Mitigating Black Hole Attacks in Wireless Sensor Networks Using Node-Resident Expert Systems

Vincent F. Taylor
Department of Computer Science
University of Oxford
Oxford, United Kingdom
Email: vincent.taylor@cs.ox.ac.uk

Daniel T. Fokum
Department of Computing
University of the West Indies
Mona, Jamaica
Email: daniel.fokum@uwimona.edu.jm

***Abstract*—Wireless sensor networks consist of autonomous, self-organizing, low-power nodes which collaboratively measure data in an environment and cooperate to route this data to its intended destination. Black hole attacks are potentially devastating attacks on wireless sensor networks in which a malicious node uses spurious route updates to attract network traffic that it then drops. We propose a robust and flexible attack detection scheme that uses a watchdog mechanism and lightweight expert system on each node to detect anomalies in the behaviour of neighbouring nodes. Using this scheme, even if malicious nodes are inserted into the network, good nodes will be able to identify them based on their behaviour as inferred from their network traffic. We examine the resource-preserving mechanisms of our system using simulations and demonstrate that we can allow groups of nodes to collectively evaluate network traffic and identify attacks while respecting the limited hardware resources (processing, memory and storage) that are typically available on wireless sensor network nodes.**

I. Introduction

Wireless sensor networks are groups of inexpensive, geographically distributed nodes that may be used to monitor such things as environmental pollution, vehicle safety, building safety, warehouse inventories and the health of patients. As the technology continues to mature, we expect to see more widespread deployments of these networks over a broader range of applications. Given the critical nature of some of these applications, it is important to ensure that the wireless sensor network architecture is robust enough to remain useful and secure even in the presence of adversaries.

Nodes are designed to be small, cheap and lightweight and thus have limited computing capacity and batteries. This poses a problem because the traditional cryptographic solutions that work for PCs and workstations are infeasible on these low-power sensor nodes [1]. The utility of wireless sensor networks sees them being deployed in dangerous or inaccessible areas and this adds to the complexity of securing the network since it is usually out of the physical reach of the operator. Adversaries therefore have the advantage of being able to steal nodes and reverse-engineer them to obtain cryptographic keys or other sensitive information. More worrying is the fact that adversaries can also introduce their own malicious nodes into the network to disrupt the regular flow of network traffic or to wreak havoc such as exhaustion, misdirection, greed, black holes and collisions [2].

Black-hole attacks, also called packet drop attacks or sinkhole attacks, are one type of denial-of-service attack that can be employed on an ad hoc network such as a wireless sensor network [2]. Given the collaborative nature of a wireless sensor network, a malicious node can severely impact the performance of the network by broadcasting false routing information. Traditional routing protocols for use on wireless sensor networks include the Ad-hoc On-Demand Distance Vector (AODV) [3] protocol and the Dynamic Source Routing (DSR) [4] protocol. AODV is a reactive, distance vector routing protocol and computes routes only when they are required. DSR is similar to AODV in many ways but its main distinction is that it uses source routing instead of relying on the routing table of intermediate devices. To execute a black hole attack on a network running DSR as its routing protocol, a malicious node would simply pretend to have a short route to the destination to trick other nodes into sending it traffic. This malicious node would then drop this traffic instead of passing it on as expected by the other nodes.

In this paper, we focus on identifying and mitigating the effects of a black hole attack on a wireless sensor network running DSR using a system called ADIOS: Advanced Detection of Intrusions On Sensor networks. This paper builds on the authors' previously published work [5] by explaining the ADIOS architecture in greater detail and providing results of simulations that were performed on the system. ADIOS uses a combination of a watchdog mechanism, similar to the one described in [6], as well as a node-resident expert system to make judgements on neighbour behaviour when trying to identify an attack. ADIOS is a novel means of network defence for wireless sensor networks because it uses an expert system for identifying and mitigating the effects of attacks. ADIOS is also very extensible since the knowledge that is fed into the expert system can be updated to add new "intelligence" without having to modify any of the existing architecture.

We developed and tested ADIOS using an expert system building tool called CLIPS: C Language Integrated Production System [7]. CLIPS likely requires too many resources to run on an average mote but we chose it for our research since it was

Vincent F. Taylor is currently a student at the University of Oxford but this research was done while he was a student at the University of the West Indies.

978-1-4799-1297-1/14/$31.00 ©2014 IEEE

lightweight, modular, portable, low-cost and easily extensible, mimicking all the features that a real-world implementation of an expert system for use on sensor nodes would have. Developing an expert system for use in wireless sensor networks using CLIPS required us to capture and engineer facts, rules and definitions to feed into the expert system so that it could then use its artificial intelligence to make decisions on whether a node on the network was behaving maliciously.

The rest of this paper is organized as follows: Section 2 gives background on black hole attacks in sensor networks and the mitigating strategies that other researchers have used. Section 3 describes the architecture of ADIOS and details strategies for reducing resource consumption. Section 4 shows the results obtained from our simulations and does an analysis on the significance of these results. Section 5 concludes the paper by summarizing our findings and suggesting areas for future work.

II. Background

Various means of attack on wireless sensor networks have been covered by the literature. Ahmed et al. [8] assert that "sensor networks are highly susceptible to denial of service attacks due to their inherent characteristics, i.e., low computational power, limited memory and communication bandwidth coupled with the use of an insecure wireless channel." Black hole attacks in sensor networks can be mitigated using authorization, monitoring and redundancy. Authorization uses cryptographic methods to ensure that nodes that are sending out routing updates do so legitimately and ensures that spoofed messages can be identified. Redundancy is a technique that takes advantage of multiple routing paths to the destination to mitigate the effects of black hole nodes. Monitoring aims to identify malicious activity by examining network announcements to identify anomalies.

Yin and Madria [9] describe an approach for black hole mitigation that works by introducing authorization and cryptography into the routing protocol. This is a valuable contribution to solving the problem, but it assumes that the cryptographic keys used by the nodes won't get in the hands of an attacker. Papadimitratos and Haas [10] also propose a secure routing scheme which they argue mitigates the effects of routing misbehaviour. This scheme also uses cryptographic methods to ensure that spurious route responses are rejected by recipients. Given that sensor nodes are usually small and out of the reach of the network operator, theft and reverse-engineering is a real possibility and it does not suffice to assume that the cryptographic keys will always be secret; thus we require alternate methods to protect against black hole attacks.

Marti et al. [6] propose a system that uses a watchdog and pathrater to examine neighbour behaviour and routing messages to determine whether an advertised path to a destination is valid and reliable. This watchdog and pathrater system dynamically values the confidence in a route advertised by a neighbour based on whether that same neighbour is overheard (in promiscuous mode) passing on messages that it promised it would. Karakehayov [11] proposes a system called REWARD that is suitable for use on sensor networks containing nodes that are able to tune their transmission power. The REWARD system detects black hole attacks by watching neighbour transmissions and taking advantage of broadcast inter-radio behaviour. REWARD then creates a database of suspicious nodes and their locations whenever anomalies are discovered.

Geographic routing protocols aim to make black hole attacks obsolete by using the location information from a node to determine the best routes to the destination [12]. These protocols show much promise but are unsuitable for use in some environments. Some geographic routing protocols rely on GPS information to identify the location of a node, but depending on the application, GPS modules may be too costly, too complex to integrate or require too much power. In the case where nodes are deployed undersea or underground, obtaining reliable location information from a GPS module may be very difficult.

Another valuable contribution to mitigating black hole attacks is the use of multipath routing. With a multipath routing scheme, more than one routes from source to destination are established to provide load balancing and/or fault tolerance [13]. Ganesan et al. suggest a multipath routing mechanism that aims to rapidly identify alternate paths between source and destination [14]. This approach could potentially be used to mitigate the effects of black hole attacks but does not seek to identify the actual malicious node that was participating in the routing misbehaviour in the first place. An ideal solution needs to not only alleviate the problem but also mitigate the cause of the problem.

Intrusion detection and prevention systems (hereafter called IDS) are designed to identify and prevent intrusions into a network. IDS are broadly divided into two categories: host-based IDS and network-based IDS. In wireless sensor networks, host-based IDS may be more suitable for mitigating black hole attacks since the protection does not rely on dedicated and trusted hardware on the network. The literature speaks of decentralized intrusion detection mechanisms for use on wireless sensor networks [15]. Existing systems check for message integrity, delays by neighbours and repeat transmissions from neighbours.

Krontiris and Dimitriou [16] focus on the MintRoute protocol used in TinyOS and suggest rules that can be used by an intrusion detection system to detect black hole attacks. Their intrusion detection system is a rule-based system but these rules would be typically implemented in programming code to solve a particular problem and thus this intrusion detection system may suffer from extensibility issues. Kachirski and Guha [17] also propose an intrusion detection system for use in wireless sensor networks. In this system, they use mobile agents for network monitoring to increase modularity and allow additional functionality to be built into the IDS with minimal effort.

ADIOS fills some gaps in the existing literature by providing defence against black hole attacks without relying on cryptographic techniques or new routing protocols that may be infeasible for some applications. ADIOS not only alleviates the problems caused by a black hole node, but it actively seeks to identify the malicious node responsible for the routing misbehaviour in the first place. ADIOS also seeks to be more robust and extensible than existing host-based IDS by utilizing an expert system for the decision making process. By using an expert system, the "knowledge" used to identify black hole attacks, and indeed other attacks, can be continuously updated when new techniques are identified without requiring much change, if any, to the underlying IDS framework.

In the following section, we take a closer look at the architecture of ADIOS and examine its resource-preserving features.

III. Expert System Based Intrusion Detection

During the development of ADIOS, four major assumptions were made about the properties of the nodes on the wireless sensor network. The assumptions are outlined below:

- The wireless network interface cards on the nodes being used are capable of promiscuous mode. If Node A is in promiscuous mode, it means that it would be able to overhear and process transmissions from Node B as long as it is within range of Node B, even if Node B was actually sending this data to another node, Node C.
- The antennas on the nodes being used would be omnidirectional antennas. This assumption does not need to strictly hold, but having omnidirectional antennas in use on the network makes the watchdog mechanisms on each node more efficient since they would be better able to hear transmissions from neighbouring nodes.
- There is bidirectional communication symmetry between nodes. This means that if Node A is able to receive a transmission from Node B at a particular time, then Node B could instead have received a transmission from Node A at that time.
- The routing protocol in use on the network is the Dynamic Source Routing protocol. DSR is a popular choice of routing protocol for use on wireless sensor networks. Although ADIOS was developed to mitigate black hole attacks on networks running DSR, many of the concepts can be generalized to work with other routing protocols.

A. The Dynamic Source Routing Protocol

The Dynamic Source Routing protocol is an on-demand source routing protocol commonly used in wireless sensor networks. DSR uses two types of routing packets for finding routes to destinations: Route Request Packet (RREQ) and Route Reply Packet (RREP). An RREQ contains the address of the destination device and is flooded to all devices on the network. When a node receives an RREQ with its address, it creates an RREP in response and sends it back to the original sender. An example of route discovery in DSR is shown in Figure 1 and explained below:

(a) Node S wants to find a route to Node D so it broadcasts an RREQ with the address of Node D to its neighbours.

(b) A node receiving an RREQ will check to see if the packet is addressed to it. If the RREQ is not addressed to the node, the node appends its address to the packet before broadcasting it.

(c) When Node D receives the RREQ with its name, it responds with an RREP addressed to Node S. Node D will retrieve the addresses that were appended to the RREQ and put them in the RREP. Node D now has enough information to route the packet back to Node S.

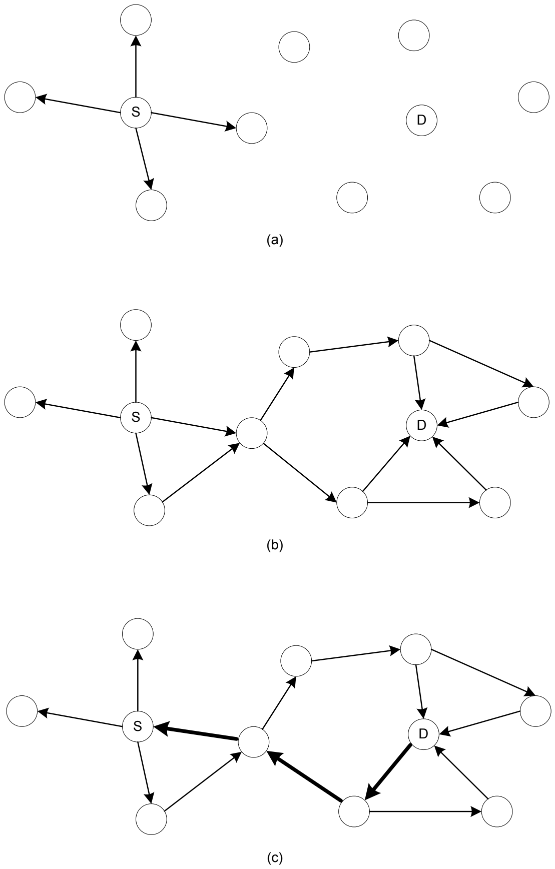

Fig. 1. Example of route discovery using the DSR protocol.

From Figure 1 we see that there may be many alternate paths between a source and destination. DSR chooses the route with the "best" metric and caches the remaining routes as backups for future use. A malicious black hole node would respond to RREQ packets, falsely claiming that it has the best routes to the destination. It does this to trick the sending node into routing information through it; this information then gets dropped and never reaches the destination.

B. The Watchdog Mechanism

The watchdog system is used to provide input to the intrusion detection system and is the interface between ADIOS and the network. The watchdog system takes advantage of the promiscuous mode feature of wireless interface cards to overhear transmissions from neighbouring nodes. Using the watchdog mechanism, a node can listen to see if a neighbouring node did indeed forward a packet that it claimed to have had a path to.

Figure 2 shows how the watchdog system used by ADIOS works. Node S wants to send some data to Node D via nodes A, B and C. Node A does not have enough power to transmit all the way to Node C so it transmits to Node B as an intermediary. Although Node A is out of the radio range of Node C, it is usually able to overhear what Node B sends to Node C since

Node B is still within its reception range. It is important to note that due to physical characteristics of the transmission medium or lower layer collisions, not all transmissions from B to C will be heard by A all the time.

The watchdog system, in addition to listening in promiscuous mode, maintains a table of what has been transmitted and overheard and provides the main input of data to the expert system. It is from this data that the expert system then makes inferences to determine whether attacks are happening on the network.

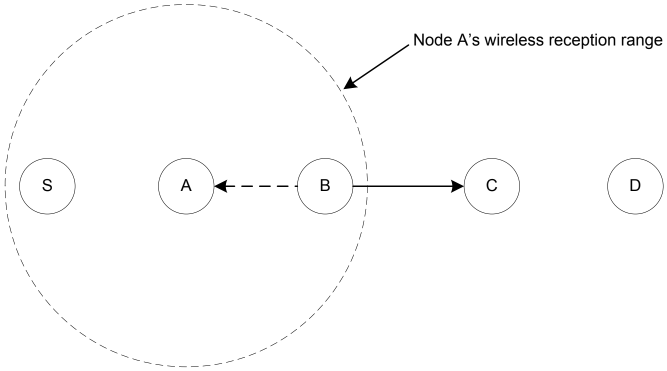

Fig. 2. Diagram of how a watchdog system works.

It is important to note that this watchdog system needs to know where packets are supposed to be in two hops in order to function optimally. This is why it is suitable for use on networks using a source routing protocol. If there is not enough information to determine where a packet should be in two hops, a malicious node may "forward" a packet to a non-existent node to trick the watchdog into assuming that the packet was transmitted to a legitimate next hop.

C. The Architecture that ADIOS Uses

ADIOS is comprised of five main components: lightweight expert system, expert knowledge, memory resident table, majority voting system and the watchdog system. Figure 3 demonstrates how the watchdog fits in with the rest of ADIOS.

1) Lightweight Expert System: The lightweight expert system (LES) lies at the center of ADIOS and provides the intelligence that drives the IDS. This LES is where the machine reasoning and inferencing happens and it is from here that judgements are made regarding suspicious network activity. The LES takes input directly from all other modules except the watchdog system. The LES passes data to and from the majority voting system and memory resident table.

2) Expert Knowledge: The expert knowledge module contains the rules, definitions and sequence of events that describe to the LES what a black hole attack looks like. This expert knowledge is effectively a definitions file that is processed by the LES and as such, the expert knowledge can be swapped out for other expert knowledge which gives ADIOS extensibility in detecting new attacks and variations on old attacks. Constructing expert knowledge requires understanding the features of the ADIOS architecture as well as the language used by the LES at the center. Once these requirements are met, it is a trivial matter to construct expert knowledge to detect attacks once the author has an understanding of how the particular attack works.

3) Watchdog System: The watchdog module contains the network interface card (in promiscuous mode) and has the ability to read and write to the memory resident table. The watchdog records activity of interest in the memory resident table for the LES to analyze. This watchdog is similar to the one proposed by Marti et al. in [6]. It takes advantage of the fact that an interface in promiscuous mode is able to receive transmissions bound for other destinations and can leverage this information to identify routing misbehaviour on the network.

4) Memory Resident Table: The memory resident table (MRT) is basically a common workspace that ADIOS uses. Network events of interest, such as route requests, route replies and abnormal forwarding by a neighbour are stored temporarily in the MRT for processing by the LES. The MRT is expected to be the subsystem that consumes the bulk of the memory used by ADIOS. A bigger MRT can keep track of more network events and may be more effective at identifying attacks but the resource-constrained nature of sensor nodes dictates that trade-offs be made with regards to memory consumption and attack detection accuracy.

5) Majority Voting System: The majority voting system is partly responsible for mitigating attacks on the network. When invoked, the majority voting system communicates with the node's neighbours regarding a potential attack. By using majority voting, more than one node on the network decides whether a particular node is behaving maliciously. This leads to a more robust system since the network would take into consideration the "opinion" of more than one nodes when trying to determine if a node is behaving maliciously. Once a malicious node is decided on, it can be evicted from the network through blacklisting and/or key revocation. The literature goes into detail regarding majority voting [18] and

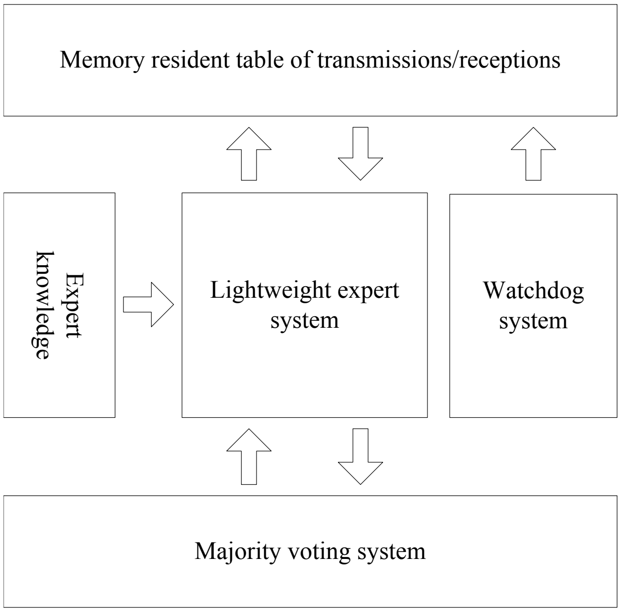
Memory resident table of transmissions/receptions
Expert knowledge
Lightweight expert system
Watchdog system
Majority voting system

Fig. 3. Architectural diagram of ADIOS.

key revocation [19]. The inner workings of majority voting and key revocation protocols are outside of the scope of this paper.

D. How Attacks are Detected

After the watchdog module captures traffic of interest, it is passed on to the LES via the MRT. It is the LES that makes the final decision on whether malicious activity has occurred. The MRT constantly has new facts being added to it and these facts will remain there until the attack detection process (ADP) is invoked. The ADP is run at preset (and customizable) intervals and it is at this time that the LES uses its expert knowledge to make inferences about the data in the MRT.

ADIOS is designed so that data related to normal node behaviour is retracted when the ADP is run. When anomalies are detected, the data related to the anomaly is flattened into a single fact that represents the anomaly and this fact is then handled. Thus at the end of an ADP the MRT will be empty. The interval that each ADP is run at is determined by any one or a combination of the following triggers:

1) Time-based Triggers: Using time-based triggering, the ADP will be run at preset or random intervals. As an example, a preset interval of one minute will cause the ADP to be triggered every minute. Using random intervals, the ADP will be run at intervals chosen at random between a minimum and maximum threshold. Running the ADP at random intervals mitigates an adversary's ability to attack at specific times to go longer without being detected if they were to discover the intervals that nodes ran their ADP at.

2) Memory-based Triggers: Using memory-based triggering, the ADP will be run any time the contents of the MRT reaches a predefined size threshold. Like time-based triggers, memory-based triggers can also be adjusted depending on the nature of the application. This allows the user to prevent ADIOS from consuming too much memory. Memory-based triggering can also be combined with time-based triggering whereby the ADP would be run at a particular time or when the MRT reached a predefined size, whichever came first. This would give ADIOS the flexibility of random intervals while ensuring that a predefined maximum amount of RAM was not exceeded.

3) Event-based Triggers: Using event-based triggering, the ADP will be run any time that specified network events happen. Depending on the application, it may make sense to run the ADP after a certain amount of RREQ or RREP have been sent on the network. This gives the operator added flexibility with regards to when the ADP is run. Of course, event-based triggering can also be combined with time-based or memory-based triggering as required.

E. How ADIOS Responds to Intrusions

ADIOS handles routing misbehaviour and network anomalies in one of two ways:

- Manipulating local routing table metrics
- Invoking the majority voting subsystem

ADIOS is able to mitigate the effects of a black hole in the network by choosing alternate routes to send data and manipulating the metrics in the local routing table so that these routes are preferred by the node. This is an easy way to mitigate the effect of a black hole from the local node's point of view without relying on the majority voting subsystem to involve external nodes.

ADIOS can also mitigate the effects of a black hole in the network by invoking the majority voting subsystem to get a consensus from other nodes on the network with regards to whether a node should be evicted. What action to take in the event of routing misbehaviour is up to the operator of ADIOS but it might be judicious to change local routing table metrics for minor infractions and invoke the majority voting process when major malicious behaviour is concretely observed.

F. Reducing the Amount of Resources that ADIOS Consumes

Sensor nodes have limited processing power and RAM and this limits the effectiveness of any IDS that is installed on them. Below we look at ways of reducing memory consumption and processor utilization:

1) Reducing Memory Consumption: Instead of storing all network events in the MRT, a random subset of these events can be stored instead. By randomly choosing what network events to monitor and store, ADIOS can consume less memory while mitigating an adversary's ability to use a selective forwarding attack to circumvent the protection offered by the system. The attacker would not be able to know, a priori, what network events were being monitored and thus unable to tailor their attack to take advantage of that.

Given that there is usually more than one pair of nodes in a neighbourhood, with each node monitoring a subset of network events, as the size of the neighbourhood gets larger, so does the probability that more than one node is monitoring the same event. Thus we can have each node monitoring a subset of network events to conserve memory, but the neighbourhood on a whole would have seen much more of the total network traffic.

We can also reduce memory consumption by running the ADP more often. Recall that at the end of the ADP, the MRT is empty. Therefore, by increasing the rate at which we run the ADP, the average memory consumption on the node would be reduced, since the MRT would be emptied more frequently.

2) Reducing Processor Utilization: The technique of monitoring only a subset of network events described above also reduces the processor utilization. This is because less network events mean that there is less to process when the ADP is run and that the ADP can be run less frequently.

To further limit the processor utilization, small expert knowledge can be loaded into the LES. Smaller expert knowledge will require less processing to evaluate and this will correspond to lower processor utilization across the system. Using this method, we sacrifice versatility in detecting various attacks in favour of lower processor utilization. Of course, the expert knowledge used in each application would be chosen based on the capability of the hardware.

G. Simulating a Lightweight Expert System on Nodes

To simulate CLIPS on nodes, we used the CLIPS Java Native Interface (JNI) which is a library that allows Java applications to communicate with a CLIPS backend. Nodes were modelled as Java program, each having their own ADIOS implementation and CLIPS backend.

We used purpose-written Java simulators to evaluate the performance of the ADIOS architecture on nodes in a wireless sensor network. We opted to use purpose-written simulators instead of existing simulators, such as ns3 [20], because we wanted to analyse architecture-specific artefacts such as resource consumption and how the MRT behaved under different ADP triggering regimes. We also wanted to analyse how our resource optimization policies performed at the IDS architecture level with limited regard to the lower layer metrics that would be provided by ns3 and other simulators.

IV. Simulation Results and Analysis

To examine how memory consumption varied with the frequency of ADP invocation, we simulated a single sensor node running ADIOS. Network events were fed into the node via an M/D/1 queue for various intervals of ADP invocation and the average memory consumption was measured. Memory consumption was calculated as the amount of network events in the input buffer waiting to be processed. This data is presented in Figure 4.

From the diagram we see that the average memory consumption is less when the ADP is run more frequently. This suggests that the average memory requirement of ADIOS can be reduced by running the ADP more frequently. With no concern for processing power or battery life, ADIOS could be tuned to consume negligible memory, but since processing and battery need to be conserved this is not a feasible solution.

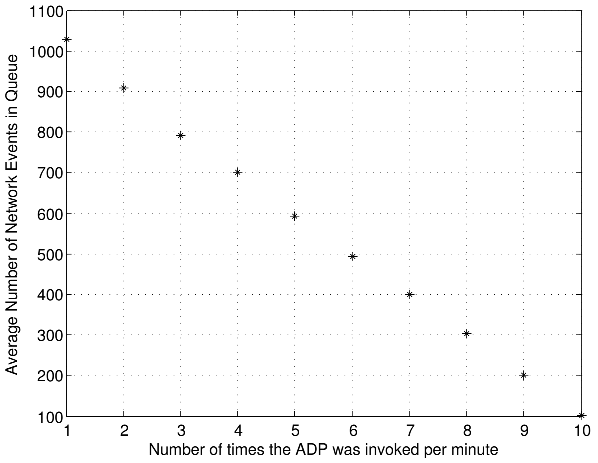

Fig. 4. Average memory consumption vs. Frequency of ADP invocation.

One strategy for combating the problem of limited resources is to have each instantiation of ADIOS only monitor a random subset of network events instead of greedily trying to monitor all network events. We leverage the fact that several nodes in a neighbourhood, each monitoring a fraction of events, are able to collectively cover a larger share of total network events than a single node by itself.

To simulate the efficacy of each node monitoring a random subset of network events, random network traffic containing black hole attacks was generated. These attacks randomly happened throughout the traffic dump. Nodes were instructed to only monitor random portions of this traffic and report when anomalies were found. The number of nodes in a neighbourhood and the number of attacks were varied during the simulations. The simulations were run 1000 times, averages taken, and the results shown in Figure 5.

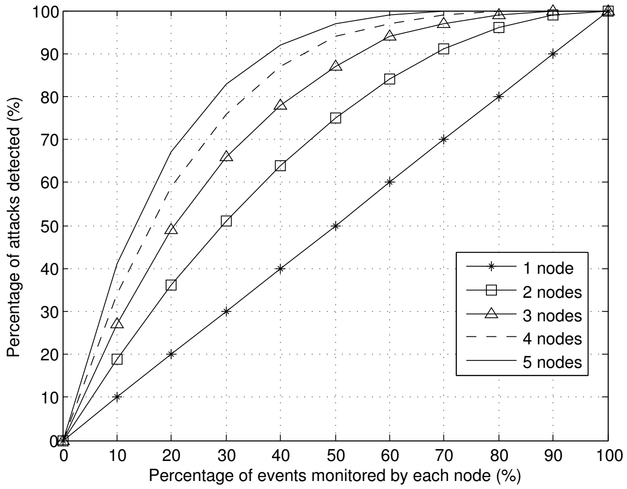

Fig. 5. Percentage of attacks detected vs. Percentage of events monitored.

In the trivial case, we see that a single node detects the same percentage of attacks as the percentage of traffic that it analyses. When more nodes are added to a neighbourhood, a greater number of attacks is detected even though the percentage of packets being analysed stays the same for each node. This is because multiple nodes randomly analysing traffic increases the total amount of unique traffic that is analysed by the neighbourhood as a whole. Our simulations show that 1 node listening to a random 30% of network events identifies 30% of attacks, while 5 nodes listening to their own random 30% of network events are able to identify more than 80% of network attacks. As the number of nodes continues to increase, we notice that the benefit to be gained (in terms of percentage of attacks detected) from increasing the amount of nodes decreases.

Sensor nodes are also limited by processing power and battery life. Running the ADP more frequently leads to more processor utilization and thus greater demand on the battery. To reduce energy consumption the ADP can be triggered less frequently, but this introduces the risk that attacks will go undetected between runs of a node's ADP. This concern is mitigated when we consider that there are several nodes in a neighbourhood and each node would run its ADP at a different time, thus the network as a whole would be able to detect attacks in a shorter time than the lowest ADP interval on the network.

To evaluate the efficacy of this idea, we generated network events containing black hole attacks randomly distributed throughout and recorded the number of attacks that were present as well as the time that these attacks happened. In this simulation, nodes used one minute intervals between ADP runs but chose a random time to start their first ADP. The results we obtained are shown in Figure 6.

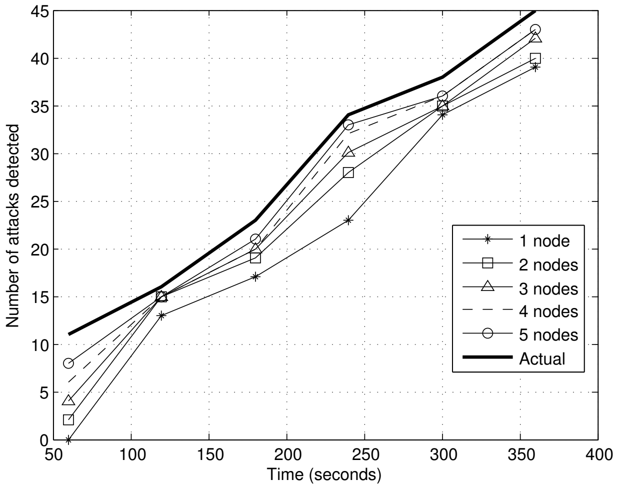

Fig. 6. Number of attacks detected in a given time based on neighbourhood size.

From the figure, we see that as the number of nodes in a neighbourhood increases, the number of attacks detected after a particular time approaches the actual number of attacks that exist in the network traffic. This suggests that we can conserve on processing and battery by running the ADP on each node less frequently, while being able to detect attacks in less time than the lowest ADP interval being used.

V. Conclusion and Future Work

In this paper, we demonstrated that it is possible to detect black hole attacks on wireless sensor networks using an IDS based on an expert system. We detailed the architecture of such a system, which we call ADIOS, and suggested ways of reducing its resource consumption so that it would be able to function properly in a resource-constrained environment. Using simulations, we demonstrated that we can leverage the fact that multiple nodes are on a network to reduce the burden of processing, memory and battery life on each node while still maintaining the ability to detect attacks timely and accurately.

We plan to extend this research by developing a lightweight expert system suitable for use on low-power devices before deploying ADIOS on physical hardware to test its real-world performance in terms of resource consumption and attack detection accuracy. We also plan to do further research and simulations to examine how ADIOS performs under varying levels of sensor node mobility. Plans are also on the horizon to generalize our approach to detecting multiple denial-of-service attacks against wireless sensor networks. This research is expected to take us a step closer to perfect security for sensor networks.

References

[1] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar, "SPINS: Security protocols for sensor networks," in *Wireless Networks*, 2001, pp. 189–199.

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

[3] C. Perkins and E. Belding-Royer, "Ad hoc on-demand distance vector (AODV) routing," RFC 3561, Tech. Rep., Jul. 2003.

[4] D. Johnson, Y. Hu, and D. Maltz, "The dynamic source routing protocol (DSR) for mobile ad hoc networks for IPv4," *RFC4728*, pp. 2–100, 2007.

[5] V. Taylor and D. Fokum, "Securing wireless sensor networks from denial-of-service attacks using artificial intelligence and the clips expert system tool," in *Southeastcon, 2013 Proceedings of IEEE*, 2013, pp. 1–6.

[6] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *International Conference on Mobile Computing and Networking*. ACM, 2000, pp. 255–265.

[7] G. Riley, "CLIPS: An expert system building tool," in *The Second National Technology Transfer Conference and Exposition, NASA, Washington*, vol. 2, 1991.

[8] N. Ahmed, S. S. Kanhere, and S. Jha, "The holes problem in wireless sensor networks: a survey," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 9, no. 2, pp. 4–18, Apr. 2005.

[9] J. Yin and S. Madria, "A hierarchical secure routing protocol against black hole attacks in sensor networks," in *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, 2006*, vol. 1, Jun. 2006.

[10] P. Papadimitratos and Z. J. Haas, "Secure routing for mobile ad hoc networks," in *Proceedings of the SCS Commnication Networks and Distributed Systems Modeling and Simulation Conference (CNDS)*, 2002, pp. 193–204.

[11] Z. Karakehayov, "Using REWARD to detect team black-hole attacks in wireless sensor networks," *Wksp. Real-World Wireless Sensor Networks*, pp. 20–21, 2005.

[12] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking*, ser. MobiCom '00. New York, NY, USA: ACM, 2000, pp. 243–254.

[13] S. Mueller, R. Tsang, and D. Ghosal, "Multipath routing in mobile ad hoc networks: Issues and challenges," in *Performance Tools and Applications to Networked Systems*, ser. Lecture Notes in Computer Science, M. Calzarossa and E. Gelenbe, Eds. Springer Berlin Heidelberg, 2004, vol. 2965, pp. 209–234.

[14] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly-resilient, energy-efficient multipath routing in wireless sensor networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 5, no. 4, pp. 11–25, Oct. 2001.

[15] A. da Silva, P. R., M. H. T. Martins, B. P. S. Rocha, A. A. F. Loureiro, L. B. Ruiz, and H. C. Wong, "Decentralized intrusion detection in wireless sensor networks," in *Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks*, ser. Q2SWinet '05. New York, NY, USA: ACM, 2005, pp. 16–23.

[16] I. Krontiris, T. Dimitriou, T. Giannetsos, and M. Mpasoukos, "Intrusion detection of sinkhole attacks in wireless sensor networks," in *Algorithmic Aspects of Wireless Sensor Networks*, ser. Lecture Notes in Computer Science, M. Kutylowski, J. Cichon, and P. Kubiak, Eds. Springer Berlin Heidelberg, 2008, vol. 4837, pp. 150–161.

[17] O. Kachirski and R. Guha, "Intrusion detection using mobile agents in wireless ad hoc networks," in *Proceedings of IEEE Workshop on Knowledge Media Networking*, 2002, pp. 153–158.

[18] F. Chou and J. Tan, "A majority voting scheme in wireless sensor networks for detecting suspicious node," in *Second International Symposium on Electronic Commerce and Security, ISECS '09*, vol. 2, May 2009, pp. 495 –498.

[19] G. Dini and I. Savino, "An efficient key revocation protocol for wireless sensor networks," in *International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM*, 2006, pp. 452–454.

[20] T. R. Henderson, S. Roy, S. Floyd, and G. F. Riley, "ns-3 project goals," in *Proceeding from the 2006 workshop on ns-2: the IP network simulator*. ACM, 2006, p. 13.