Energy-based adaptation in simulations of survivability of ad hoc communication

Massimiliano Raciti, Jordi Cucurull, Simin Nadjm-Tehrani
Department of Computer and Information Science, Linköping University
SE-581 83 Linköping, Sweden
Email: [massimiliano.raciti,jordi.cucurull,simin.nadjm-tehrani]@liu.se

***Abstract*—Mobile wireless handheld devices can support ad hoc communication when infrastructure systems are overloaded or not available. Unfortunately, the constrained capacity of their batteries and the energy inefficiency inherent to the ad hoc communication poses a challenge causing a short lifetime. Protocols and application layer services, such as security, can be designed (offline) to do an efficient use of the resources. Real-time adaptation can further minimise their impact on the energy consumption, increasing the network lifetime thus extending the availability of network communication.**

In this paper, we propose an energy-aware adaption component for an Intrusion Detection System (IDS) in mobile ad hoc networks (MANET). The component is in charge of adjusting the parameters of the IDS based on the current energy level, using the trade-off between the node's response to attacks and the energy consumption induced by the IDS. The approach is based on a model for accounting CPU energy consumption in network simulation, which has been implemented in an existing IDS in ns-3. Simulations demonstrate that the adaption has a positive impact on the battery life time, increasing it by 14%, without deteriorating the network-wide performance of the IDS.

***Index Terms*—Adaptation, energy-awareness, CPU model, energy modelling, survivability, intrusion detection.**

I. Introduction

The prevalence of handheld devices such as smartphones will bring with it unforeseen opportunities for cooperation and distributed sensing. While the infrastructure-based mode of communication (cellular, WiFi) provides almost continuous connectivity in time and space, it typically does not pay attention to handheld devices' energy constraints when optimising networking algorithms. On the opposite end of the spectrum we have the totally distributed infrastructure-less mode of communication whereby each device opportunistically connects to neighbours in its vicinity in order to establish an ad hoc chain of dissemination, collaboration, or distributed sensing.

The deployment of opportunistic ad hoc communication scenarios is unlikely on a large scale. Part of it is due to business models and lack of trust and adequate security mechanisms. Another major obstacle is the inherent inefficiency of using the ad hoc interface in current handsets. However, the distributed nature of this setting makes it robust to failures, and interesting as a platform for studying novel ideas in the distributed setting. This paper addresses the energy issue in distributed ad hoc communication, by providing means to study the (global) network life time in presence of (local) node level energy-based adaptation.

978-1-4577-2028-4/11/$26.00

All handheld devices are power-hungry. In order to extend the operating life time of a set of cooperating nodes protocol design has a role to play. It may economise the use of energy, by using low signalling overhead, both in terms of message transmissions and CPU operation. However, other layers are also important; both application and other service layers such as security. In most cases, the adoption of security solutions is in fact hampered due to the power drain on the handsets. So in some sense, making studies on the security mechanisms' energy footprint is useful no matter what mode of communication is envisaged.

When it comes to security, there is an obvious trade-off. We get more protection if we have endless energy. In this paper we show that we can adapt the sensitivity of security mechanisms, tuning them based on energy estimates. We demonstrate this idea of energy-based adaptation using an ad hoc protocol that was created for surviving maximally in a disaster scenario, even in hostile environments. This dissemination protocol and the associated general survivability framework, in which local anomaly detection, diagnosis, and mitigation are part of the application needs, have been developed in a larger project on Hastily Formed Networks [1] and published elsewhere [2], [3], [4]. This paper focuses on how the environment for large scale studies, such as ns-3 simulation platform can be extended in order to model energy-based adaptations of protocol and service layer modules. This will enable studies of network life time in presence of various threats and different mobility patterns. The paper also points out how each protocol/service that is subject to study should be studied from an energy perspective – both in terms of CPU usage and transmission power.

The contributions of this work are as follows:

1) We present the impact of energy-aware adaptation for a network (protocol) level anomaly detection architectures.
2) We demonstrate that evaluation of energy-aware adaptation can be based on fairly simple models of CPU utilisation applied to networking protocols in simulation platforms, thus enabling evaluations of communication scenarios that are hard to evaluate by large scale deployments.
3) We illustrate the above contributions on top of an

energy-efficient protocol and an intrusion detection framework earlier devised for disaster area scenarios, and show the extended life time of the network despite attack-induced energy drain and protocol/IDS overhead.

II. Related Work

The work presented in our paper brings together an adaptive application, an IDS for MANET, with real-time energy awareness. There is a broad variety of approaches of intrusion detection applied to MANET (the interested reader is referred to [5]). The energy aspect, when considered, is either used to balance the workload in hierarchical IDSs [6], optimised offline in terms of power consumption of the IDS itself [7], or used as a feature to detect anomalous conditions from suspicious discharge behaviours [8]. With regard to adaptation, some approaches include techniques of tuning the detection thresholds to cope with the changes in the network [9] [10]. On-line energy-based adaptation is not covered in any of the reported approaches.

In an early work, energy aware adaptation has been introduced in the context of mobile computing by Flinn and Satyanarayanan [11] [12]. The operating system is in charge of monitoring the energy supply and adapting multimedia applications, degrading their quality according to the decreasing energy availability. More similarly to our case, self-adaptation at application level is proposed by Peddersen and Parameswaran [13]. This work first suggests some techniques to generate self-adaptive applications, such as inserting code that behaves differently based on the available energy. Two online adaptation algorithms specifically targeting multimedia processing are then proposed.

A. Adaptive security

Our approach adapts the tradeoff between the sensitivity of the intrusion detector and its energy consumption. The adaptation of the tradeoff between security provisioning and resource consumption is currently a hot topic. A framework for self-adaption of security at application level is proposed by Ferrante et al. [14]. This work proposes a domain-independent approach to adapt the security policies depending on the current state and the security requirements. In the work of Chigan et al. [15], a preliminary offline optimisation methodology is proposed to select, among all the available security services, the suboptimal candidate sets of cross-layer security protocols that guarantee the minimum redundancy of functionalities and performance cost at the desired level of security. The online self-adaptive module adapts the security level based on the perceived malicious activity. Although there is resource-awareness, real-time adaptation is triggered by malicious activity instead of energy level. Switching security policies causes high signalling overheads.

B. Energy modelling in simulations

Energy modelling in network simulations is required to perform energy evaluation of protocols and applications and to enforce energy awareness. An energy model for the ns-3 simulator is presented in [16]. In this framework, the device energy consumption and energy source are modelled as two separate elements. Although there are many energy source models, characterised by different discharge curves, there is currently only one device energy model available, the WiFi energy consumption model.

A similar energy model targeting Wireless Sensor Networks (WSN) has earlier been proposed by Chen et al. [17] for OMNeT++ [18]. In addition, a simple CPU model, that accounts the energy consumption in the active or inactive state, is included. For the same simulation platform, a generic energy model for wireless networks has been proposed by Feeney et al. [19]. It improves the battery depletion handling, compared to the existing model, but CPU energy consumption is not modelled. The common limitation of network simulators is that they normally assume unlimited computational power and ignore process execution time. This hinders CPU modelling and more detailed energy accounting.

Simulation of power consumption of programs running on real systems can be done with instruction sets simulators, such as ARMulator [20] or SimpleScalar [21] among the others. Unfortunately, they are not suitable for simulating complex programs, neither can they simulate networking scenarios.

Combined network simulation and CPU instruction set simulation has been proposed in SunFlower [22], Real-Time Network Simulator (RTNS) [23] or SliceTime [24]. However, the first lacks nodes mobility support, while the two latter works do not simulate energy consumption. In all the cases the CPU instruction set simulator and the network simulator need to be synchronised, and the simulation time is rather slow. The CPU energy model for network simulation proposed in our work could represent a way to include CPU energy consumption, thus enabling adaptation studies based on energy.

III. Background

The General Survivability Framework (GSF) is a modular architecture designed to provide a comprehensive security approach for mobile ad hoc communications in challenging environments, such as disaster area networks. In such environments, with the hypothesis that spontaneous ad hoc networks need to be created on the fly, pre-existing trust relationships among nodes cannot be assumed, limiting the use of encryption-based or collaborative protection mechanisms. The GSF, instead, is a standalone architecture which is installed on each node. The framework is composed of four independent modules that cooperate in order to detect, diagnose, and react to network attacks (see Figure 1). The first module, the anomaly detector [3], is responsible for monitoring the network traffic from the vicinity in order to detect anomalous conditions. It will have to be trained in order to build a normality model prior to deployment in an attack state. When an alarm is raised, the diagnosis component is triggered to classify the attack type among the known cases. Then the mitigation component is engaged to apply the appropriate countermeasures to reduce or eliminate the attack impact. Otherwise, if the attack cannot be matched to any of the known

cases, a generic mitigation strategy is employed. The adaption module is responsible for changing the parameters of the other components in order to cope with the changes in the state of the network and the node. The work presented in this paper is focused on the development of this module. Figure 1 shows

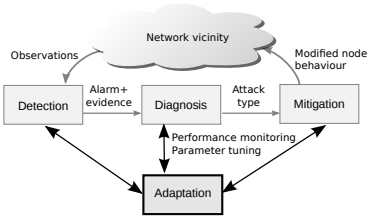


Fig. 1: The General Survivability Framework control loop

the interconnections of the four modules, which is similar to a closed loop control system, with the difference that the state of a given node is dependent on the global state of the network, i.e. the collective behaviour of other nodes. The framework has been tested on top of Random-Walk Gossip (RWG) [2], a manycast partition-tolerant protocol designed to efficiently disseminate messages in disaster area networks.

IV. Adaptive detection

The adaptation process normally involves monitoring the system under control, detecting changes, deciding and reacting to adjust the system parameters in order to bring it to the desired state, which often optimises towards some target performance. The survivability framework presented earlier differs from this concept due the fact that the state of the network is not fully observable and controllable from the point of view of an individual node, which has a partial view restricted to its vicinity. The emerging global response of the network determines a node's reaction to the attack. In this context the adaptation process that we aim for is a self-adaptation [25] approach, meaning that the control system itself (i.e. the GSF) should be adjusted with the changing conditions of both the node and the network. The adaptation solution proposed in this paper takes into account the perceived state of the network, focusing mainly on the most valuable feature of the internal state of the node, the energy. The assumption is that while supporting resistance to the attacks, the nodes should also be aware of their energy budget, adapting their behaviour in an efficient way in order to extend their lifetime as much as possible. With the support of energy modelling and simulation, we show that aggressive energy-agnostic attack survivability strategies, with excellent detection performances, could become useless once their impact on the energy consumption reduces the lifetime of the network.

A. Adaptation component

The adaptation component proposed (see Figure 2) consists of a function that takes as input the current energy level of the node, the perceived attack situation and the current parameter set configuration. The output is a new set of parameters that

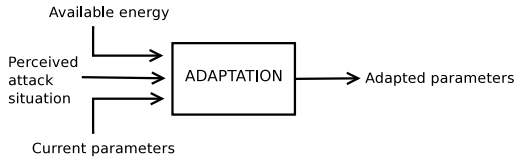


Fig. 2: Energy-aware adaptation of IDS parameters

is both relevant for the detection accuracy and has a strong impact on the energy consumption.

The function is based on a decision table. Each action is a parameter set configuration according to the perceived attack situation, modelled as conditions, and different energy states, modelled as rules. Rules based on energy levels could make the system reactive to energy changes, tilting the trade-off towards the energy aspect when this resource is limited. Prediction of the energy depletion based on the actual traffic load and CPU consumption could be the basis of a more proactive adaptation of the IDS parameters. All the combinations can be selected during a pre-design study on the trade-off between resource usage and security provisioning.

B. Case study

As a case study, we consider the GSF presented earlier. In this framework, there is a relevant parameter that governs the detection, diagnosis and mitigation cycle and has a strong impact on the CPU utilisation. It is the aggregation interval I_a in which the network state observations are aggregated and the alarm state is evaluated. The shorter the interval the faster the response to the attacks, but at the same time the detection accuracy is lower and the power consumption is higher. The longer the interval, the better the detection accuracy, but the detection latency would increase as well. In this case the power consumption of the detector would be lower, but a high latency could allow the attack to spread throughout the network causing subsequent negative consequences.

In an extension of the work presented in [3], detection accuracy and minimum latency has been obtained with I_a at 50 seconds. If we consider the energy aspect using the energy model proposed in section V-A, a longer interval which gives a better detection accuracy at the cost of a higher latency could still be acceptable if the overall impact on the energy consumption outperforms the case in which the latency is shorter.

The decision table implemented in this scenario takes into account two conditions: attack or non attack condition. A reactive solution has been implemented considering the following four energy level ranges: 100% to 60%, 60% to 40%, 40% to 20%, and 20% to 0%. Since, as mentioned earlier, the global response to the attack is given by the collective response of the nodes in the network, the choice of the aggregation interval is based on the idea that nodes with more energy should compensate the higher latency of nodes with a limited battery level that try to extend their lifetime slowing themselves. Nodes that have a battery level bounded between 100% to 60%, in normal conditions, set the shortest interval I_a

Energy fraction (%)	Attack	Non-attack
100-60	IDS_FAST	IDS_MEDIUM
60-40	IDS_FAST	IDS_SLOW
40-20	IDS_MEDIUM	IDS_SLOW
20-0	IDS_SLOW	IDS_SLOW

TABLE I: Aggregation interval based on energy levels

(namely IDS_FAST), to react more quickly in case of attacks. During attacks, instead, the interval is set to longer value (IDS_MEDIUM) in order to improve the detection accuracy and save energy at the same time. With a lower battery level, between 60% and 40%, the interval during attack is further increased (IDS_SLOW), to save more energy. When the level is between 40% and 20% the interval during attack-free periods is set to IDS_MEDIUM, since at that time the nodes should start preserving battery more consciously. Below 20%, the node samples slower in all the cases, to preserve its energy. Table I summarises the interval adaptations.

V. Energy modelling

Energy aware applications require access to the energy level of the system in which they are running. As mentioned in section II, network simulators focus on the energy consumption of the network interface, ignoring the contribution of other power-hungry components such as the CPU.

In this section we propose a simple model for accounting CPU energy consumption in network simulators.

A. A CPU model for network simulation

Our model is based on the assignment of an energy footprint to each application simulated. The model consists of a state machine for each application that runs in the simulation environment. Each state represents a possible operational mode of the application, that differs from the others in terms of behaviour and consequent CPU usage. By the application running at each of its states in a real device, the isolated impact on the power consumption can be profiled. Functions that produce power consumption values depending on the inputs of the application which have an impact on the energy consumption can then be associated to the corresponding states. In the case in which the power consumption cannot be directly measured, one could for example analyse the CPU utilisation increase caused by the application, which may be translated into power consumption values for simulation purposes. In the simulation, the global CPU energy consumption is then given by the combination of all the individual power contributions of the modelled applications at the current inputs.

To illustrate the approach, we show how this model can be applied to account for the CPU energy consumption of our case study.

B. Application of the model

In our environment, the applications that need to be accounted for their energy consumption are the RWG protocol and the GSF. For the RWG protocol, we can characterise the following states: *RWG*, *RWG_mit_gray* and *RWG_mit_drain*. The first state represents the normal working conditions of the RWG protocol. The individual energy footprint of the protocol running at different transmission rates can be measured as in [4]. The function of the traffic load of the system that provides the power consumption contribute of RWG can be assigned to this state. The other states represent the protocol behaviour when attack mitigation strategies change the way the protocol operates. In the current implementation, RWG can be operated in grey hole or drain attack mitigation. A power consumption function can be assigned to those states with the same logic as before. A separate finite state machine is created for the GSF application, which consists of the intrusion detection, diagnosis and mitigation selection components. Three states, that capture the frequency at which the analysis cycle is performed, are defined: IDS_FAST, IDS_MEDIUM and IDS_SLOW for short, medium and long interval *Ia* used to tune the GSF operation frequency. Again, power consumption functions for each of these states can be extracted from real devices emulating this application under similar working conditions. Finally, the total CPU energy consumption is then given by the sum of the power consumption contribution of the RWG application and the GSF application, as depicted in Figure 3

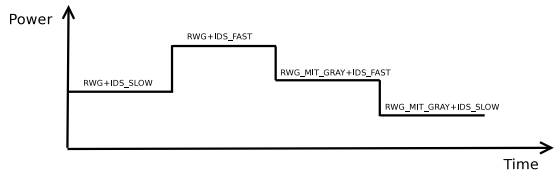


Fig. 3: CPU power consumption of the RWG and IDS framework over some time interval

VI. Evaluation

This section evaluates the implementation of an energy-aware adaptive function applied to the survivability framework for a disaster area network in ns-3. The goal of the evaluation is to show that local adaptation based on per-node estimates of the available energy leads to better overall performance in the network and extends its lifetime.

A. Implementation and simulation setup

As baseline, the scenario presented in [3] has been considered. The network consists of 25 nodes moving in a disaster area network [26]. The load of the network is 15 messages per second sent from randomly chosen nodes. The messages are disseminated in manycast to at least $K = 10$ nodes and have an expiration time of 400 seconds.

The ns-3 energy framework has been included and extended with the proposed CPU energy model to create a model for energy awareness. For simplicity, we have used the energy source model characterised by an ideal linear discharge curve. To consider the energy impact of the wireless device, the WiFi energy model has been employed in the simulation. The current draw values assigned to the different operational states

of the wireless interface are the same as in the work by Wu et al. [16].

The CPU model is as described in section V-A. In order to assign the power consumption values to the states that characterise the RWG application model, the results from Vergara et al. [4] have been considered. To be compliant to the ns-3 energy model, the CPU energy model should specify current draw values in Ampere instead of power in terms of Watts. As the ns-3 source model assumes a constant battery voltage, the conversion between Watts and Ampere is immediate. Assuming that the transmission rate is 15 messages per second, according to the load injected in the network, the energy consumed by the RWG application in the normal operation mode at this rate is 0.025W, as depicted in Figure 7 in Vergara et al. [4]. Furthermore, we consider the additional contribution of 0.1W as the power consumption due to message deletion at the same rate. The power consumption value at a rate of 15 messages per second is then 0.125W. Considering 3.7V constant battery voltage, the current draw that we associated to this state is 0.034A. RWG in mitigation mode usually performs less operations, since the information contained in some signalling packets is discarded or not processed. A lower footprint has been assigned to both of the considered mitigation states. Table II summarises the current draw assigned to RWG.

Rwg application state	**Current draw (A)**
RWG	0.034
RWG_Drain_Mitigation	0.018
RWG_Greyhole_Mitigation	0.018

TABLE II: Current draw of the RWG application

For GSF, as mentioned earlier, the three states modelled correspond to when the detection-diagnosis-mitigation analysis are performed within short, medium or long, intervals, as specified by the parameter I_a. The interval of aggregations I_a chosen for IDS_FAST is 50 sec, IDS_MEDIUM is 75 seconds and IDS_SLOW is set to 100 seconds. The assigned constant energy footprint of the IDS states is shown in Table III.

IDS application state	**Current draw (A)**
IDS_FAST	0.040
IDS_MEDIUM	0.025
IDS_SLOW	0.010

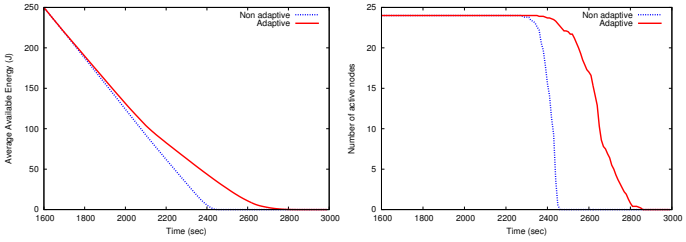
TABLE III: Current draw of the IDS application

In order to illustrate the benefits of adaptation, we focus on two of the possible attack types: the drain attack and the grey hole attack. In the first attack type the malicious nodes act in order to drain the battery of the victims, injecting fake signalling packets that cause benign node to perform a lot of unnecessary disseminations. In the second type of attack, malicious nodes target the message dissemination, by sending fake signalling packets that cause the interruption of the process before the messages have actually been delivered to the intended K nodes. Both attacks are interesting to analyse with regard to the adaptive function, since they are complementary in terms of energy consumption. In the drain attack nodes waste energy as a consequence of the attack, thus good detection accuracy and low latency are necessary to avoid energy waste. On the contrary, the goal of the grey hole attack is that the dissemination of the messages in the network is diminished, thus the nodes should consume less energy due to the reduced amount of network traffic. Longer detection latency in this case could be tolerated, from the energy perspective, but adaptation should still guarantee good detection to ensure network functionalities to be executed.

B. Simulation results

In order to test the complete set of actions performed by the adaptivity function during an entire simulation period, the initial energy level assigned to the nodes is selected to be lower than needed to conclude the simulation over 3000 seconds. In this way, one can determine whether the adaption extends the lifetime of the network compared with the non-adaptive case. In all of the following simulations, five randomly placed malicious nodes start the attack at second 2067 and this lasts until the end of the simulation (details same as [3]). Ten runs of the same simulation are performed, and the results are averaged.

The first attack type we simulated is the drain attack. Figure

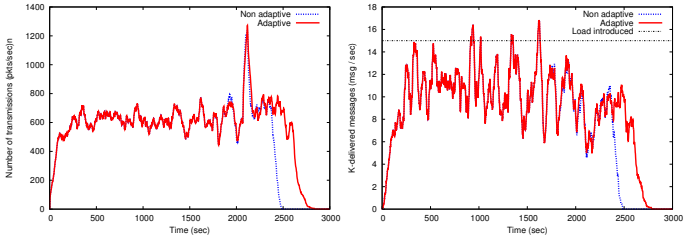


(a) Average energy in the network (b) Number of active nodes

Fig. 4: Energy consumption and number of active nodes in the drain attack

4a shows the comparison of the average available energy in the network in the adaptive case compared to the non-adaptive one when the network is under the drain attack with mitigation enabled. As it can be observed from the figure, the adaptive function outperforms the non-adaptive case by extending the lifetime of the network by over 250 seconds. The difference of the energy level is larger in the second half of the time window, since there the nodes start preserving their battery adopting longer aggregation intervals I_a. The confidence interval of the ten rounds is up to 2% as long as all the nodes have battery capacity, but can drop to 50% when the nodes start to drop out due to depletion. Figure 4b shows the number of active nodes in the network. A good detection accuracy and the minor energy consumption due to the lower CPU utilisation of the adapted IDS application has an impact on the lifetime of the network, although the linear discharge behaviour in both cases is still caused by the energy consumption of the wireless interface in ad hoc mode (recall this mode has a strong impact

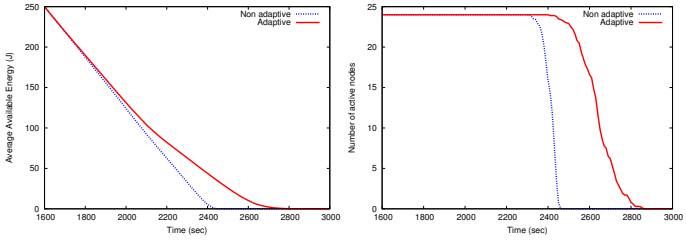
on the consumption due to the constant idle listening [27]). This causes the nodes to have a similar discharge behaviour, which results in a sharp drop of the number of active nodes, as can be observed in Figure 4b.



(a) Number of packets sent (b) k-delivery rate

Fig. 5: Survivability performance in the draining attack

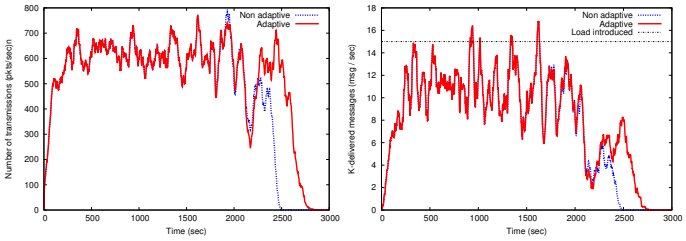
The impact of the adaption on the survivability performance during the same attack is shown in Figure 5. Two metrics are used to measure the network performance: the packet transmission rate and the packet k-delivery rate. The first indicates the number of transmitted packets (including data and signalling packets) during the interval of study, which is useful to analyse the impact of the attack on the bandwidth usage. The second metric shows the performance of the network as number of packets successfully delivered to K nodes over the interval of study. We expect that if adaptation of the interval is successful the results of attack detection measured in terms of network performance are not any worse than when we are not adapting. In Figure 5a, we can see a peak on the number of transmissions at the beginning of the drain attack (at second 2100). This is caused by some detection latency, that leaves some bogus messages being disseminated in the network. This number is slightly higher in the adaptive case, but afterwards the number of messages sent during the attack is similar to the case in which the interval is not adapted, meaning that the adaptation does not decrease the detection accuracy. In Figure 5b, we can see how the delivery ratio is also very similar to the non-adaptive case, indicating that the overall performance is preserved.



(a) Average energy in the network (b) Number of active nodes

Fig. 6: Energy consumption and number of active nodes in the grey hole attack

The results of the adaption on the greyhole attack are presented in Figures 6 and 7. As in the case of the drain

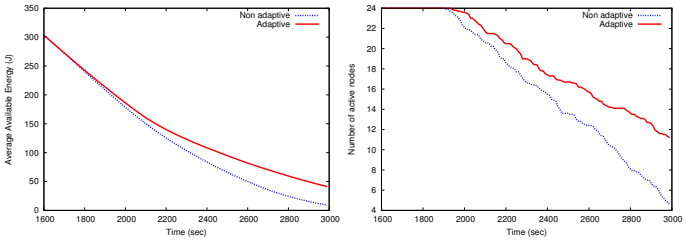


(a) Number of packets sent (b) k-delivery rate

Fig. 7: Survivability performance in the grey hole attack

attack, the average available energy in the network is higher in the adaptive case compared to the non-adaptive one, as shown in Figure 6. However, a more accurate analysis of the survivability performance should be undertaken since in this attack scenario a bad detection could give energy saving. As shown in Figure 7, there is again some latency at the beginning of the attack, in which we can see that a sharp decrease in the number of transmissions is caused by the malicious nodes causing packets to be dropped. After that, however, the number of transmissions is greater in the adaptive case compared to the non-adaptive one, showing that a longer interval of aggregation in this case is a benefit and improves the detection accuracy. Figure 7b in fact shows that the number of deliveries is higher after second 2200.

For a more realistic scenario, we studied the case in which nodes started with random initial battery levels. This test shows how heterogeneity on adaptation and attack response impacts our major metrics: the available energy and the survivability of the network. In this case, all the nodes are assigned a random initial energy between 100J and 500J.



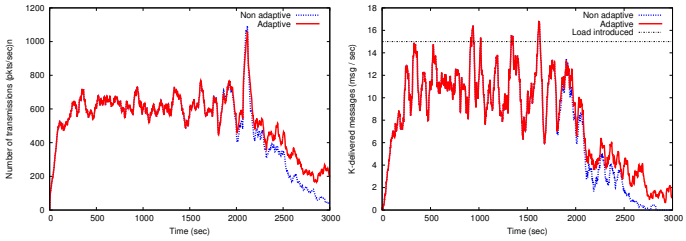
(a) Average energy in the network (b) Number of active nodes

Fig. 8: Energy consumption and number of active nodes in the drain attack with random initial energy levels

Again, Figure 8 shows that adaptation provides energy efficiency while Figure 9 shows that the global impact on the network survivability is further improved.

VII. Conclusion and Future Work

This paper described the design and implementation of an on-line energy-based adaptation component for the attack survivability framework in the context of mobile ad hoc communication. In order to study the network lifetime,



(a) Number of packets sent (b) k-delivery rate

Fig. 9: Survivability performance in the drain attack with random initial energy levels

guaranteeing at the same time a good level of protection, a decision table-based approach has been chosen. The adaptation module selects, in a real-time fashion, the pre-configured IDS parameter set depending on the available energy of the node and the perceived attack situation. This module then adjusts the trade-off between attack response time and energy consumption based on the available energy. Furthermore, in order to enable CPU energy consumption accounting in network simulators, a CPU energy model has been proposed. The adaptation component has been tested in ns-3 simulating the network under two complementary kinds of attack in terms of energy and detection latency impact. The results have shown that the adaptation component gives an extension of about 14% of lifetime without degrading the survivability performance.

Further research must address proactive energy-based approaches, as for example foreseeing the depletion based on the current workload and adapting in advance. A more detailed CPU model should also be included to enable more realistic CPU usage and energy consumption simulation.

Acknowledgements

This work has been supported by the Swedish national Graduate school in Computer science (CUGS) and the Swedish Civil Contingencies Agency (MSB) Information Security program.

References

[1] HFN project, www.ida.liu.se/∼rtslab/HFN, accessed 21 July 2011.

[2] M. Asplund and S. Nadjm-Tehrani, "A partition-tolerant manycast algorithm for disaster area networks," *IEEE Symposium on Reliable Distributed Systems*, pp. 156–165, 2009.

[3] J. Cucurull, M. Asplund, and S. Nadjm-Tehrani, "Anomaly detection and mitigation for disaster area networks," in *Recent Advances in Intrusion Detection*, ser. Lecture Notes in Computer Science, S. Jha, R. Sommer, and C. Kreibich, Eds., vol. 6307. Springer Berlin / Heidelberg, 2010, pp. 339–359.

[4] E. J. Vergara, S. Nadjm-Tehrani, M. Asplund, and U. Zũrutuza, "Resource footprint of a manycast protocol implementation on multiple mob ile platforms," in *Next Generation Mobile Applications, Services and Technologies, 2011 NGMAST '11. The Fifth International Conference on*. IEEE, Sept. 2011.

[5] C. Xenakis, C. Panos, and I. Stavrakakis, "A comparative evaluation of intrusion detection architectures for mobile ad hoc networks," *Computers & Security*, vol. 30, no. 1, pp. 63 – 80, 2011.

[6] H. Otrok, N. Mohammed, L. Wang, M. Debbabi, and P. Bhattacharya, "A game-theoretic intrusion detection model for mobile ad hoc networks," *Comput. Commun.*, vol. 31, pp. 708–721, March 2008.

[7] S. Sen, J. A. Clark, and J. E. Tapiador, "Power-aware intrusion detection in mobile ad hoc networks," in *Ad Hoc Networks*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Springer Berlin Heidelberg, 2010, vol. 28, pp. 224–239.

[8] G. Jacoby and N. Davis, "Mobile host-based intrusion detection and attack identification," *Wireless Communications, IEEE*, vol. 14, no. 4, pp. 53 –60, august 2007.

[9] K. Nadkarni and A. Mishra, "A novel intrusion detection approach for wireless ad hoc networks," in *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE*, vol. 2, march 2004, pp. 831 – 836 Vol.2.

[10] B. Sun, K. Wu, Y. Xiao, and R. Wang, "Integration of mobility and intrusion detection for wireless ad hoc networks: Research articles," *Int. J. Commun. Syst.*, vol. 20, pp. 695–721, June 2007.

[11] J. Flinn and M. Satyanarayanan, "Energy-aware adaptation for mobile applications," in *Proceedings of the seventeenth ACM symposium on Operating systems principles*, ser. SOSP '99. ACM, 1999, pp. 48–63.

[12] ——, "Managing battery lifetime with energy-aware adaptation," *ACM Trans. Comput. Syst.*, vol. 22, pp. 137–179, May 2004.

[13] J. Peddersen and S. Parameswaran, "Energy driven application self-adaptation at run-time," *Journal of Computers*, vol. 3, no. 3, 2008.

[14] A. Ferrante, A. V. Taddeo, M. Sami, F. Mantovani, and J. Fridkins, "Self-adaptive security at application level: a proposal," in *ReCoSoC 2007, Jun. 2007, in proceedings of ReCoSoC 2007*, June 2007.

[15] C. Chigan, L. Li, and Y. Ye, "Resource-aware self-adaptive security provisioning in mobile ad hoc networks," in *Wireless Communications and Networking Conference, 2005 IEEE*, vol. 4, march 2005, pp. 2118 – 2124 Vol. 4.

[16] H. Wu, S. Nabar, and R. Poovendran, "An energy framework for the network simulator 3 (ns-3)," in *4th International ICST Conference on Simulation Tools and Techniques (SIMUTools '11)*, 2011.

[17] F. Chen, I. Dietrich, R. German, and F. Dressler, "An Energy Model for Simulation Studies of Wireless Sensor Networks using OMNeT++," *Praxis der Informationsverarbeitung und Kommunikation (PIK)*, vol. 32, no. 2, pp. 133–138, June 2009.

[18] OMNeT++, http://www.omnetpp.org/, accessed 21 July 2011.

[19] L. M. Feeney and D. Willkomm, "Energy framework: an extensible framework for simulating battery consumption in wireless networks," in *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques*, ser. SIMUTools '10. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2010, pp. 20:1–20:4.

[20] ARMulator, www.arm.com, accessed 21 July 2011.

[21] T. Austin, E. Larson, and D. Ernst, "Simplescalar: An infrastructure for computer system modeling," *Computer*, vol. 35, pp. 59–67, February 2002.

[22] P. Stanley-Marbell and D. Marculescu, "Sunflower: full-system, embedded, microarchitecture evaluation," in *Proceedings of the 2nd international conference on High performance embedded architectures and compilers*, ser. HiPEAC'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 168–182.

[23] L. Palopoli, G. Lipari, G. Lamastra, L. Abeni, G. Bolognini, and P. Ancilotti, "An object-oriented tool for simulating distributed real-time control systems," *Software: Practice and Experience*, vol. 32, no. 9, pp. 907–932, 2002.

[24] E. Weingärtner, F. Schmidt, H. V. Lehn, T. Heer, and K. Wehrle, "Slicetime: a platform for scalable and accurate network emulation," in *Proceedings of the 8th USENIX conference on Networked systems design and implementation*, ser. NSDI'11. Berkeley, CA, USA: USENIX Association, 2011, pp. 19–19.

[25] M. Salehie and L. Tahvildari, "Self-adaptive software: Landscape and research challenges," *ACM Trans. Auton. Adapt. Syst.*, vol. 4, pp. 14:1–14:42, May 2009.

[26] N. Aschenbruck, E. Gerhards-Padilla, M. Gerharz, M. Frank, and P. Martini, "Modelling mobility in disaster area scenarios," in *MSWiM '07: Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems*. ACM, 2007, pp. 4–12.

[27] L. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, 2001, pp. 1548 –1557 vol.3.