face obstacles just to first place a jammer close enough to the deployment, let alone to penetrate its signal from the edge of the network into the central area.

B. Packet Errors Caused by Constant Jamming

We want to understand packet errors caused by different levels of constant jamming. In order to minimize the effect of random interferences from coexistent radio transmitters, We carefully set up an isolated shared channel among the CC2420 radios [15] of three Z1 nodes. They are connected by SMA coaxial cables and passive components, which form an isolated propagation path. The transmitter and jammer's attenuated RF signals are superimposed by a 2-to-1 RF power combiner, before being divided by a 1-to-2 RF power splitter into two equal halves, each feeding into the receiver and a spectrum analyzer respectively [16]. We use the spectrum analyzer to calibrate the measured received signal strength (RSS), since the on-chip RSSI register readings are known to be inaccurate [17]. Our set-up shields the radios from a large proportion of external interferences, as illustrated in Figure 1.

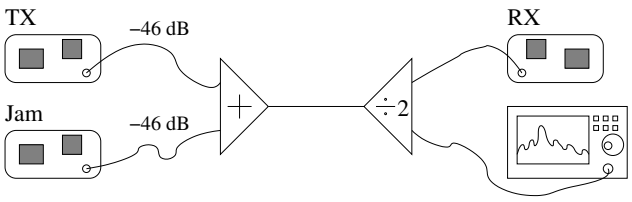

Fig. 1: Hardware set-up for packet error rate measurements.

We set the transmitter's power at a certain constant level, and adjust the jammer's power level to create different signal-to-interference-and-noise ratios (SINR) in range [-2 dB, +23 dB]. We then observe the packet error rate (PER) over 10,000 transmitted raw 802.15.4 PHY frames. We use a 20-byte frame length, as specified by the IEEE 802.15.4 standard for PER tests [18]. We generate two different jam waveforms, the unmodulated carrier and the modulated carrier, available as CC2420 TX test modes [14]. We show in Figure 2 the relation between the lower-portion of signal-to-interference-and-noise ratios (SINR) and respective PERs, the RSS being at -62 dBm. Continuous jamming attains 100% blocking at low SINRs, but packet errors decrease steeply from 100% down to close to 0% as soon as the SINR exceeds a certain threshold. For the modulated carrier, we observe a 3 dB threshold, which conforms perfectly to the CC2420 radio's nominal co-channel rejection ratio. We repeat the same test on three different RSS levels, emulating weak, intermediate and strong links, and the results are consistent. We can conclude that a jammer's effective range depends heavily on the interference power picked up by a specific receiver relative to the signal strength of another transmitter. In a dense network, a jammer's range therefore tends to be constrained and fragmented by the strong links formed between close neighbor nodes.

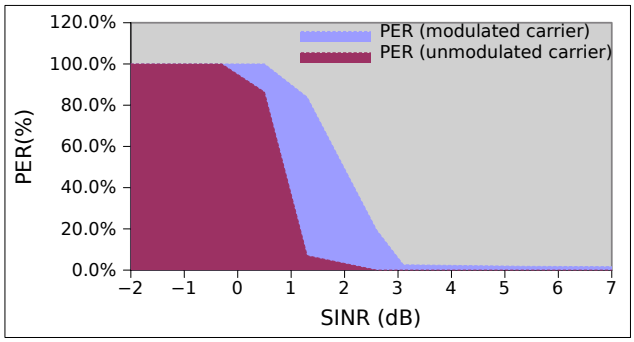

Fig. 2: SINR vs. PER under jamming. Jamming can block a link completely, but quickly loses effect when overpowered by a rival transmitter's signal.

C. Further Range limitations by Reactive jammers

Reactive jammers rely on detection of preamble or frame header to trigger jam transmission, such as DEEJAM's *activity jammer* and *interrupt jammer* [4]. They can be highly power-efficient compared with constant jamming, Its range is however further constrained by the *hidden transmitter* effect [12]. In such a scenario, a reactive jammer is close enough to the receiver, but still fails to interfere an incoming packet originated from a sender outside its sensitive range (Figure 3). Our hardware set-up enables us to hide the transmitter completely from the jammer by adding sufficient attenuation between them. We have been able to verify that the reactive packet jammer has a smaller range than a continuous jammer, by repeating an experiment in Section V of our previous work [19].

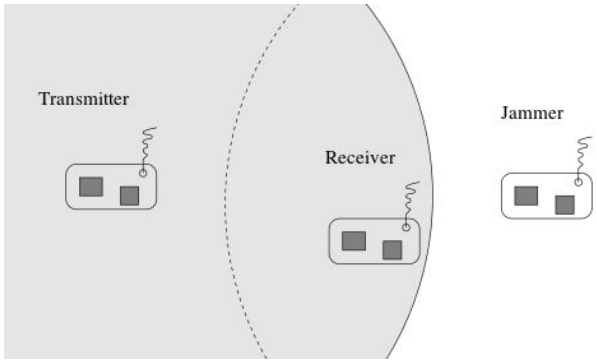

Fig. 3: A reactive jammer fails to interfere the signal sent to a close-by receiver from a remote, hidden transmitter.

IV. Principles of Droplet Attack

A truly efficient DoS attacker based on low power radios must overcome the range disadvantage compared with its targets. Utilizing its radio transmitter's full link budget would enable an attacker to launch disruptive attacks from a remote distance. We describe how broadcasting empty-payload frame headers allows us to disrupt strong links from a long range. We term this method Droplet attack, because of the small size of the headers and the low power level required for radio transmission.

A. Mysterious packet losses outside the jamming range

The PER measurements shown in the previous section highlights the range limitations of the two CC2420-based jamming methods. Because the modulated carrier mode has a slight range advantage over the unmodulated carrier mode, it has been used to create artificial background noise levels [20] and packet loss patterns [19] for experimental purposes. Our PER measurements with the presence of the modulated carrier interference reveal an intriguing phenomenon. When the SINR exceeds the 3 dB co-channel rejection level, i.e., the receiver is outside of the jammer's range, there continues to be a 2.5% PER, as shown in Figure 4. In an open air experiment, this small fraction of packet losses across the high SINR range would have been easily mistaken ed as losses caused by collisions with other coexisting channel activities. Since we have excluded most of the external interferences while obtaining the measurements, however, we need to find out the true reason for such packet losses.

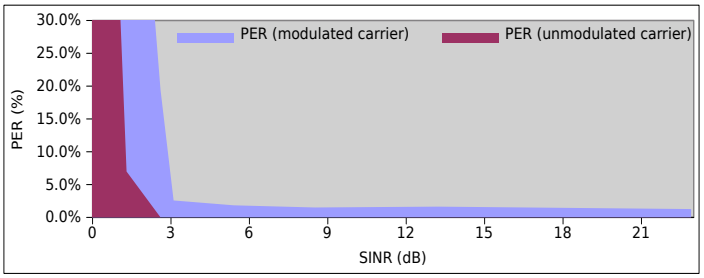

Fig. 4: Residual packet losses under modulated carrier mode.

B. Droplet attack

It turns out that these packet losses are caused by a different source than the continuous presence of a jamming signal. The 64 kb pseudo random data used to modulate the carrier happens to embed a few IEEE 802.15.4 *synchronization headers*. The latter is a constant bit sequence that precedes every frame. It contains a few zero preamble bytes followed by an 8-bit start-of-frame (SFD) header, which together synchronize a radio receiver to the transmitter. Detection of the synchronization header automatically starts decoding of subsequent data. To enable correct termination of frame decoding, a frame length field is often included at the beginning of the frame. IEEE 802.15.4 defines its PHY header to be such an 8-bit length field, so that a compliant receiver can use this information to buffer up a whole frame for later upper layer processing. The synchronization headers embedded inside the random data sent by a remote jammer can therefore accidentally trigger a receiver's synchronization and frame buffering mechanism, making it temporarily deaf to other concurrent transmitters. Only after the buffered frame with junk data is dropped, due to an CRC error, does the receiver become available again for reception. This explains why the random data jammer keeps incurring a small percentage of packet losses to the receiver, even when its signal drops to 25 dB lower than a rival transmitter.

We can design a malicious transmitter that exploits this vulnerability to disrupt communication around its neighborhood. By emitting a small, bogus frame that terminates immediately after a length field indicating a maximum frame size, such an attacker can trick a listening receiver into a lengthy but futile frame decoding process, disabling it from receiving from other nodes while it is trying to decode a shadow payload ensuing the length field from random noise. For IEEE 802.15.4 radios, decoding a maximum-size, 127-byte frame wipes more than $4\,\mathrm{ms}$ off the receiver's available channel time. By contrast, the attacker only needs to transmit a 6-byte frame header to cause the damage, using only 5% of active radio duty-cycles of its victim, while spending the rest of its own time in a low power sleep mode. Frequent transmissions of such headers can cause denial-of-service to nodes within the attacker's full transmission range. We term this attack the Droplet attack, because the small headers are analogous to intermittent rain droplets falling on a roof that creates a resounding disturbance through a room. We illustrate the mechanism of Droplet attacks in Figure 5, and shows the actual time scale of events triggered by a Droplet attack in Figure 6 from a signal trace captured from a logic analyzer [21].

Normally, further software processing is needed for the host MCU to read out the junk payload from the radio's hardware buffer and to eventually drop the frame upon detecting a CRC error. In this work, we do not try to quantify the extra software processing latency incurred by the Droplet attack, because it is dependent on the implementation of the radio driver and the speed of the communication bus between the MCU and radio. Instead we focus on measuring a victim node's radio receiver's deprived bandwidth and sleep time. Based on the maximum throughput of 250 kbps of a IEEE 802.15.4 radio, there are approximately 238 full length frame periods per second. By transmitting Droplet frames at a frequency close to 238 Hz, an attacker can repeatedly force a full-duty cycle radio receiver into receiving mode, thus depriving it of most of its available bandwidth. When the attacker's transmission range covers a large number of nodes densely deployed in its surrounding area, the damage is multiplied.

C. High Frequency Droplet Attack: Drizzle

The scenario becomes very different for networks running on a low duty-cycle MAC protocol. In such networks, nodes only wake up for a short period every long cycle to detect communication requests and exchange data. The wake-up period is usually set to a minimal length necessary for detecting a communication request. An attacker unsynchronized with its target's duty-cycling schedule thus hits the latter's short wake-up windows with only low probability. Even if a Droplet frame occasionally falls into such a wake-up period, it might still fail to lock up the receiver at the presence of competition from a

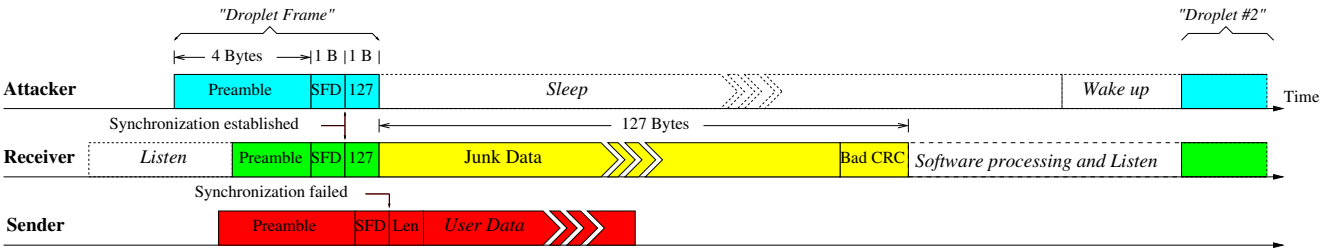

Fig. 5: The Droplet attack. The receiver picks up the attacker's 5-byte synchronization header first; after decoding the frame length, it starts decoding 127 more bytes, the maximum PSDU size mandated by the IEEE 802.15.4 standard. Meanwhile a concurrent sender's legitimate packet is ignored.

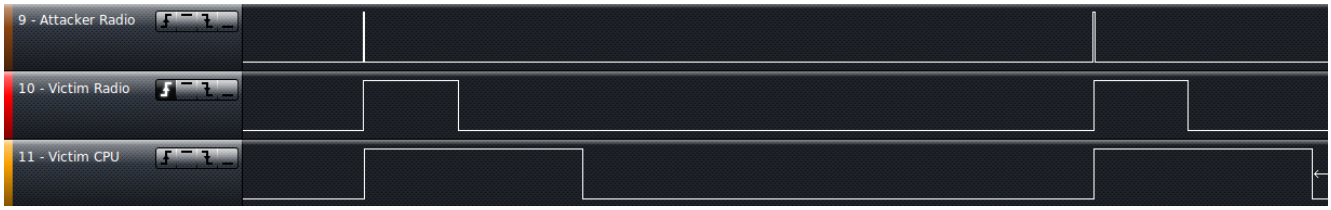

Fig. 6: Time traces of three interrupt signals during two Droplet attacks: 1. The attacker's SFD pin goes active during transmission of a Droplet header; 2. The victim's SFD pin goes active during its reception of the frame. 3. The victim's FIFOP pin goes active when frame buffering starts and goes down when the host MCU starts reading the frame out of the hardware buffer. The widths of the pulses indicate an event's time duration.

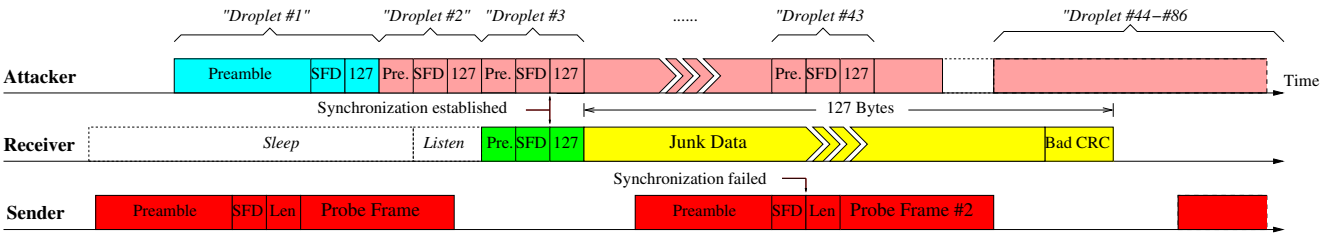

Fig. 7: The Drizzle attack against ContikiMAC. A small gap exists between two consecutive MAC probe frames that are used to synchronize with the receiver. When the receiver wakes up to detect an upcoming probe, it has a high probability to detect instead one of the Droplet headers embedded in the chain of almost continuous Drizzle frames, and thus fails to synchronize with the sender.

concurrent sender. For example, a sender running XMAC [22] or ContikiMAC [23] transmits a repeated frame sequence to ensure high detection rate by the intended receiver during its wake-up period. The tight intervals between these hand-shaking frames leave only very small gaps for any attack frames to sneak in and alter the protocol state.

In order to bolster the attack probability on low duty-cycle networks, we can increase the Droplet attack's transmission frequency. A minimum inter-frame interval required by the radio hardware however imposes an upper limit on how often the Droplet attack can be launched.

To eliminate the inter-frame interval, we can pack a succession of Droplet headers together into a single long frame. The droplet headers can be shorted to just three bytes by keeping only one of the four preamble bytes, therefore a 127-byte frame may contain up to 43 Droplet headers. Using CC2420's cyclic FIFO transmission mode, we can transmit this frame repeatedly in an almost seamless manner, plugging inter-frame time gaps needed for buffer refill. We thus can achieve a transmission rate beyond 5000 frames/sec for Droplet frames. This essentially turns the periodic droplets into a full duty-cycle "drizzle", whose temporal density helps it prevail in the contention for a receiver's first attention after waking up.

We illustrate the Drizzle attack in Figure 7. It transmits frames at a rate close to the maximum channel throughput, thus creates a similar interference effect as a continuous jammer to close-by nodes. Its real strength though, is reaching nodes far out in its transmission range, disrupting links otherwise are robust against jamming.

V. Implementation

In this section we discuss some important details of the implementation of the Droplet and Drizzle attacks using a CC2420 radio-based sensor node.

A. The Droplet Attack

We implement our Droplet attacker by customizing the CC2420 radio driver of Contiki 2.6. We first attempt to use the serial TX mode in CC2420 in order to have full control of the content of our Droplet frame at the bit level. Attaining this level of fine control, however, requires the implementation of an interrupt service routine with a latency smaller than the $4\,\mu s$ bit interval. Despite doubling the default clock frequency of the Z1's MSP430 MCU to 16 MHz, we find it difficult to guarantee a precise ISR latency to cope with the bit-rate interrupt signal from CC2420. This sometimes prevents us from sending a maximum frame length equaling 127 bytes, which leads to less bandwidth reduction to the target receiver.

We revert to the normal buffered TX mode, by writing the length byte into the TX FIFO and issuing a transmission command strobe to start the transmission of a Droplet frame. When the CC2420 detects that the length byte is not followed by any payload, a FIFO underflow occurs, terminating the transmission automatically and putting the radio back to sleep mode. After each transmission, we need to issue a command strobe to flush the TX FIFO in order to clear the FIFO underflow state. A further catch is that the TX termination switches off the power amplifier (PA) prematurely by a half byte period, corrupting the Droplet frame's crucial length header. In order to ensure the transmission of the full Droplet frame, we therefore append it with an extra bogus byte to prolong the termination of the radio's PA.

Using Contiki's high resolution *r(ealtime)timer*, we are able to regulate the transmission intervals between every two Droplet frames at a precision of $30\,\mu s$. A transmission frequency of 225 Hz has proved to be sufficient to severely cripple a full-duty cycle receiver.

B. The Drizzle Attack

The major bottleneck in our frequent Droplet attacks is the time needed by the radio hardware to prepare each transmission. Switching on the radio from sleep mode takes at least $128\,\mu s$. In order to plug the time gaps between consecutive Droplet frames for launching a seamless Drizzle attack, we leverage the CC2420's cyclic FIFO TX mode. This mode allows the user to fill the 128-byte TX FIFO with his own data; then the radio transmits the same data repeatedly disregarding TX FIFO underflows. The automatic repetition of transmissions of this mode suits our purpose perfectly, enabling us to generate a very high density of Droplet frames in time. We can encapsulate up to 43 Droplet frames, each only 3-byte long (one preamble byte, one SFD byte and one frame length byte), inside a single Drizzle frame. Each Droplet frame is thus spaced at just $96\,\mu s$ in time, much smaller than the smallest probe intervals of X-MAC ($4\,ms$) and ContikiMAC ($500\,\mu s$). This implies that a receiver that has just woken up from its sleep cycle has a large probability to encounter a Droplet header first, which forces it to drop the subsequent MAC probes from the transmitter.

VI. Evaluation

We evaluate the Droplet attack's effectiveness against a full-duty cycle receiver, whose strong link to a transmitter is disrupted by weak Droplet frames. We further launch the Drizzle attack against the low-duty cycle ContikiMAC, whose resilience against sporadic Droplet attacks yields under the pressure of its high frequency alternative.

A. Measurement set-up

We use the same hardware set-up as the previous jamming resistance measurements, replacing the jammer node with a Droplet attacker. In order to attain an accurate evaluation of the Droplet effect, we want to control unwanted packet losses caused by occasional collisions between Droplet frames and data frames sent by the transmitter. We carefully set the attacker's transmission power to just -25 dBm, resulting in -85 dBm RSS at the receiver, a level just high enough for the latter to drop less than 1% of Droplet frames.

B. Attacking a full-duty cycle receiver

A full-duty cycle receiver, e.g., the gateway node of a large network or a normal sensor node scanning for neighbor announcements or time synchronization beacons, is very vulnerable to Droplet attacks, because they spend a lot of time in idle listening mode. The attacker emits Droplet frames at a high frequency so that their shadow payloads forms a heavy burden on the victim's radio receiver.

We set our attacker to transmit Droplet frames at an average frequency of 225 Hz, each frame transmission separated by the average interval plus/minus a small time jitter. The transmitter's transmission power is adjusted from -25 dBm up to 0 dBm to generate different SINRs. At each SINR level, we conduct a PER measurement for 1000 transmitted 20-byte frames consisting of random payload data. In addition to the PER, we also measure the average *Correlation Errors* (CE) of correctly received frames in order to reveal the level of signal degradation. We obtain each received frame's CE value by deducting its LQI value from 108, the maximum LQI level we have ever observed from a CC2420 receiver.

We plot the result of the 225 Hz Droplet attacker together with data collected from the previous jamming measurements in Figure 8. A jammer fails as soon as its signal fades a few dBs below its rival. By contrast, Droplet's effect is independent of SINR, as indicated by the flat PER across the 25 dB range. The effective range of Droplet thus is the same as the radio's transmission range, which reaches a few tens meters indoors to a few hundred meters outdoors for IEEE 802.15.4. The graph also shows that, at the boundaries of the jammers' effective range where PER falls between 20% and 80%, many packets still get through despite a notable degree of signal quality degradation, as indicated by the correlation errors. This is because the radio receiver's built-in error recovery mechanism can tolerate a certain level of signal degradation. By contrast, the Droplet attack results close-to-zero correlation errors, because its short frames seldom waste energy colliding with those of its rivals.

An attacker located at the peripheral of a network can potentially penetrate deep into its targeted area, and only starts

to fade as its signal drops below the receiver sensitivity (-95 dBm for CC2420). When transmitting at 225 Hz, a Droplet attacker causes a high 93.5% PER to all full-duty cycle nodes within its range, disrupting normal network operations.

Improved radio sensitivity (e.g. Atmel AT86RF230 transceiver [24] or TI CC2591 external low noise amplifier [25]) normally does not alter a node's anti-jamming capability, because both signal and interference are amplified by the same amount on such a receiver. But such a measure would exacerbate the damage caused by Droplet attacks, as better sensitivity results in a larger transmission range of the attacker.

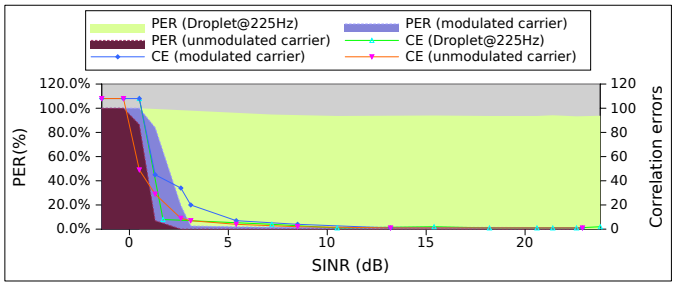

Fig. 8: Droplet attack's range advantage over jamming. At low SINRs, Droplet destructs a small amount of data frames by collisions just as jamming does. While jamming loses effect once SINR exceeds the 3 dB threshold, Droplet continues to cause a 93.5% damage rate.

In the next experiment, we set the transmitter's power at a constant level, resulting in a SINR of 22.6 dB, equivalent to a power ratio of 180 times between the transmitter and the attacker. We adjust the attacker's frequency in 25 Hz steps between 25 Hz to 225 Hz. For each attack frequency, we measure the average PER obtained over 5 repeated sequential transmissions of 1000 frames, using both 5-byte and 125 byte frame payloads. Figure 9 shows the relation between the droplet frequency and the resultant packet error rates. We see an almost linear relation between the attack frequency and packet errors, which is a very desirable property of an energy constrained attacker, who might want to regulate its attack rate to cause just enough disruptions without spending too much energy. We also see that Droplet attacks are equally effective against small and large frame sizes. By contrast, jamming tends to work against long frames more effectively than short frames, because long frames have a higher probability of being corrupted by an interference signal.

C. Droplet and Drizzle attacks on ContikiMAC

ContikiMAC is a sender-initiated, low-duty cycle MAC protocol. A node running ContikiMAC wakes up periodically from low power sleeping in order to check for channel activity, and accepts incoming data if it detects a packet from a sender. In a unicast, the sender transmits the data packet repeatedly until its intended receiver wakes up, detects the packet, and sends back an acknowledgment. The detail operations of the protocol can be found in its technical report [23]. Because a ContikiMAC node's wake-up period contains two consecutive clear-channel assessment (CCA) checks, each lasting a

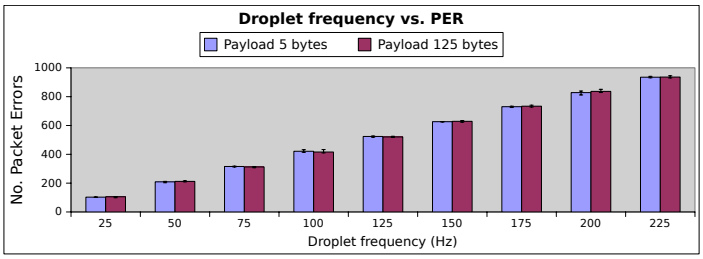

Fig. 9: A Droplet attacker can assert precise control over PER by adjusting its frequency. Both small and large data frames are attacked with the same effect.

mere 192 µs, it can keep a very low duty-cycle under a low throughput setting. This leaves only small chances for Droplet attacks to hit those short CCA check periods of a waken node and occupy its radio. We therefore try to increase the hit rate by attack the ContikiMAC using the high frequency Drizzle attacks.

We set up a pair of sender and receiver, both running on ContikiMAC with a wake-up frequency of 16 Hz. This is equivalent to about 6 ms awake time per second. The sender transmits a unicast packet to the receiver on an average frequency of 4 Hz. We disable higher layer retransmissions in order to avoid excessive data delivery delays and to simplify our PER analysis. We reuse our previous hardware set-up, but remove the spectrum analyzer in order to reduce asymmetry of the bidirectional path between the transmitter and receiver (Figure 10).

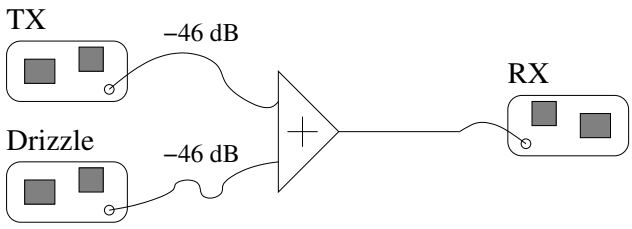

Fig. 10: Hardware set-up for Drizzle attacks against a pair of nodes running on ContikiMAC.

We program the sender to send four different sizes of packets in successive runs, ranging from the smallest size supported by ContikiMAC to 120-byte packets. In each run, we transmit 2500 packets in five batches, resetting the receiver between each batch to randomize the relative phase offset between the transmitter and the receiver. We set our Drizzle attacker's radio TX power to the lowest possible level, which is 22 dB weaker than the sender's. To verify that the almost seamless transmissions of Drizzle frames do not interfere the receiver at this low power level, we try to transmit the Drizzle frames using instead reverse-phase modulation [1], and confirm that the receiver gets 100% of the data packets at presence of such a low power noise spectrum. On the other hand, the large attenuation between the sender and the attacker ensures that

[1]By setting CC2420's MDMCTRL1 register's modulation mode bit

the former is hidden from the latter. Therefore, the sender's data packet transmissions are not throttled by the attacker's signal power. We repeat the measurements using a Droplet attacker with 225 Hz attack rate to provide a comparison with the Drizzle attacker. Figure 11 shows the results of of our attacks on ContikiMAC-based unicasts. While Droplet attacks can only cause about 10% packet losses, Drizzle achieves an average 90% PER with high probability. We can imagine that at such a high PER, all normal network services are essentially disabled.

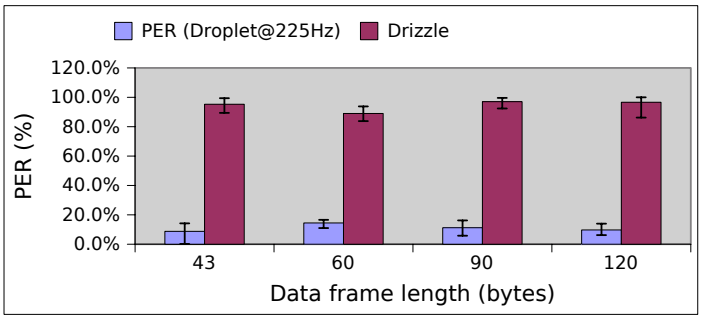

Fig. 11: Comparison of Droplet and Drizzle's attack effects to ContikiMAC. Drizzle achieves an average 90% PER, sufficient to break the strong link between the transmitter and the receiver.

We want to note that, under more general conditions, the sender has a large likelihood to also fall under the range of the Drizzle attacker, together with its intended receiver. In that case, packet losses are further aggravated when the sender's ability to perform carrier sensing and receive acknowledgments are hampered.

D. Defenses against Droplet and Drizzle attacks

A few techniques can be employed to mitigate Droplet and Drizzle attacks. First, hardware assisted address header decoding can be enabled to filter out frames decoded with a mismatching destination address to the local address. But the radio receiver usually still has to wait until the indicated frame length duration has elapsed, because the frame can actually be a valid one destined to another node [2]. Therefore this mechanism can only reduce some software processing from the host processor, but cannot recover receiver bandwidth lost due to a Droplet frame.

Channel switching can avoid a single-channel attacker. Nevertheless, because the number of available channels is limited, a group of attackers can co-ordinate to occupy each of the channels respectively.

Using a customized SFD among the protected network, as DEEJAM's frame masking method has shown, is an effective counter-measure against Droplet attacks. Such a custom SFD is hardly a well kept secret, however, because the number of possible bit-sequences for a good SFD is very small. Furthermore, using custom SFDs hinders inter-operation of devices from different vendors; most commercial radio chips support only the standard SFD. The same problem applies to other non-standard techniques, such as changing the modulation.

[2]"If a frame is rejected, CC2420 will only start searching for a new frame after the rejected frame has been completely received (as defined by the length field) to avoid detecting false SFDs within the frame." *–CC2420 datasheet*

VII. Conclusion

We have re-examined the common jamming attack model for low power wireless sensor networks. Jamming attacks have significant range limitations, which make them infeasible in most realistic scenarios. We propose instead the Droplet attack, which is based on common low power sensor radios, but can launch effective DoS attacks to all nodes within its transmission range. We have demonstrated its effectiveness against full duty-cycle receivers, marked by a 93.5% packet error rate. Based on the Droplet attack's principle, we further design a high frequency alternative, the Drizzle attack. We show that Drizzle attacks can disrupt even strong links between very low duty-cycle sensor nodes to great effect.

Acknowledgments

This work has been supported by the Uppsala VINN Excellence Center for Wireless Sensor Networks (WISENET) and by SSF.

REFERENCES

[1] A. Mpitziopoulos, D. Gavalas, C. Konstantopoulos, and G. Pantziou, "A survey on jamming attacks and countermeasures in wsns," *Communications Surveys & Tutorials, IEEE*, vol. 11, no. 4, pp. 42–56, 2009.

[2] A. Wood and J. Stankovic, "Denial of service in sensor networks," *Computer*, vol. 35, no. 10, pp. 54–62, 2002.

[3] Y. Law, P. Hartel, J. den Hartog, and P. Havinga, "Link-layer jamming attacks on S-MAC," in *Proceeedings of the Second European Workshop on Wireless Sensor Networks*, 2005, pp. 217–225.

[4] A. Wood, J. Stankovic, and G. Zhou, "DEEJAM: Defeating Energy Efficient Jamming in IEEE 802.15.4-based Wireless Networks," in *The IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (IEEE SECON)*, Jun. 2007.

[5] A. Wood and J. Stankovic, "A taxonomy for denial-of-service attacks in wireless sensor networks," *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*, pp. 739–763, 2004.

[6] W. Eddy, "Tcp syn flooding attacks and common mitigations," *IETF RFC 4987*, 2007.

[7] M. Pajic and R. Mangharam, "Anti-jamming for embedded wireless networks," in *Information Processing in Sensor Networks, 2009. IPSN 2009. International Conference on*. IEEE, 2009, pp. 301–312.

[8] F. Ashraf, Y. Hu, and R. Kravets, "Bankrupting the jammer," in *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2011 8th Annual IEEE Communications Society Conference on*. IEEE, 2011, pp. 149–151.

[9] A. Perrig, J. Stankovic, and D. Wagner, "Security in wireless sensor networks," *Communications of the ACM*, vol. 47, no. 6, pp. 53–57, 2004.

[10] A. Wood, J. Stankovic, and S. Son, "Jam: A jammed-area mapping service for sensor networks," in *Real-Time Systems Symposium, 2003. RTSS 2003. 24th IEEE*. IEEE, 2003, pp. 286–297.

[11] Y. W. Law, M. Palaniswami, L. V. Hoesel, J. Doumen, P. Hartel, and P. Havinga, "Energy-efficient link-layer jamming attacks against wireless sensor network mac protocols," *ACM Trans. Sen. Netw.*, vol. 5, no. 1, pp. 1–38, 2009.

[12] T. S. Rappaport *et al.*, *Wireless communications: principles and practice*. Prentice Hall PTR New Jersey, 1996, vol. 2.

[13] Zolertia, "Z1 Datasheet (rev. c)," 2010. [Online]. Available: http://www.zolertia.com/ti

[14] C. A. Boano, Z. He, Y. Li, T. Voigt, M. Zuniga, and A. Willig, "Controllable Radio Interference for Experimental and Testing Purposes in Wireless Sensor Networks," in *Proceedings of the 4th International Workshop on Practical Issues in Building Sensor Network Applications (SenseApp)*, Zurich, Switzerland, Oct. 2009.

[15] Texas Instrument, "CC2420 Datasheet (rev. 1.41b)," 2007. [Online]. Available: http://www.ti.com/

[16] Rigol, "DSA1030 3 GHz Spectrum Analyzer)," visited 2012-12-13. [Online]. Available: http://www.rigolna.com/products/spectrum-analyzers/dsa1000/dsa1030/

[17] Y. Chen and A. Terzis, "On the mechanisms and effects of calibrating rssi measurements for 802.15. 4 radios," *Wireless Sensor Networks*, pp. 256–271, 2010.

[18] "IEEE standard 802.15.4," IEEE Computer Society, New York, NY, USA, Oct. 2003.

[19] Z. He and T. Voigt, "Precise packet loss pattern generation by intentional interference," in *5th International Workshop on Performance Control in Wireless Sensor Networks*. IEEE, 2011, pp. 1–6.

[20] C. Boano, T. Voigt, C. Noda, K. Römer, and M. Zúñiga, "JamLab: Augmenting Sensornet Testbeds with Realistic and Controlled Interference Generation," in *Proceedings of the 10th international conference on information processing in sensor networks (IPSN)*, 2011.

[21] Saleae, "Saleae Logic 16 logic analyzer," visited 2013-03-13. [Online]. Available: http://www.saleae.com/logic16

[22] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks," in *Proceedings of the International Conference on Embedded Networked Sensor Systems (ACM SenSys)*, Boulder, Colorado, USA, 2006.

[23] A. Dunkels, "The ContikiMAC Radio Duty Cycling Protocol," Swedish Institute of Computer Science, Tech. Rep. T2011:13, Dec. 2011.

[24] Atmel, "AT86RF230 Datasheet (rev. E)," 2009. [Online]. Available: http://www.atmel.com/

[25] Texas Instrument, "CC2591 Datasheet (rev. A)," 2008. [Online]. Available: http://www.ti.com/