American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)
ISSN (Print) 2313-4410, ISSN (Online) 2313-4402

http://asrjetsjournal.org/

Modeling of Wireless Sensor Networks Jamming Attack Strategies

Chukwuemeka Duru[a]*, Azubuike Aniedu[b], Onyeyili Tochukwu Innocent[c], Alagbu Ekene E.O[d]

[a]*Imo State University PMB 2000 Owerri, Imo State Nigeria*
[b,c,d]*Nnamdi Azikiwe University PMB 5025 Awka, Anambra State Nigeria*
[a]*Email: chukwuemekacduru@gmail.com*
[b]*Email: anoca505@gmail.com*
[c]*Email: ti.onyeyili@unizik.edu.ng*
[d]*Email: ee.alagbu@unizi.edu.ng*

Abstract

Jamming attacks like constant jamming, deceptive jamming, random jamming and reactive jamming disrupt the normal operation of nodes in wireless sensor networks thereby drastically affecting the network throughput, delay and energy consumption by sending random signals in the network without following the underlying Media Access Control rules. These signals collide with the legitimate signals causing undue traffic in the network leading to Denial of Service Attack. This work provides a model for the different jamming attacks experienced in wireless sensor networks using Unified Modeling Language. Modeling the different jamming attacks would help in better description of their behavior and strategy. Established models can be employed in the design of jamming detection and mitigation framework for DoS attacks mitigation in wireless sensor networks that involve sensors which are known to have limited and constrained resources.

Keywords: Jamming attack; Media Access Control; Denial of Service Attack; Sensors.

1. Introduction

Wireless Sensor Networks (WSNs) are prominent and versatile area of research due to their wide range of application in critical areas of the society like healthcare, infrastructures and military surveillance systems.

---

* Corresponding author.

Nodes in this grade of network can be of equal or varied capacity depending on their design architecture. Generally, there are three major forms of WSNs thus; cluster based, flat-based and hierarchical wireless sensor networks with the major difference being the arrangement of node and how collected data are transferred from the sensor node to the base station[1] During deployment, WSN find application in the following distinctive areas thus; underground, multi-media, Mobil and terrestrial WSNs **[**2**]**. These sensors are usually autonomous in operation and are located at remote areas where they are employed in fields to collect some physical data maybe analyze them based on some inbuilt algorithms and forward same to a central unit for further use [3]. Attacks based on generation of random signals without following the underlying MAC rules have shown to interfere with normal operation of most wireless networks. These interferences have shown to be more common in wireless networks than their wired counterparts [4]. The nodes in WSNs are known to have low communication bandwidth, short communication range, limited energy storage capacity and generally poor hardware and system resources like, memory, storage and computational power [5]. This implies that most security attack mitigation methods available for tradition networks implemented on the conventional computer systems would either be too large to be used on these grades of system or would not function as intended when applied to these nodes due to their limitations.

1.1 Security Issues in Wireless Sensor Networks (WSNs)

The main purpose of wireless sensor networks is to transmit data in a secure and fast manner from a node to a sink node that in turn sends same to a base station through multi-hop transmission [6]. The sink node is responsible for receiving data from the various sensors associated with it in a particular network and forwards same to the base station. Due to their peculiar importance in wireless sensor networks, they are usually located at strategic point in the network to reduce energy requirement of the overall network.

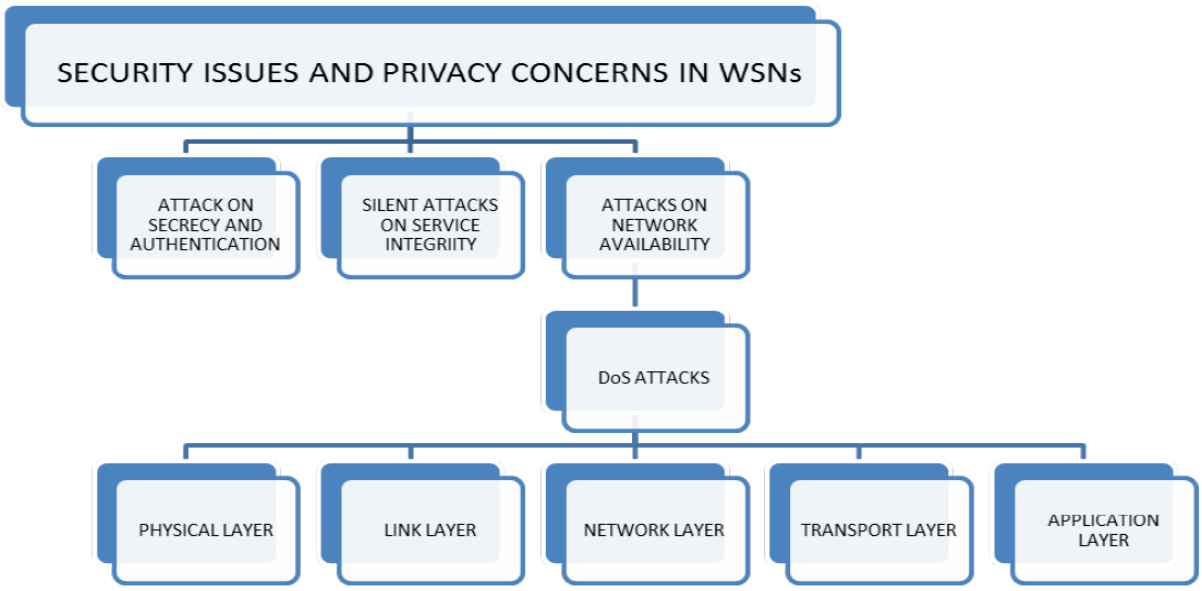

Figure 1: Hierarchical Diagram of Security Issues in Wireless Sensor Network

Most sensor networks being data-centric requires that the privacy, integrity, confidentiality of data must be ensured owing to the fact that most of the data carried by these sensors if compromised could lead to industrial/national espionage or even threat to life [7]. Security of communication between nodes and between nodes and sink nodes must be guaranteed in a WSN for improved efficiency and trust. A number of attacks exist in a wireless sensor network which can lead to breach in communication, sensor node routing or even information leakage. These attacks are shown in figure 1

From figure 1, attacks that can be carried out on a wireless sensor network can be grouped into three areas [8]:

i. Secrecy and authentication attacks
ii. Service integrity silence attacks
iii. Network availability attacks: DDoS which prevents legitimate users access to authorized network resources falls under this category.

Different attacks on the different layers of the WSN shown in figure 1 are described below for better understanding of the nature of attacks that WSN are susceptible to.

a. DoS attack on the physical layer: In the WSNs, this layer is responsible for carrier frequency selection and generation, modulation and demodulation, encryption and decryption, transmission and reception of data [9].

This layer is attacked mainly by the following schemes

i. Jamming: This type of attack compromises the communication channels between two nodes thereby preventing them from exchanging information.
ii. Node tampering: This is the process of physically tampering with the nodes of the WSN with the sole aim of extracting sensitive information.

b. DoS attack on the link layer: Multiplexing of various data streams, detection of data frames, media access control (MAC) and error control are all the functions of the link layer. The reliability of point – to - point or point – to – multipoint communication is also guaranteed by this layer [10]. The following DoS attack occur in this layer;

i. Collision: This may occur when two nodes in a network sense the channel to be idle at the same time and starts transmission of data packets on the same frequency channel. Collision of packets may results in packet modification or even loss which may lead to identification mismatch at the receiving node. Once this happens, a re-transmission request may is initiated by the receiving node hence, making the channel busy and overworked maliciously (Ghildiyal and his colleagues 2014).
ii. Unfairness: This is a collision based attack usually refered to as exhaustion based attacks (Ghildiyal and his colleagues 2014).
iii. Battery Exhaustion: Here, high traffic activities is maliciously perpetrated on the channel thereby making the channels accessibilty to authorized nodes limited. This is usually caused by a lot of re-

transmission requests and request to send signals.

c. DoS attack on the network layer: In WSN, network layer handles all routing requirements. Some DoS attacks that occur in this area is described below;

i. Spoofing, traffic misdirection and replaying
ii. Hello flood attack: Here, the channel is congested with an unusually high number of useless messages thereby causing high traffic in the channel. This attack employs a single malicious node in generating the useless message which is in turn replayed by the attacker causing high traffic in the channel.
iii. Homing: It begins with a search in the network for cluster heads and key managers. Compromising key mangers and cluster heads if successful can shut down an entire network.
iv. Selective forwarding: A compromised node here only sends information to a few selected nodes instead of all the nodes in the network. The choice of nodes to forward data to is made by the attacker in other to achieve its malicious objectives.
v. Sybil: Sybil attacks involved the attacker replicating a single node and presenting it with multiple identities to other nodes in the network.
vi. Wormhole: This DoS attack relocates bits of data from the designated position in the network to another. Bits relocation is achieved through data bits tunneling over a low latency link.
vii. Acknowledgement flooding: Routing algorithm requires acknowledgments. This attack spoofs this acknowledgment with the aim of providing false information to the designated node(s).

Figure 2: Types of Denial of Attack in Wireless Sensor Network

d. DoS attack on Transport Layer: Transport layer handles data transmission reliability and smoothening of congestions that may result from high traffic in routers. A few DoS attacks in this layer are;

i. Flooding: Here, the attacker consciously floods the network with unnecessary messages causing high traffic in the channel.

ii. De-synchronization: This attack employs fake messages at one or both endpoints requesting or initiating re-transmission for correction of an error that never occurred in the network. During this process, one or both end-point looses energy as a result of carrying out the spoofed instruction.

DoS attack on the Application layer: This layer to an extent handles traffic management in WSN. But the main purpose of this layer is to provide different software applications that would carry out the translation of data into a comprehensible form and also help in information aggregation by sending queries [10]. Here, a compromised sensor node is forced to initiate a huge traffic in the route towards the base station there by making the path inaccessible by legitimate nodes **[**11**]**.

2. Jamming attacks in wireless sensor networks

The sensors that constitute the nodes of WSNs are often deployed in inaccessible, most times dangerous and remote areas and are known to have limited system resources. These characteristics alongside the high vulnerability of wireless networks have made WSNs quite susceptible to DoS attacks especially jamming [12]. Recently, the Mirai malware was reported to have been used to cause large scare DoS attack infecting over 100,000 nodes on the same day making the underlying network inaccessible to these devices. This attack was considered to be the largest botnet attack ever recorded with a magnitude in the range of 1.2Tbps [13]. In the situation of jamming activity, an adversary generates random radio frequency at the same strength with the legitimate nodes in the network with the intentions of disrupting transfer of data between these nodes. Although jamming can occur legitimately by some natural factors like noise, collision and interference [14], this notwithstanding, jamming attack can also be intentional in situations whereby the attacker limits or halts transmission of signals in the network by directing malicious signals towards the channel thereby making it inaccessible to legitimate nodes. Repeated sensing of the channel to determine idle time causes depletion of resources like battery like and bandwidth in the network. The following sections would show a model of the strategies employed by these jammers to achieve their sole objective in the network which are, increased energy consumption, low throughput and high delay in the overall network.

3. Modeling of Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA)

Since sensor nodes are wireless and have an adhoc arrangement, there should be provision for an access method to be implemented by all participating device with regard to the use of the channel, how long it should be occupied by a device, what to do if the channel is busy and provision for error. CSMA/CA is the default access method for wireless networks due to its ability to avoid collision[15]. A UML Activity model diagram is employed for this purpose.

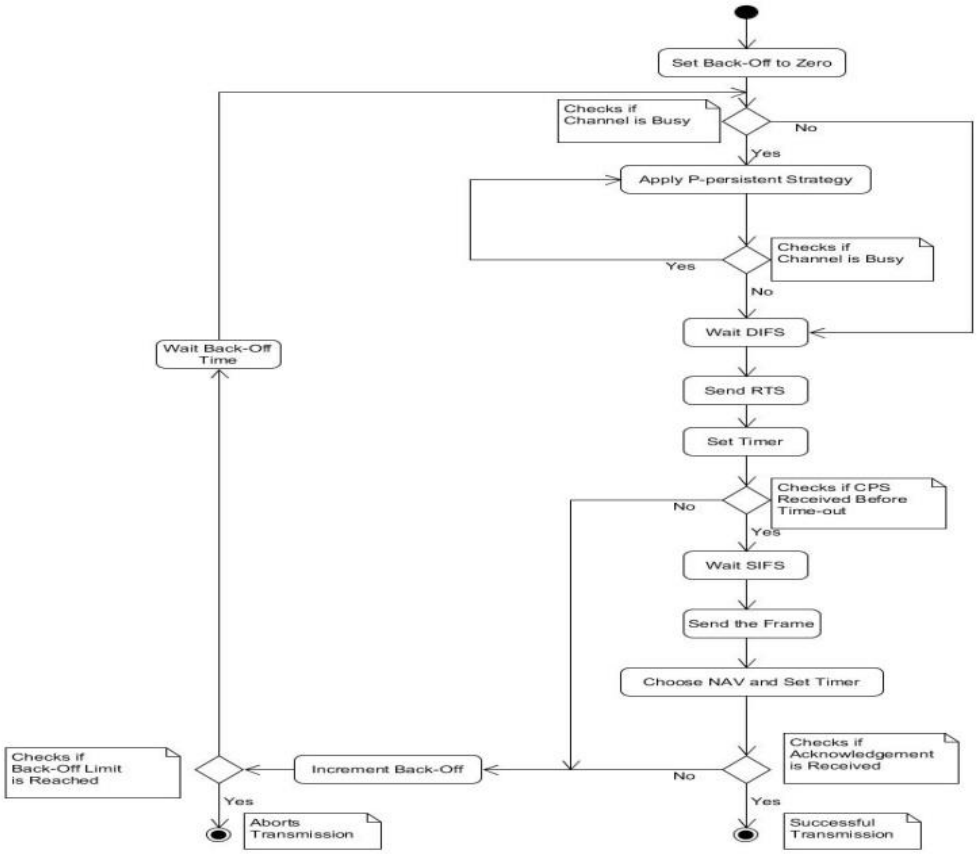

Figure 3: Activity Diagram of the CSMA employed

The diagram of figure 3 is explained in steps thus;

i. Prior to sending, the sending station senses the channel by evaluating the energy level at the carrier frequency. If the channel is busy, then the following steps are taken by the sending node;

a. A persistent method in this case, p-persistent method is applied with back-off until the channel is idle.
b. Once the sender finds the channel to be idle, it waits for a period of time known as distributed interframe space (DIFS); a control frame known as Request to Send (RTS) is subsequently sent after this wait period.

ii. The sending station does not wait indefinitely to receive RTS from the destination station. Its sets a maximum amount of time it would wait for the Clear to Send (CTS) from the receiving station once this time elapses, it increments its back-off time and goes back to sensing the channel again and applying the p-persistent method.

iii. The amount of time the sending station would occupy the channel is also contained in the RTS signal sent across the channel which is seen and read by every other station using the same channel.

iv. All other station would employ this time in evaluating their Network Allocation Vector (NAV). During this time, they won't sense the channel. This would go a long way in conserving energy for the attached nodes (stations).

v. Once the RTS is received by the destination station, it waits a short interface space (SIFS) before

sending its own control signal CTS to the sending station. Sending this frame signifies that the destination or receiving station is set for data transmission.

vi. The sending station would then send data after waiting for a period of time equivalent to SIFS.

vii. The receiving station sends an acknowledgement signal to show that the frame have been received after it must have waited an amount of time equivalent to SIFS.

viii. Acknowledgement is the only means the sending station confirms that data sent successfully got to the intended recipient.

A UML Sequence diagram is employed to model how nodes on the channel employ NAV when applying CSMA/CA algorithm. Sequence diagrams are best used to describe system calls or request and show in great details the interaction between processes in a system. In the diagram of figure 4 for instance, the tiny rectangles on the lines represents a process and the length of these triangle is proportional to the duration of that particular process. The model is shown in figure 4;

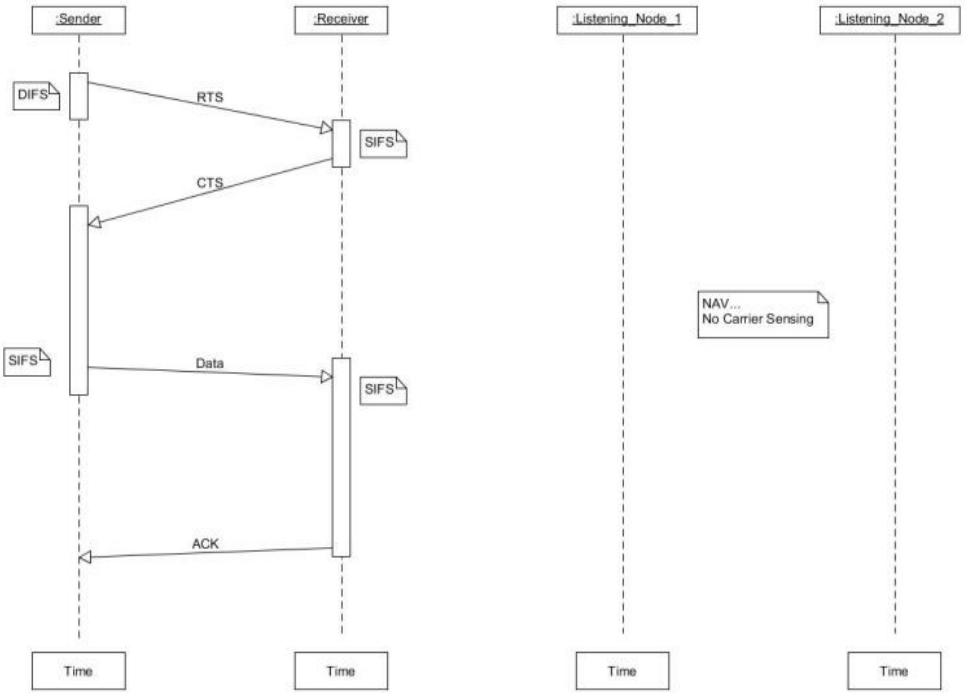

Figure 4: CSMA/CA and NAV

The diagram above shows the interaction of four nodes (used only for reference here) employing CSMA/CA as access method. The sending node above first sends its control signal (RTS) to the receiving node. This control signal is also broadcasted to all other nodes from which they compute the NAV. Other nodes would not sense the channel until the NAV is exhausted. The receiving node will in turn send its own control signal (CTS) implying that it is ready to receive data from the sending node. To ensure that data sent was successfully transmitted, the receiving node sends back an acknowledgment signal to the sender denoting end of transmission (which is known as handshaking).

4. Jamming Attack Modeling

A behavioral model of the different jamming attacks is presented in this section. The developed model is based on Unified Modeling Language (UML) sequence and activity diagrams. These two features of the UML modeling language were chosen because of their ease and simplicity in describing structural and functional processes of a software or digital process. The four forms of jamming attacks to be mitigated by the proposed framework are modeled thus [16];

4.1 Constant Jamming Modeling

The constant jammers strategy is to emit radio signals with intervals between them thereby making the channel busy at all time hence, denying legitimate users access to the channel as they would always presume it busy at every trial. One major drawback of this channel is power exhaustion due to the fact that the jamming node has to be on always to be able to emit constant radio signals to keep the channel busy. It should be noted that a Jamming node do not apply the CSMA/CA algorithm as described above. This implies that it goes against all MAC rules.

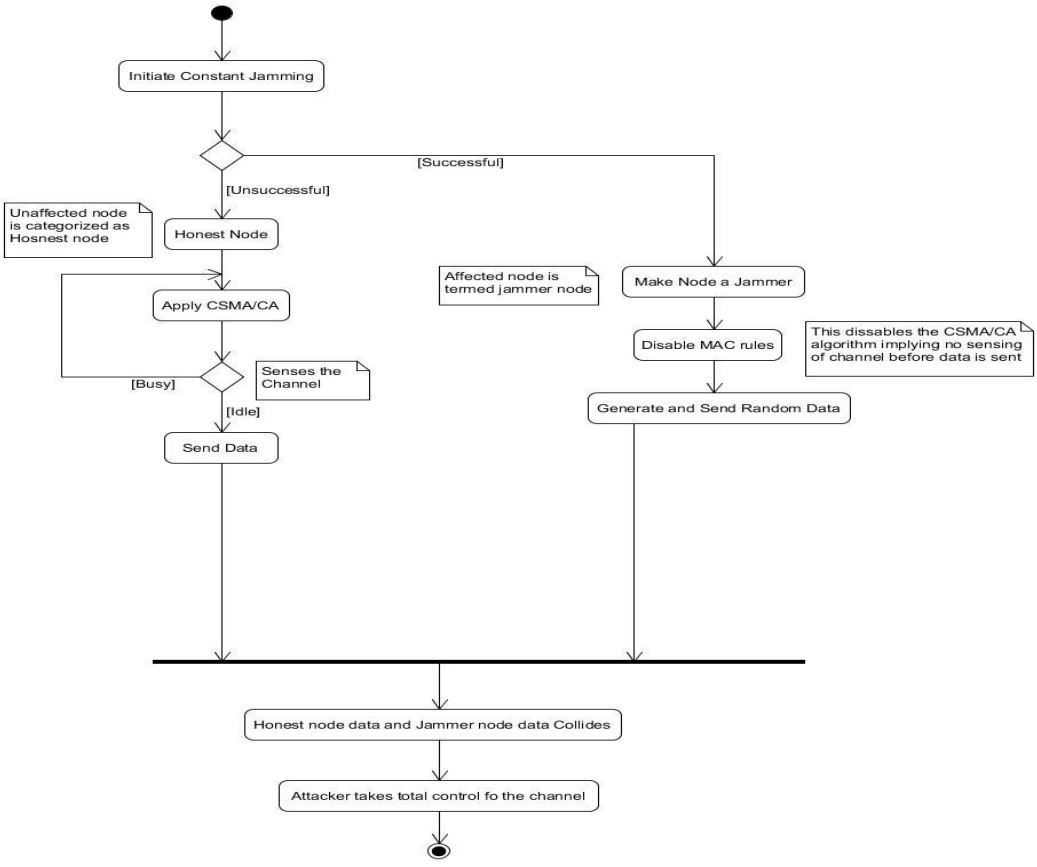

Figure 5: Activity modeling of constant jammer

A. Activity Diagram of Constant Jamming Scenario: An Activity diagram is employed to model the behavior of the system. It shows the control flow from start point to finish point showing different decision

paths that exist while the jamming activity goes on. From the diagram above, nodes that are not under attack whose MAC rules like CSMA/CA have not been disabled are grouped as honest nodes. The attacker tries to compromise a node by making it function as a constant jammer if successful, MAC rules are disabled for that particular node and random data are generated and sent without prior sensing of the channel to find out if it's busy or idle. Other nodes applies the default multiple access method in this CSMA/CA and sends their data into the same channel. Since the jammer node constantly sends signal, at some point the data from the honest node collides with that from the jammer node as shown in figure 5.

B. Sequence Modeling of Constant Jamming Attack: To capture the interaction of objects involved in the jamming process, a sequence diagram is employed to model the constant jamming attack.

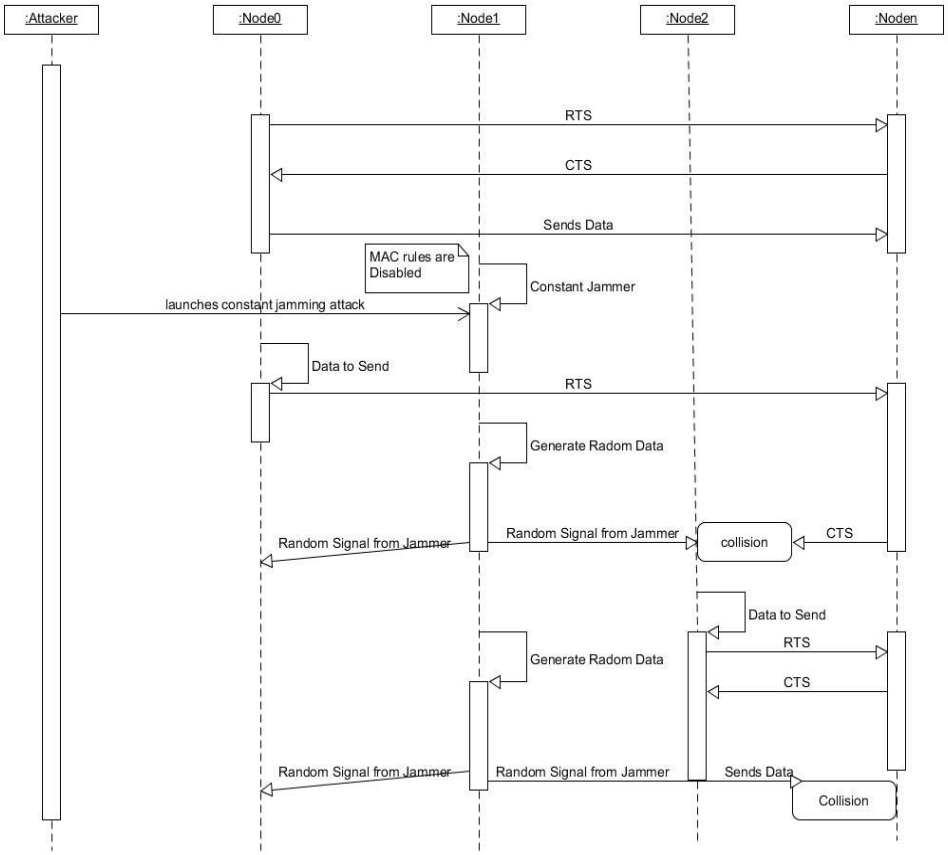

Figure 6: Sequential modeling of constant jamming attack

From the above model, Node0: after applying the default MAC rules and CSMA/CA algorithm defined in figure 3, sends RTS to Noden: Which in turn sends a CTS to it, signifying its readiness to receive data. Once this control signal is received, Node0 sends its data through the channel to Noden. At this point the attacker hasn't succeeded hence all nodes are still honest nodes. Once the attacker captures node1 as shown in figure 6, it makes it a jammer node and subsequently disables its MAC rules and access method (CSMA/CA). Node0 at a later point tries to send data to Noden by sending its control signal RTS which Noden responds by sending CTS that collides with the random data generated by node1 as a jammer node. This same process occurs for Node2 whom after sending RTS to Noden and obtaining CTS has its data collide with the jammer nodes random data.

This process continues and eventually collapses the entire network.

4.2 Deceptive Jamming Modeling

The major difference between the constant jammer and deceptive jammer is that the later sends randomly generated data without any interval between them once the node has been compromised. This strategy makes other nodes believe that there is a legitimate communication hence putting all other nodes in a receive state.

A. Activity Diagram of Deceptive Jamming Scenario: A model of the behavior of the deceptive jamming is shown in figure 7 using UML activity diagram. It shows the flow control from start of the jamming attack to when the attacker takes over the whole network.

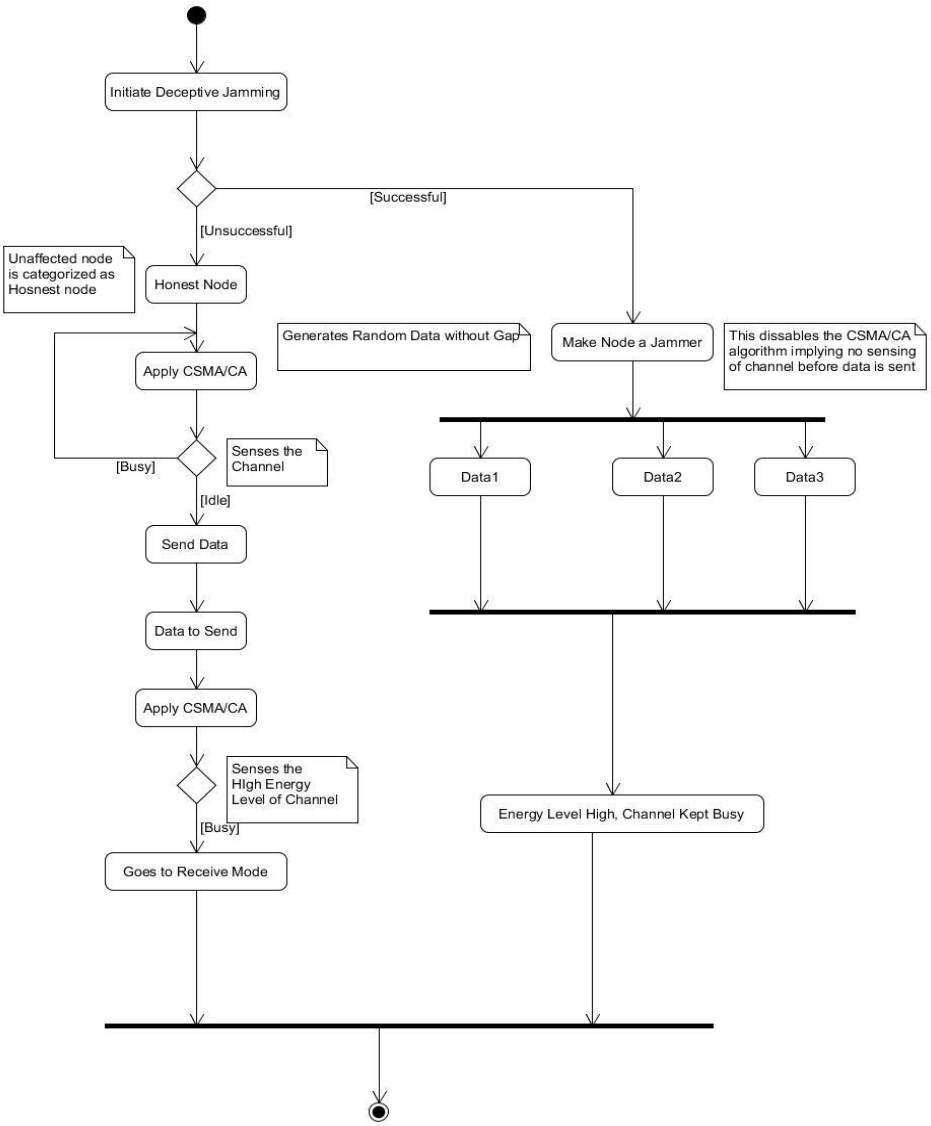

Figure 7: Activity Modeling of Deceptive Jamming Attack

From figure 7, the nodes that are not affected by jamming attack are classified as 'Honest nodes'. The Deceptive

jamming is first initiated and if successful, it makes the comprised node a jammer. If unsuccessful the normal sensing methods like CSMA/CA is applied following the existing MAC rules. Once the node has been made a jammer, it then would employ the deceptive jamming strategy described above by generating random data with no interval between them. This makes the other nodes trying to access the channel to assume that there is a legitimate transmission ongoing in the channel thereby putting them in receive mode throughout the jamming duration.

B. Sequence Modeling of Deceptive Jamming Attack: To capture the interaction of objects involved in the jamming process, a sequence diagram is employed to model the Deceptive jamming attack thus;

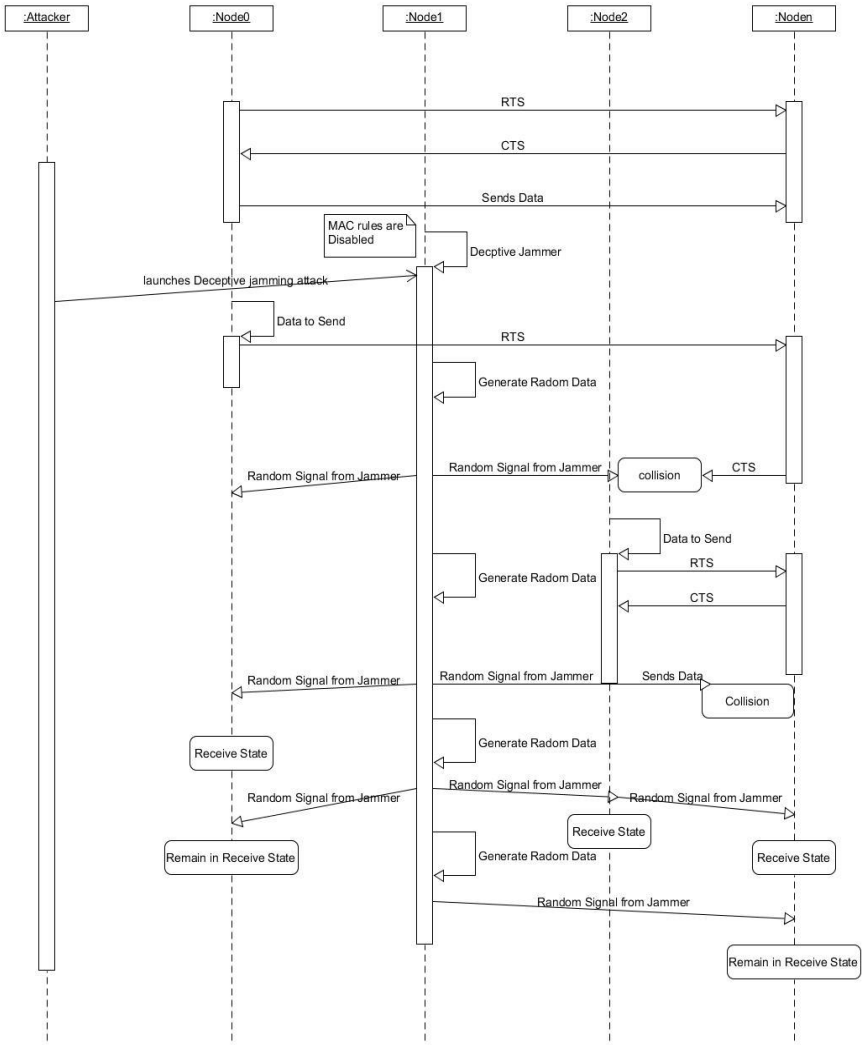

Figure 8: Sequence Modeling of Deceptive Jamming Attack

Figure 8 shows the sequence model of the Deceptive jamming attack, showing the duration of the attack and when Node1 was captured. The capturing of Node1 takes place just after successful data transfer between Node0 and Noden. Once Node1 is successful made to act as a deceptive jammer, it continuously sends random data around in the channel with no gap between them as depicted in the system call line in the model. This

random data collides with the clear to send (CTS) control signal coming from Noden to Node0 after it successfully receives the RTS from Node0. This collision is sensed by Node0 at a later point which keeps this Node in receive state. Also, the same applies to Node2 which after sending RTS to Noden, receiving CTS from Noden and resuming data transfer has its data colliding with the random data from the jammer node. This collision is sensed by Node2 at a later point making it to assume that there is legitimate activity going on in the channel thereby keeping it in receive state.

4.3 Random Jamming Modeling

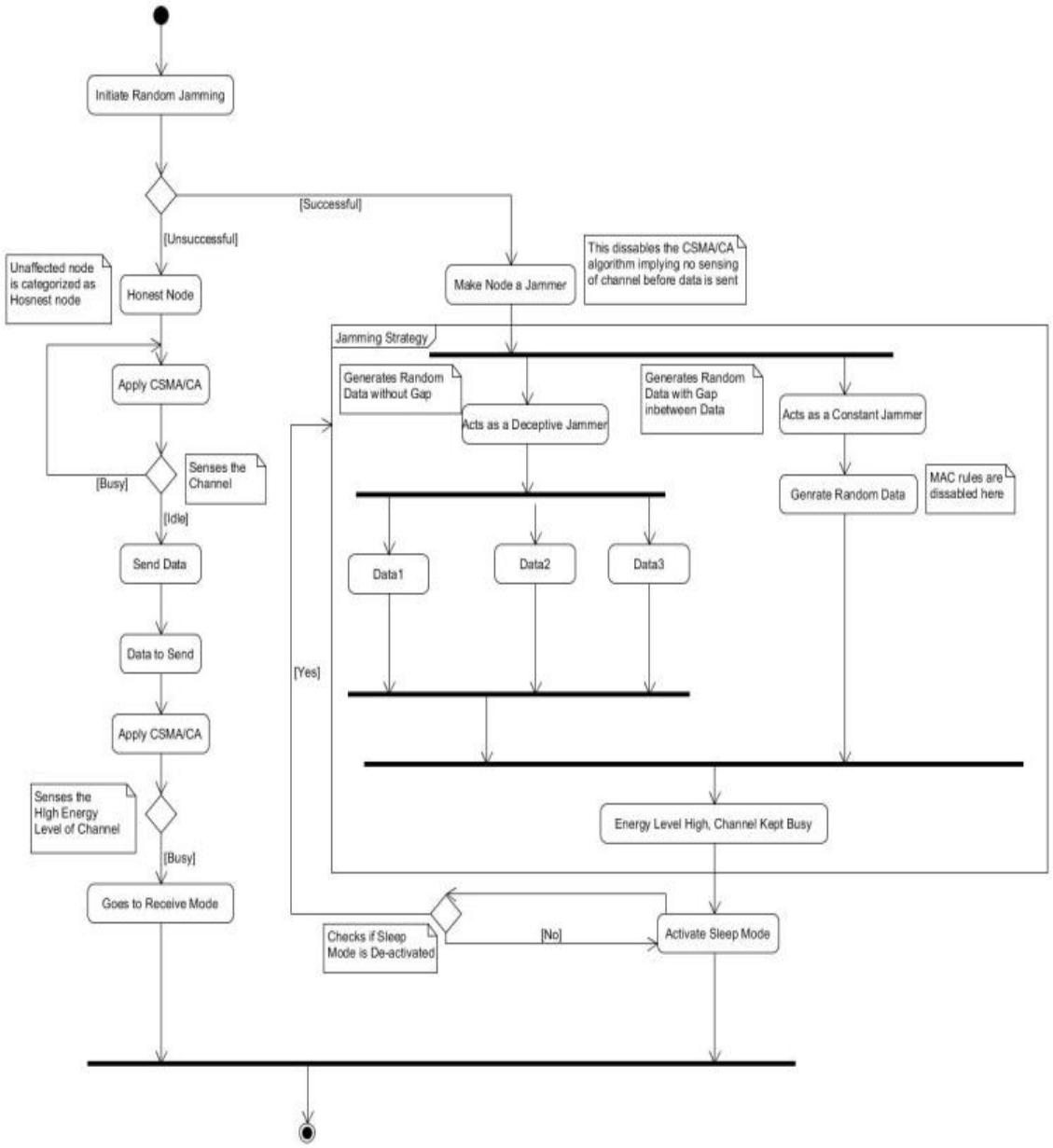

Figure 9: Activity model of Random Jamming strategy

Random jamming is a bit more tactical than the already described jamming strategies. It employs ways of

conserving its energy. After attack for a while it goes into sleep mode, it could then behave either as a constant or deceptive jammer. This jamming strategy is modeled below.

A. Activity Diagram of Random Jamming Scenario: A model of the behavior of the random jamming is shown in figure 9 using UML activity diagram. It shows the flow control from start of the jamming attack to when the jamming node goes to sleep and resumes either as a constant or deceptive jammer.

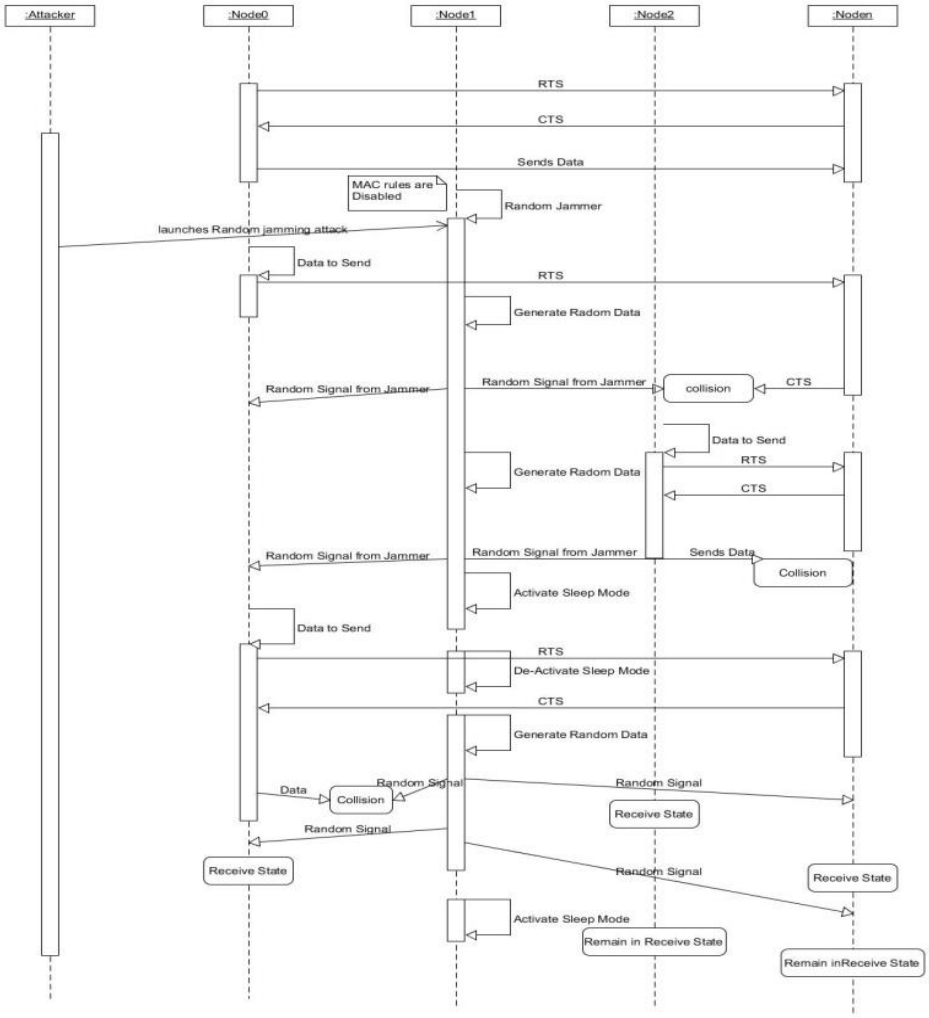

Figure 10: Sequence Modeling of Random Jamming Technique

From figure 9, the nodes that the attacker is not able to compromise are classified as 'Honest Nodes'. Once the attacker is successful, it converts one of the nodes in the network as a jammer node, making it behave like a random jammer. From figure 9, we can see that if the attack is unsuccessful, the 'honest nodes' applies the normal sensing methods and MAC rules guiding channel access. But if the attack is successful, it makes the node a random jammer and can employ either of the jamming strategies of the constant or deceptive jammer described above. After jamming for a while, it goes to sleep and resumes at a predetermined time to carry out the jamming over again. During Jamming, the 'honest nodes' tries applying the guiding sensing methods and MAC rules only to discover that the channel is busy (high energy level) this causes it to back-off and remain in

sleep mode. It should be noted that random jammer last longer since it allows the jammer to sleep for a while after jamming for a specified or random period (according to the design of the attack) thereby conserving its energy.

B. Sequence Modeling of Random Jamming Attack: To capture the interaction of objects involved in the jamming process, a sequence diagram is employed to model the Random jamming attack thus;

From figure 10, Node1 is captured and converted to a Random jammer after successful data transmission from Node0 to Noden. Once the node is captured, it is made to act in this first instance as a deceptive jammer, generating random data without interval between them. This is depicted by the long system call line in the model. This jamming strategy goes on for a while and affects the CTS signal coming from Noden to Node0. The jamming instance at this time also affects the data going from Node2 to Noden after Node2 must have received CTS signal from Noden. The data sent by Node2 to Noden collides with that random data generated by the jamming node. After this first instance, the jamming node goes to sleep to resume at a later time. The model shows that on resumption, the jamming node employs constant jamming strategy this is clearly shown by the short system call line. During this instance, the jamming node (Node1) sends Random data through the channel which affects the data being sent from Node0 to Noden. This occurrence put the nodes associated with this network in receive state, making the attacker to take full control of the channel. It can be seen from the model that after this jamming instance, the jamming node goes to sleep mode to resume at a later time either as a constant jammer of random jammer according to the design of the attack.

4.4 Reactive Jamming Modeling

Reactive jamming is the most complex of all the jamming strategies and it's the only active jamming strategy (the other three are classified under passive jamming strategy). The reactive jammer remains in a quiet or idle state until it senses an activity in the channel by checking the energy level in the channel. Once it notices activities in the channel, it begins transmission of random data causing occurrence of jamming. By behaving this way, reactive jammers last longer since the idle or sleep mode enables it to conserve energy.

A. Activity Diagram of Reactive Jamming Scenario: A model of the behavior of the reactive jammer is shown in figure 11 using UML activity diagram. It shows the flow control from when the jamming node senses activity in the channel, resumes jamming to when it goes to sleep after a successful jamming operation to resume when next it senses activities in the channel.

From figure 11, once the attacker succeeds in converting a node in the network as a reactive jammer, unlike the other jamming strategies, it does not commence random signal generation immediately. The reactive jammer first senses the channel to detect if there is an activity in the channel. If there no activity, it goes back to sleep but if there is an activity it generates and sends random signal across the channel. This random data could then collide with the RTS signal, CTS signal or even data being sent by an honest node. Several of these occurrences may bring the whole network to a halt.

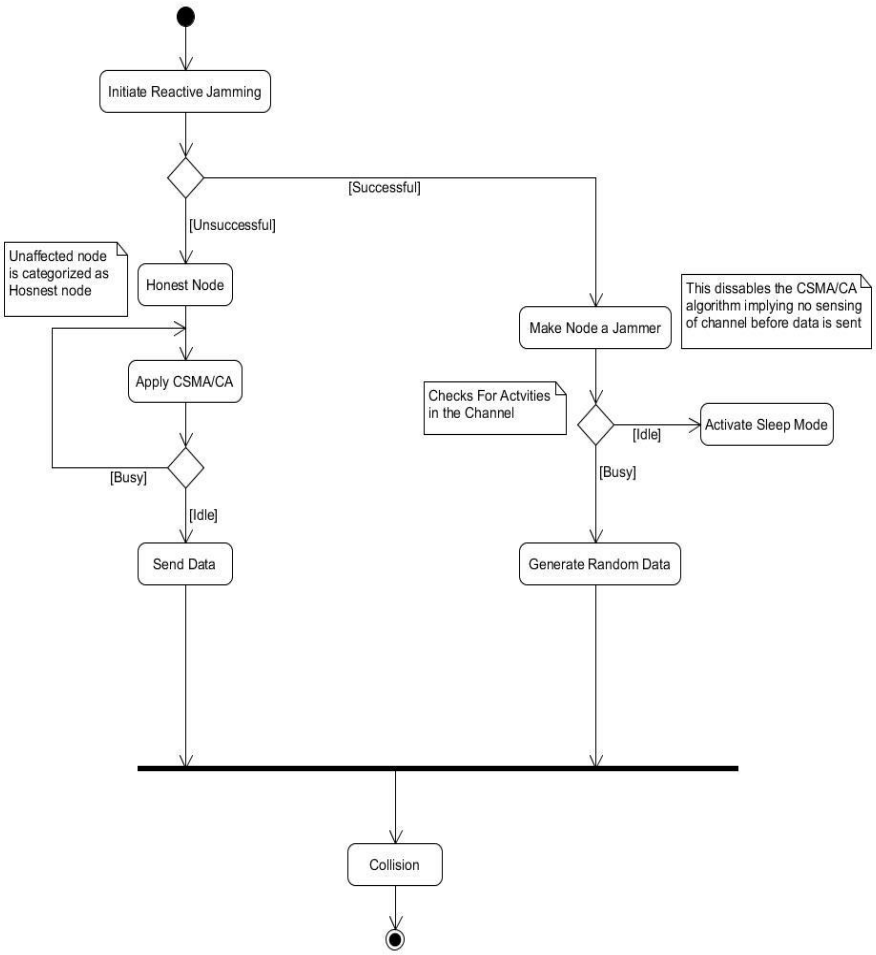

Figure 11: Activity Modeling of Reactive Jamming Attack

B. Sequence Modeling of Reactive Jamming Attack: To capture the interaction of objects involved in the jamming process, a sequence diagram is employed to model the reactive jamming attack thus;

From figure 12, the attacker only succeeds in converting Node1 into a jamming node after there has been a successful transmission of data between Node0 and Noden. Once Node1 is made a jammer, it does not commence jamming operation immediately. It first senses the channel for activities and in this case observes the RTS sent by Node0 to Noden at this instance, it commences the jamming operation by generating and sending random data. This random data collides with the CTS control signal sent by Noden to Node1. This collision sends Node0 to receive mode, since there is no more activity in the channel, Node1 goes back to sleep mode. Another instance of jamming is when Node2 sends RTS to Noden and subsequently receives CTS from Noden again; Node1 senses activities in the channel and resumes its jamming operation by generating and sending jamming signals that eventually collides with the data being sent by Node2 to Noden. This collision also put Node2 to receive state halting the network.

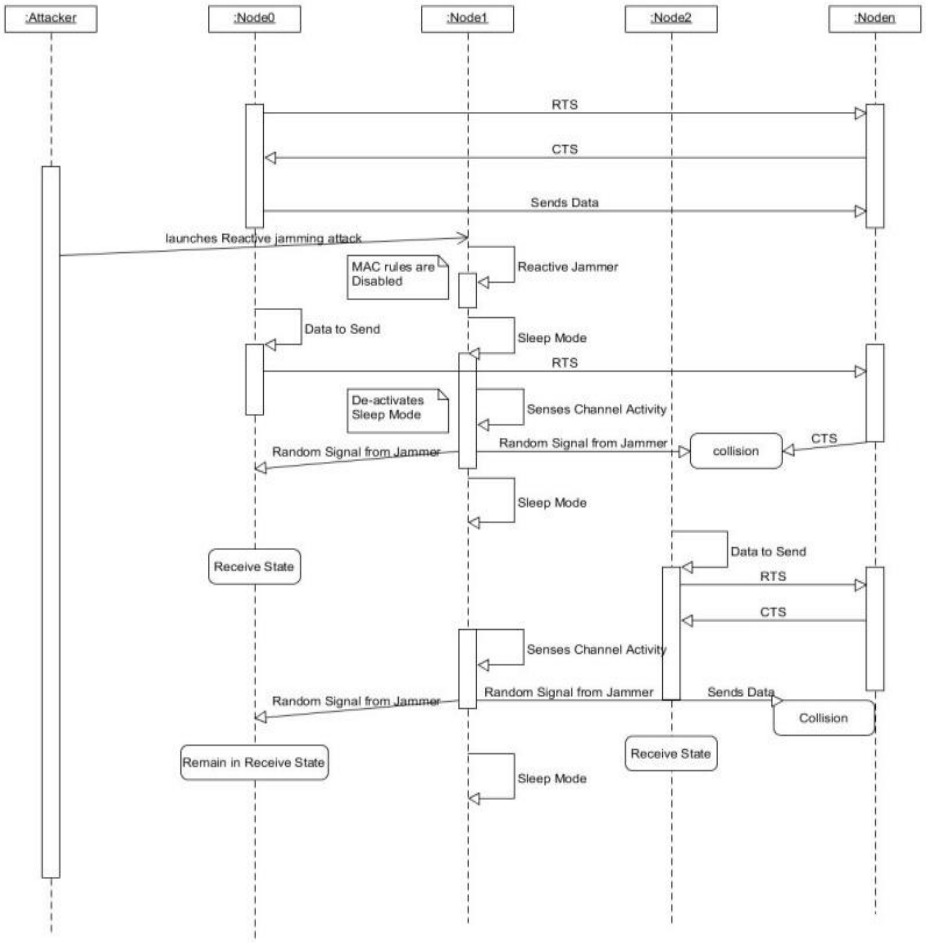

Figure 12: Sequence Modeling of Reactive Jamming Attack

5. Conclusion

Jamming attack is a prevalent security issue in wireless networks especially those involving sensor nodes that are known to have low system resources. It is even harder to manage and track since most of these sensors are located in areas that may be inaccessible to human or even hazardous like nuclear reactors. Based on this, algorithms to allow these nodes in WSNs self heal themselves and mitigate these attack when they occur is a major requirement prior to their deployment in fields. In this work, a model of the strategies employed by the four major jamming attack instance like constant, random, deceptive and reactive jammer was provided. The model also shows that the reactive jammer is the most intelligent kind of jammer due it its active nature and may be the most difficult to detect as the energy level in the network is not reasonably affected since the jammer only resumes activities when there is data movement in the channel. In terms of battery conservation, the reactive jammer also showed good features implying that its jamming activities would be sustained more than the other three. The deceptive jammer would definitely have the quickest battery exhaustion as a result of the strategy it employs. The models above would go a long way in helping researchers in developing algorithms for Denial of Service attack mitigation in wireless sensor networks

6. Recommendation

For future work, it is recommended that the models be integrated into frameworks for jamming attack mitigation in Wireless sensor networks associated with IoT nodes. They should also be employed as a critical component of intrusion detection systems design in wireless sensor network based on the behavioral models provided. This holds true since the above models have shown clearly how these attacks operate and how they affect the respective nodes and networks.

References

[1]. Abror Abduvaliyev, Al-Sakib Khan Pathan, Jianying Zhou, Rodrigo Roman, and Wai-Choong Wong, "On the Vital Areas of Intrusion Detection Systems in Wireless Sensor Networks," IEEE Communications Surveys & Tutorials, vol. 15, no. 3, pp. 1223-12237, 2013.

[2]. R Swetha, Amarnath V Santhosh, and Sofia Anita, "Wireless Sensor Network : A Survey," International Journal of Advanced Research in Computer and Communication Engineering, pp. 2292–2330, 2018.

[3]. Babak D. Beheshti, "A Framework for Wireless Sensor Network Security," in IEEE Long Island Systems, Applications and Technology Conference (LISAT), Farmingdale, NY, USA, 2016, pp. 1-4.

[4]. Bo Yu Lu-Yong Zhang, "An Improved Detection Method for Different Types of Jamming Attacks in Wireless Networks," in The 2014 2nd International Conference on Systems and Informatics (ICSAI 2014), Shanghai, China, 2014, pp. 553-558.

[5]. Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal, "Wireless sensor network survey," Computer Network, vol. 52, pp. 2292–2330, 2008.

[6]. Qiuwei Yang, Xiaogang Zhu, Hongjuan Fu, and Xiqiang Che, "Survey of Security Technologies on Wireless Sensor Networks," Journal of Sensors, pp. 1-9, 2015.

[7]. Aykut Karakaya and Sedat Akleylek, "A Survey on Security Threats and Authentication Approaches in Wireless Sensor Networks," in 6th International Symposium on Digital Forensic and Security (ISDFS), Antalya, Turkey, 2018, pp. 1-4.

[8]. Aashima Singla and Ratika Sachdeva, "Review on Security Issues and Attacks in Wireless Sensor Networks," International Journal of Advanced Research in Computer Science and Software Engineering, p. 2, 2013.

[9]. Pavan Bhat. (2011, Aug.) sensors-and-networks.blogspot. [Online]. http://sensors-and-networks.blogspot.com/2011/08/physical-layer-for-wireless-sensor.html

[10]. Alhameed Ahmad Abed Alkhatib and Gurvinder Singh Baicher, "Wireless Sensor Network Architecture," in International conference on computer networks and communication systems, Singapore, 2012, p. 4.

[11]. Sunil Ghildiyal, Amit Kumar Mishra, Ashish Gupta, and Neha Garg, "Analysis of Denial of Service (DoS) Attacks in Wireless Sensor Networks," International Journal of Research in Engineering and Technology, p. 5, 2014.

[12]. Opeyemi Osanaiye, Attahiru S. Alfa, and Gerhard P. Hanck, "A Statistical Approach to Detect Jamming Attacks in Wireless Sensor Networks," Sensors, pp. 1-15, 2018.

[13]. Manos Antonakakis et al., Understanding the Mirai Botnet. Vancouver, BC: USENIX} Association, 2017.

[14]. Mingyan Li, Iordanis Koutsopoulos, and Radha Poovendran, "Optimal Jamming Attack Strategies and Network Defense Policies in Wireless Sensor Networks," IEEE Transactions on Mobile Computing , pp. 1119-1133, 2010.

[15]. Behrouz A. Forouzan, Data Communication and Networking. New York, USA: McGraw Hill, 2007.

[16]. Sachin D. Babar, Neeli R. Prasad, and Ramjee Prasad, "Jamming Attack: Behavioral Modelling and Analysis," in Wireless VITAE 2013, Atlantic City, NJ, USA, 2013, pp. 1-5.