UPPSALA
UNIVERSITET

IT 15 028
Examensarbete 15 hp
Juni 2015

Attack on the Chaos Sensor Network Protocol

Oscar Gerbert

Institutionen för informationsteknologi
Department of Information Technology

UPPSALA
UNIVERSITET

Teknisk- naturvetenskaplig fakultet
UTH-enheten

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
http://www.teknat.uu.se/student

Abstract

Attack on the Chaos Sensor Network Protocol

Oscar Gerbert

As the demand for wireless sensor networks increases the need for new protocols with specific ways of distributing data emerges. Chaos is one of those protocols. Chaos has no native security countermeasures implemented, therefore it is important to test how vulnerable it is against attacks. In this thesis I present four novel attacks to test the robustness of Chaos. Experiments show that a Drizzle-attack was the most effective attack, strategic placement of the nodes was the key to a more efficient attack.

Handledare: Kasun Hewage
Ämnesgranskare: Thiemo Voigt
Examinator: Olle Gällmo
IT 15 028
Tryckt av: Reprocentralen ITC

Acknowledgements

First off I want to thank Thiemo Voigt for introducing me to the area of wireless sensor networks. It has been a very challenging and fun experience. Secondly I want to thank my supervisor Kasun Hewage for all the support and expertise throughout this project.

Contents

List of Figures

Chapter 1

Introduction

Every day more and more wireless sensor networks (WSN) are constructed. A WSN is a network where some physical or environmental data needs to be collected, for example temperature or humidity. The nodes in a WSN are often resource constrained to keep costs down and in some cases prolong battery-life. The increased demand of WSNs means the emergence of new and more complex protocols for data distribution, one of those protocols is Chaos [6]. Chaos is an efficient protocol for all-to-all data sharing. Chaos is the first protocol for WSNs with native support for all-to all data sharing. Thanks to its key techniques, synchronous transmissions and user-defined merge operators, Chaos basically parallelizes collection, processing and dissemination inside the network.

Chaos is based on two key techniques:

- *Synchronized transmissions:* All nodes that want to transmit data does so synchronously. Thanks to the capture effect several nodes are able to transmit at the same time and nodes overhearing one transmitting node will receive its packet. Nodes who receive packets successfully merge their own data with the new data and then send the merged packet synchronously.

- *User defined merge operators:* Each node merges its own data and the newly received data with a predefined merge operator specified by the user. By allowing the user to specify their own merge operators it makes Chaos more versatile. This technique is not the focus when attacking the network since the merge operations is performed at a higher level than the attacks.

Attacks focused on disrupting the availability has been performed on Glossy [2] which is the flooding architecture utilized in Chaos. Glossy proved to be fundamentally robust against attacks aimed to break constructive interference [10] thanks to tightly timed synchronized transmissions [4]. Chaos differs from Glossy when it comes to packet content. In Chaos packet content varies so it is not able to utilize constructive interference instead it solely relies on the capture effect [8] when receiving packets. The capture effect is a mechanism that allows the reception of a packet even with multiple nodes transmitting. There are no security mechanisms implemented in Chaos,

therefore it is important to test how vulnerable it is against attacks. The attacks I suggest in Chapter 3 are aimed to disrupt the flow inside the network, so called denial-of-service attacks.

The attacks I suggest are: breaking the capture effect by executing a Droplet-attack [3]and the synchronized transmissions by modifying the relay counter which Chaos uses for time synchronization. Synchronized transmissions will also be attacked by modifying the bit mask in the header field, that Chaos uses to check that data from every node has been merged together, to an incorrect one.

Evaluation of the four different kinds of attacks is done on experiments performed in the FlockLab [7] testbed. FlockLab was one of three testbeds used by Landsiedel et al. in the original evaluation of Chaos. The evaluation of the selected attacks showed that the most effective way to disrupt networks using Chaos was to execute a Droplet-attack. More attackers affected the network more, with strategic placement of the attacking nodes. Modifying the bit mask in the header field resulted in increased duty cycle. It also had an increase in average number of transmissions when setting the bit mask to zero for every node. When the bit mask was set to one for every node the result was a decrease in average number of transmissions. The average number of transmissions was so low network-wide that there was possibility of all nodes not contributing the final packet.

With this thesis I will make the following contributions:

- Present attacks aimed to break the capture effect and the synchronized transmissions. The key techniques, in Chaos, when receiving and transmitting packets.
- Evaluate the effectiveness of selected attacks in a testbed.

The rest of the thesis will look like this: In chapter 2 I will give background information about the 802.15.4 IEEE standard, Chaos, the capture effect, Glossy and constructive interference. In Chapter 3 I will describe four ways of attacking Chaos. In Chapter 4 I present the experiments and the results of those experiments. The result and how it affects networks using Chaos will be discussed in Chapter 5. Chapter 6 will present related work and Chapter 7 will conclude this thesis.

Chapter 2

Background

In this chapter I will provide background information on how the 802.15.4 IEEE standard works and how the capture effect and constructive interference works in general. I will describe how Chaos works and also the essential parts of Glossy that are utilized in Chaos. To understand the whole concept of Chaos I also give a short presentation of the user defined merge operators.

2.1 802.15.4 IEEE 2.4 Ghz

The 802.15.4 IEEE standard uses Direct Sequence Spread Spectrum (DSSS) together with Offset-Quadrate Phase Shift Keying (O-QPSK) technique over the 2.4 GHz band. The 2.4 GHz band allows for up to 16 different channels and transmission rates up to 250 kbit/s. The 802.15.4 IEEE standard consist of two layers, Medium Access Control (MAC) and a physical layer (PHY). The MAC is a management interface, manages access to the physical medium and makes sure timeslots are available. The PHY handles the transceiver, channel selection, energy and signal functionality. The layer that transmits the actual data is PHY.

2.2 Capture effect

Thanks to the Capture effect a node is able to receive a packet successfully even when multiple nodes are transmitting. It is able to receive the packet successfully due to it having higher signal-to-noise ratio (SNR) over other packets. For example in Figure 2.1(1) packet A arrives earlier than packet B and has higher SNR then transmission will be successful. In Figure 2.1(2) packet A arrives later than packet B and with packet A having higher SNR then packet A becomes lost. A higher SNR packet may arrive up to 160 μs later than that of a lower SNR packet. 160 μs is the air time for the 802.15.4 IEEE synchronization header. This is also called the preamble of the packet.

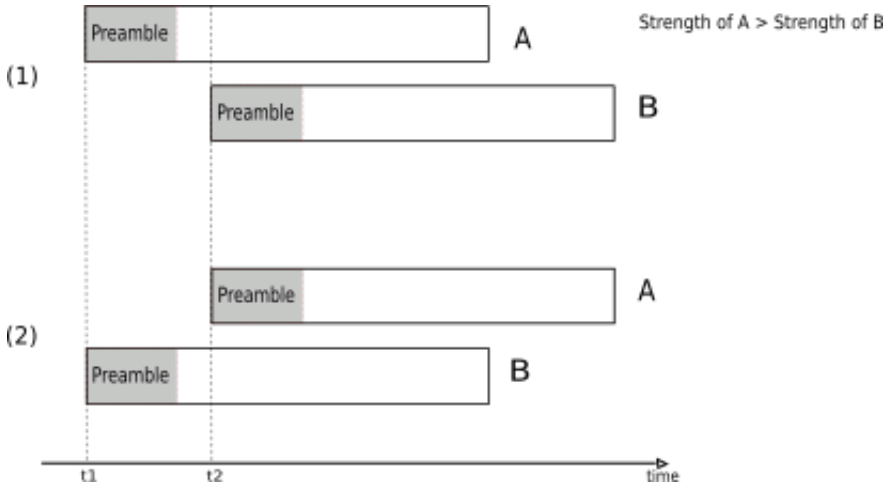

Figure 2.1: Two scenarios of transmissions where capture effect is possible. In scenario 1 packet A arrives first, where packet A is successfully received, and in scenario 2 packet B arrives first, where the reception of packet B is interrupted by packet A.

2.3 Chaos

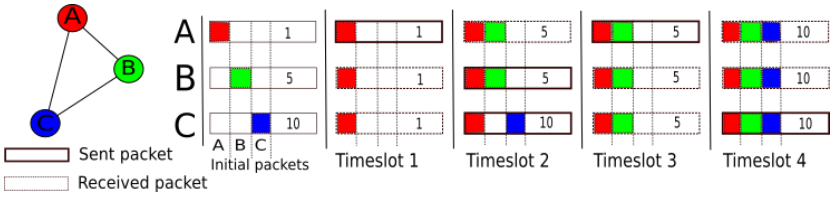

Figure 2.2: Overview of a Chaos round

To provide a better understanding of how Chaos works I provide a basic example of Chaos round, see Figure 2.2. In this example there are three nodes, one of whom is labeled an initiator. The assumption here is that every node in the network has something to send. Before the synchronous transmission takes place every node prepares its own packet consisting of two parts, flags and payload. The initiator, node A, then starts by sending its packet to the other nodes and the two nodes, B and C, will periodically turn on their radios and listen until a packet can be received. Due to node B and C being within transmission range packet reception, for B and C, is successful. The flags of both the new and the old packet is merged so that the flags corresponds to which nodes payload is contained in the packet. The merging effectively takes the logic 'OR' operand on both flags. If the flags in the packet has changed, since the previous timeslot, the node prepares to send the merged packet in the next time slot. Node B and C

applies the user defined merge operator to the newly received payload and their own. If a merge of two packets results in that every node has contributed to the payload contained in the packet it enters a completion phase where it sends the completed packet for 5 consecutive rounds. After the merging process nodes B and C calculate the time skew so that the nodes are synchronized next round. In this example the merge operator is the function max (x,y), where the maximum of x and y is returned. The new payload is then inserted back into the packet and the node sends it again synchronously.

2.4 User defined merge operators

Each node merges its own data with the new data, with the given merge operator, it has received from another node. It is possible to erase or manipulate data that is transmitted in the packets. That is not the main interest when I will be performing attacks against the protocol since the merge operators are given at application-level which is at a higher level than the attacks.

2.5 Constructive interference

Constructive interference [10] occurs when two node are transmitting within a given period and the packets are identical. That allows for successful decoding of the packet. In Chaos constructive interference only occurs when it is in the completion phase, when all nodes have contributed to the packet, because the chance of identical packets are very low until then.

2.6 Glossy

Glossy is the core which Chaos is built upon. An architecture with less than one microsecond time synchronization accuracy. Glossy comes with native time synchronization where every node in a network sends and receives data in timeslots with a predefined slot length. The slot length is defined as the period between the start of two transmissions. The size is a constant across all of the nodes. This allows for synchronized transmissions which opens up the possibility for constructive interference and the capture effect. In the packet header there is a 1-byte field reserved for a counter called 'relay counter'. This counter is set to 1 by the node and then incremented by one for every relay of the packet. To achieve synchronization the 'relay counter' is used to compute the time when the initiator started a flood. When the packet is relayed to other nodes, further away from the initiator, will be synchronized as well.

Chapter 3

Methods of Attacking and Weaknesses in Chaos

WSNs are generally resource constrained and are required to be energy efficient [1]. Therefore it is hard to apply countermeasures to protect the devices from external threats. The networks are often placed in environments where the signal is exposed to hijacking or fake nodes being inserted into the network. Due to it being wireless and objects such as walls and trees have a negative impact on the signal. Chaos has no native protection against attacks but the core transmission mechanism, which is Glossy, proved to be robust against external threats [4]. There are numerous ways of attacking/disrupting Chaos but the ones I am interested in are more specific to Chaos and its techniques. Chaos relies on the capture effect for packet reception and time synchronization for transmissions. An effect of breaking the synchronized transmissions is that the capture effect is affected. The flag header in Chaos packets are exposed to tampering and could result in either premature termination or continuous transmission until end of the Chaos round.

3.1 Modifying Flag Header (MFH)

The packets in Chaos contains a header which keeps track of what nodes has contributed to the data currently in that packet. To achieve this I change the flags in the header of the packets just before the merging of the flags is performed. The resulting flag header is either all flags equals to one or all flags equal to zero.

3.1.1 All Flags One (AFO) attack

By setting the flag in every packet to one the node and nodes which receives data from attacking node will believe that all nodes has contributed to the data in the packet, but such is not the case. The expected behavior will be premature termination for nodes affected and in case of multi-hop the data might not reach nodes further away.

3.1.2 All Flags Zero (AFZ) attack

By setting the flag to zero in every packet I will make the network believe that all nodes has not contributed to the final packet, thus the expected behavior will be that the protocol will continue transmissions until the end of the duration and not actually getting successful transmissions. To simulate the worst case scenario I also set the flag, which determines if the node transmits in the next time slot, to one. That means the node will transmit even though it has no data, according to the flags.

3.2 Drizzle Attack (DA)

To keep nodes busy with a bogus packet performing a DA (high frequency version of Droplet-attack [3]) is a viable option. The Droplet-attack is performed in such a way that the attacking node sends a 802.15.4 PHY frame with a fake payload length making affected nodes wait for data that isn't coming. Drizzle works the same way except continuously sending the frame so that the worst case scenario can be evaluated. By performing a DA I am physically jamming the affected node as opposed to the other methods where I clone a node, in the network, and change the behavior of the protocol. As part of the DA two different jamming waveforms can be generated. Modulated carrier jamming mode and unmodulated carrier jamming mode. Making it a total of 3 different ways of jamming.

3.3 Modify Relay Counter (MRC) attack

A Modify Packet attack [4] was performed on Glossy by Hewage et al. when testing the availability of Glossy. It proved to be very effective. In Glossy this attack affected the constructive interference but in Chaos packets are not identical, therefore it will not work in the same way. Chaos still relies on the relay counter for time synchronization. The MRC attack means I will modify the relay counter of a packet before sending it. By modifying the relay counter to a bogus value it will make the affected nodes become out-of-sync with the initiator. The time synchronization is calculated with the value of the relay counter.

Chapter 4

The Experiments

Evaluation of the attacks are done by performing the attacks in a testbed called Flocklab [7] and then collecting the statistics from the nodes. Figure 4.1 illustrates the placement of the nodes.

4.1 Setup of the experiments

The Flocklab testbed is a real-world environment with 31 wireless nodes. Since the nodes operate on the same frequency as wifi I choose channel 26 and transmission power 31 for the experiments. Using channel 26 excludes interference from wifi networks. Initiating node is set to 16 and attacking node to 28. Node 28 is able to reach as many node as possible for a single attacker. Three test are performed for each attack, three for each mode in DA. Each test is run for 1800 seconds with the attacks starting after 300 seconds has passed. The first 300 seconds the attacking nodes will cooperate with the rest of the nodes in the network.

4.2 Single attacker

This section will cover experiments performed with a single attacker.

4.2.1 Results from AFO and AFZ attacks

The AFO attack had no effect on the network when looking on the packet reception rate (PRR). The effects of the attack are instead seen as a increase in duty cycle evenly across the network, see Figure 4.3. The attack spans almost the entire network with the exception of a couple of nodes. Yielding a high efficiency per attacker. The average amount of transmissions for affected nodes is significantly less than normal, see Figure 4.2. The AFO attack was enough for the nodes to enter the completion phase earlier, leading to premature termination. The amount of nodes that has contributed to the payload of the final packet is unknown. By tampering with the flags the protocol has to compensate that by increasing the duty cycle.

As well as the AFO attack the AFZ attack had no effect on the PRR but here the duty cycle is also affected by this attack, as seen in Figure 4.3. Nodes affected by the attacker had a significant increase in average number of transmissions per chaos round. The AFZ attack made it harder for nodes to reach the completion phase but the affected nodes still managed to get 100% PRR.

4.2.2 Results from DA

All three types of jamming had close to the same result in ppr and only 2 nodes were affected. The affected nodes had their PRR reduced to under 20%. Many nodes avoided the attacker thanks to the capture effect meaning the packet took a detour through nodes outside of the attackers range. The result of all three type of jamming can be seen in Figure 4.4. Which had close to the same effect on the network. The efficiency of this attack was very low.

4.2.3 Results from MRC attack

The MRC attack had very little to no effect on nodes in the network, with a single attacker. The attack covered almost the whole network, leaving just one node att 100% PRR. The rest of the nodes still managed to have above 99% PRR with a standard deviation of max 0.22. Due to some nodes staying synchronized a node could still receive a packet transmitted by another node, thus mitigating the de-synchronization caused by a bogus relay counter.

4.3 Multiple attackers

When performing tests with multiple attackers the nodes 28, 20 and 33 are chosen. That is because the attackers should able to reach as many nodes as possible.

4.3.1 Results from AFO and AFZ attack

Multiple attackers only showed a slight advantage over a single attacker in terms of average number of transmissions which affected more nodes and a lower average across affected nodes. Attacking nodes are seen, in Figure 4.5, having a high number of transmissions even for AFO attack. That is due to the manipluation of the flags. Multiple attackers had a little improvement in network coverage compared to a single attacker. There is no significant increase in duty cycle compared to a single attacker. That means this attack is not as efficient as with a single attacker. This attack managed to span the whole network.

4.3.2 Results from DA

DA with multiple attackers had more effect per attacker than with a single attacker, see Figure 4.7. Mainly thanks to the strategic placement of the attacking nodes. The placement of the attacking nodes 20, 28 and 33 (see Figure 4.1) means that the nodes

that distribute packets further away are kept busy by the DA. Effectively stopping the packets from going further in the network.

4.3.3 Results from MRC attack

The MRC attack with multiple attackers affected the network sligthly more than with a single attacker. The attack still had no significant impact, see Figure 4.8. The PRR for nodes was still above 95% during the MRC attack. The reason why the PRR is still above 95% is due to not all nodes being affected at the same time. Thus compensating for nodes that are de-synchronized.

Figure 4.1: Placement of nodes in Flocklab

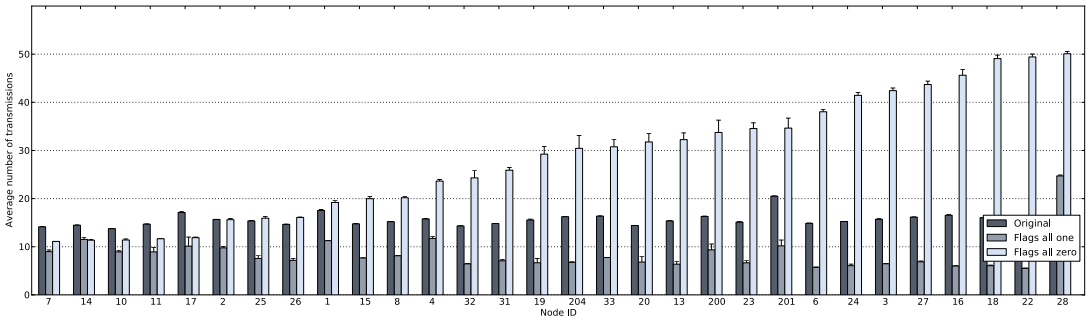

Figure 4.2: Average number of transmissions for nodes during AFO and AFZ attacks. Single attacker.

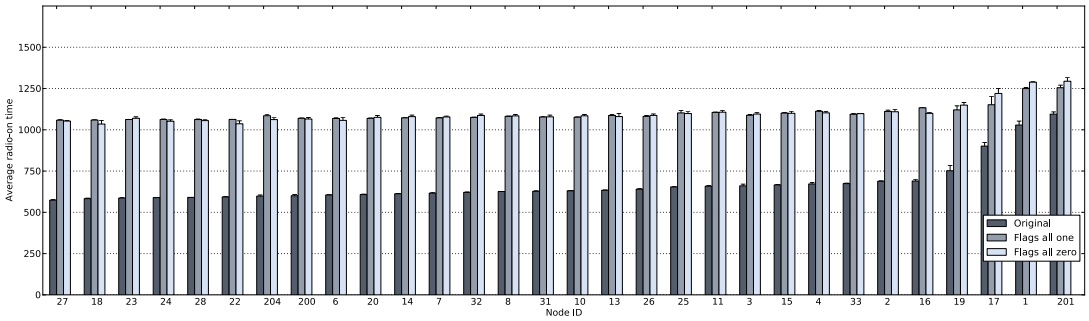

Figure 4.3: Average radio-on time for nodes during AFO and AFZ attacks. Single attacker.

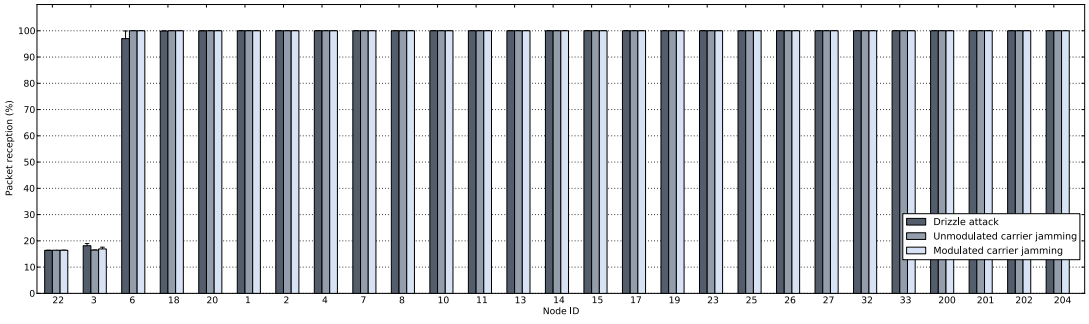

Figure 4.4: Reliability for nodes during DA, unmodulated carrier jamming and modulated carrier jamming. Single attacker.

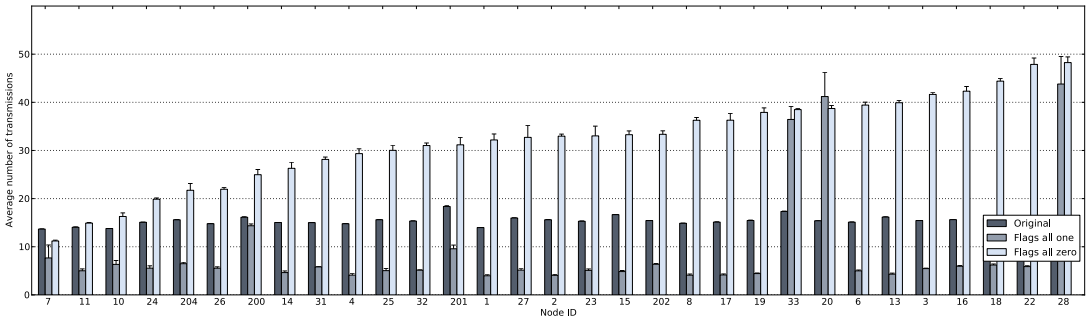

Figure 4.5: Average number of transmissions for nodes during AFO and AFZ attacks. Multiple attackers.

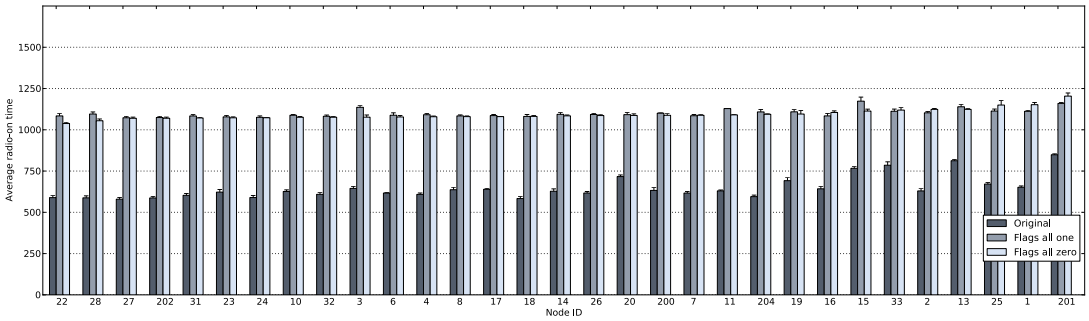

Figure 4.6: Average radio-on time for nodes during AFO and AFZ attacks. Multiple attackers.

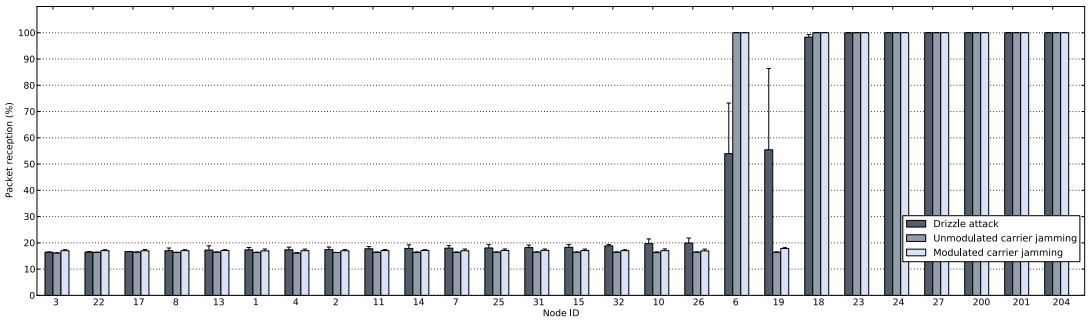

Figure 4.7: Reliability for nodes during DA, unmodulated carrier jamming and modulated carrier jamming. Multiple attackers.

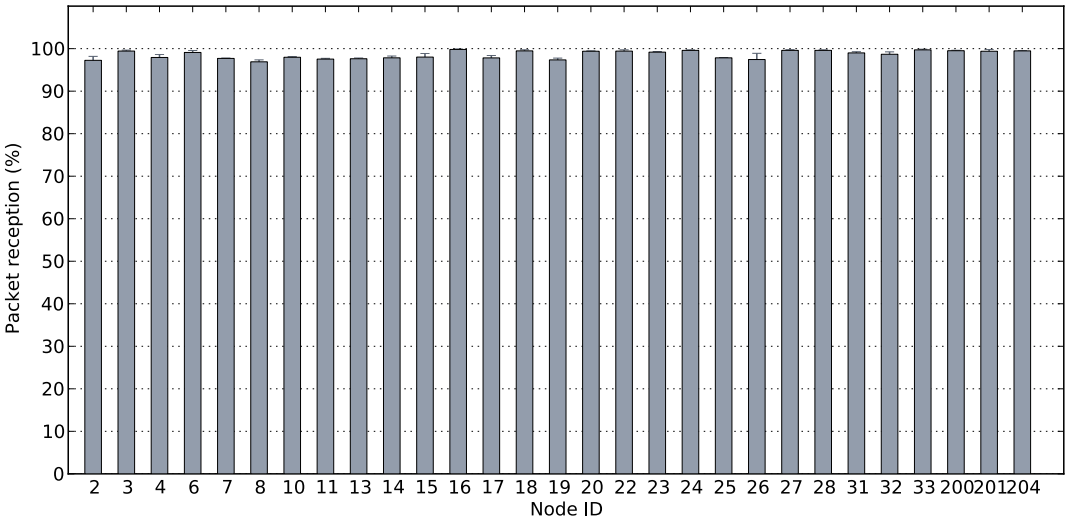

Figure 4.8: Reliability for nodes during the MRC attack. Multiple attackers.

Chapter 5

Discussion

All the attacks except DA had a bit less effect than I anticipated but the MFH attacks had another effect on the network instead. The MFH attacks increased the duty cycle across all nodes, increasing the power consumption for networks already under resource constraints. The AFO attacks decreased the average number of transmissions so I assume, since the transmissions were less than 50% of a network without attackers, that all nodes have not contributed to the payload in the final packet. Making it a possible attack to distort or erase data inside a network. Couple that with the ability create harmful user defined merge operator and the integrity of the packets are easily broken. The AFZ had the unexpected result of 100% PRR even when there are multiple nodes attacking. The network is unaffected because it still receives the packet but with incomplete payload, according to the flags in the header.

The MRC attack had very little effect on the network where almost every node was affected except one for a single attacker. It was a good way to implement the attack, by spreading it via the packets, but the relay counter is not a weak point in Chaos. Showing that the synchronized transmissions is a robust architecture. The DA had good effect on nodes withing range because it constantly kept nodes closest to it busy, with the 802.15.4 PHY frame with a fake payload length, every time they turned on.

The DA with multiple attackers shows that it is possible to isolate the initiator from the rest of the nodes. That means those nodes will just be idling until it receives a packet which might never arrive. Chaos can implement more initiators just in case of DA with multiple attackers but that will lead to the possibility of more concurrent senders in the network. That will lead to lower PRR [6] since constructive interference is not a possibility.

What makes Chaos unique and, in my opinion, interesting is the all-to-all data combined with in-network processing. Thanks to Glossy it makes this process efficient. That is because there is no routing needed in Glossy. If there is no routing in the

network there is no possibility for certain attacks related to routing. Such as selective-forwarding attacks, sinkhole attacks and wormhole attacks [5].

Because Chaos does not rely on routing it cannot perform countermeasures against fake nodes spreading false data. Due to it not being able to calculate the expected route of a packet and then reroute it through other nodes when an attack is detected. An attack can be detected by calculating a time frame where the packet, or packets, should arrive. If the packet does not arrive an alternative route can be calculated and chosen. To develop a secure protocol Wood et al. suggest that countermeasures are thought of at design time of the protocol [9]. Security is very important to be able to deploy WSNs, because not enough security compromises both the integrity and availability of a network. Thus limiting the usability for WSNs.

Chapter 6

Related work

No previous work has been done that has tested the availability of Chaos [6]. As mentioned before, the availability of Glossy has been tested by Hewage et al [4] where a DA also was performed. The results of the DAs on both Glossy and Chaos showed similar result. The affected nodes suffered greatly in PRR. The MRC attack, which is similar to the Modifying Packet Attack performed by Hewage et al, didn't have any significant effect on the network. Mainly due to Chaos not inheriting the weaknesses from Glossy. In WSNs without routing it is hard to prevent or detect physical jamming attacks, due to them being resource constrained. DA can be performed to disrupt the availability of most protocols but with routing it is easier to detect and avoid the jammed area. DA also works on most protocols because of the way it performs the jamming. It aims to collide with packets to disrupt the service. To perform a more energy-efficient jamming attack the Droplet-attack can be performed. DA is the high frequency version of the Droplet-attack.

Attacks proposed in this thesis are aimed to be Denial-of-Service attacks (DoS). When performing DoS attacks there are several ways of doing it. Wood et al. describes different types of DoS attacks in wireless sensor networks [9]. The attacks I performed can be divided into two different types. One is *collision*, which is the DA and MRC-attack. The other one is *misdirection*, which are the AFO and AFZ attacks. *Collision* attacks are when attacks intentionally collide packets to disrupt the service. An attack that injects false data to disrupt the service is called *misdirection*.

Chapter 7

Conclusions

In terms of packet reception Chaos was only affected by the DA. The MFH attack had an effect on the radio duty cycle and the average number of transmissions. Because the transmissions are so low for the AFO attack across almost every node an assumption is that every node hasn't contributed to the payload contained in the final packet. The difference between MFH with a single attacker and multiple attackers were barely significant.That indicates that it is best to use a good placement of one node and an attack that reduces the PRR and the attack can be transmitted via packets.

The MRC attack had a minimal impact. Making it less efficient than the DA which had a major impact on nodes within range of the attack. Since an initiator is needed to start the process a good method for attacking would be to isolate the initiator from the rest of the nodes.

Chapter 8

Future work

MFH still had 100% packet reception rate but the actual payload content of the packets at the end of each Chaos period is not verified. That would be an interesting aspect to look at since the flags are manipulated and the average transmissions are lower than a network without an attacker. That could indicate that every node has not merged the payload contained in the packet. I was looking into ways of disrupting the actual flow in the network rather than manipulating packets, in a harmful way, sent by the nodes. The transmissions of nodes that were exposed to the attacks I presented did not focus on a specific distribution technique, such as all-to-all or one-to-all. There is a possibility for a different result with the same kind of attacks but with transmissions only aimed to be all-to-all distribution.

When executing a DA it was important to reach as many nodes as possible with as high SNR as possible. Otherwise the packet could take a detour through other nodes. Performing a DA where the attacking nodes are placed so that they isolate the initiator, effectively stopping the signal, would be interesting to see. Not to see if it works but to see how many disabled nodes per attacker it is possible to achieve in different scenarios.

Bibliography

[1] G. Anastasi, M. Conti, M. D. Francesco, and A. Passarella. Energy conservation in wireless sensor networks: A survey. In *Ad hoc networks 7 (3)*, 2009.

[2] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh. Efficient network flooding and time synchronization with glossy. In *Information Processing in Sensor Networks*, April 2011.

[3] Z. He and T. Voigt. Droplet: A new denial-of-service attack on low power wireless sensor networks. In *Mobile Ad-Hoc and Sensor Systems*, October 2013.

[4] K. Hewage, S. Raza, and T. Voigt. An experimental study of attacks on the availability of glossy. In *Computers and Electrical Engineering*, October 2014.

[5] C. Karlof and D. Wagner. Secure routing in wireless sensor networks: Attacks and countermeasures. In *Ad hoc networks 1 (2)*, 2003.

[6] O. Landsiedel, F. Ferrari, and M. Zimmerling. Chaos: Versatile and efficient all-to-all data sharing and in-network processing at scale. In *SenSys '13: Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, November 2013.

[7] R. Lim, F. Ferrari, M. Zimmerling, and C. Walser. Flocklab: A testbed for distributed, synchronized tracing and profiling of wireless embedded systems. In *Information Processing in Sensor Networks*, April 2013.

[8] J. Lu and K. Whitehouse. Flash flooding: Exploiting the capture effect for rapid flooding in wireless sensor networks. In *Infocom 2009*, April 2009.

[9] A. D. Wood and J. A. Stankovic. Denial of service in sensor networks. In *Computer 35 (10)*, 2002.

[10] D. Yuan and M. Hollick. Let's talk together: Understanding concurrent transmissions in wireless sensor networks. In *38th Annual IEEE Conference on Local Computer Networks*, October 2013.