Proceedings of the 2011 Winter Simulation Conference
S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, eds.

SIMULATION OF WIRELESS SENSOR NETWORKS UNDER PARTIAL COVERAGE

Ruth Lamprecht
Peter Kemper

College of William and Mary
Department of Computer Science
Wiiliamsburg, VA 23187 USA

ABSTRACT

This paper presents research using simulation to explore the sensitivity of the network lifetime of a wireless sensor network (WSN) under the constraint to maintain a chosen coverage percentage when different aspects of the node model are included. Specifically, we begin with a simple sensor node that can transition between an `Awake` mode and a `Sleep` mode, dependent on meeting the coverage constraint with a simple battery model that expends energy when the node is in the `Awake` mode. We then compare this network behavior to when the battery model includes battery recovery behavior. We conclude that while the difference between the behaviors is small, they are significant enough to warrant the inclusion of a more sophisticated battery model when modeling wireless sensor networks.

1 INTRODUCTION

A wireless sensor network (WSN) is a set of sensor nodes that can be used to monitor some situation. These nodes share data through wireless connections, with the goal being to have the data collected to be analyzed somewhere else. The nodes can be all of the same type, or the network can be a heterogeneous mix of multiple different kinds of nodes. A network can consist of only a few nodes or of several hundred or thousand nodes. A common usage for sensor networks is to do observance of the area to detect activity of some kind. This may be in an environmental capacity to monitor weather and wildfire behavior (Douglas et al. 2006), water quality (Ailamaki et al. 2003), or numerable other natural occurrences. WSNs have also been applied to underwater applications for exploration and data collection (Akyildiz, Pompili, and Melodia 2005). Another use for WSNs is for structural monitoring to detect damage to buildings, bridges and large transports (Xu et al. 2004). Intrusion detection (Arora et al. 2004) is also a common application for WSNs.

Modeling WSNs is a useful practice to decide on the desired set-up of a real system. One can explore parameter setting changes and quickly see the effect on the network behavior and lifetime. But in doing so, a modeler must decide what aspects of the real system to include in the model. Excessive detail can lead to unnecessary work in the analysis. Law and Kelton (2000) provide guidelines for determining the amount of detail to include. The details should directly relate to the specific issues being studied, but more details should be included if they affect the validity or credibility of the model.

The simplest sensor node has two main modes, `Awake` and `Sleep`, a protocol for deciding when to switch between modes, and a battery that determines how long a sensor can work. The simplest battery model is a drainage of power over the time that a sensor is in the `Awake` mode, which can be improved by including battery recovery, which occurs when the node is in the `Sleep` mode, either at a constant rate or at a variable rate dependent on the rest of the state of the node.

This paper explores the sensitivity of the network lifetime of a detection sensor network to the inclusion of more complex information in a node model. The network constraint used is that a coverage percentage, or the number of sensors in the `Awake` mode, must be maintained. The focus here is how including different aspects of the node, specifically relating to the battery, into the model affect the outcome of the simulation. The aspects of interest are including, or not, battery recovery, and simple or more complex rules for such recovery. Model exploration is done with the Möbius program (PERFORM 2011). The sensor node model specifics come from Ren et al. (2005) while the more sophisticated battery model comes from Jongerden et al. (2010).

This paper continues as follows: Section 2 discusses related work and the Möbius modeling tool. Section 3 defines the model for a wireless sensor node and network used in this paper. Section 4 presents the simulation study results and Section 5 provides a brief summary of this work.

2 RELATED WORK

2.1 Modeling Wireless Sensor Networks

Ren et al. (2005) examine the detection probability for different network coverage percentages with varying densities and node activation probabilities. They use their simulation experiments to define and test an analytical model that can be used to test the quality of object detection under different network conditions and scheduling protocols, finding that a random protocol for when a node transitions between the two modes provides the best lifetime. Lu et al. (2009) examine how to maximize network lifetime while maintaining quality of service, as defined by coverage and connectivity. They examine modifying a sensor's transmission and sensing ranges to save energy, under the assumptions that all sensors are connected and any target is within the maximum sensing range of at least one sensor. They find that the more adjustable the ranges are, the longer the network lifetime. Chiasserini and Garetto (2004) propose a sensor network model to examine energy consumption, network capacity and data delivery. They verify their model through both numerical analysis and simulation. One finding from their research is that as the data delivery delay decreases, the energy consumption increases. Cao et al. (2005) propose a protocol for node sleep scheduling in order to maximize network lifetime while satisfying coverage constraints. They also find that decreasing the detection delay also decreases the lifetime, but show that their optimized schedule, where nodes calculate their own wake-up times based on the schedules of neighboring nodes, performs better that either synchronized or random transitions between the two modes. However, none of these consider battery recovery in their models. Niyato, Hossain, and Fallahi (2007) analyze and compare different sleep/wake strategies for solar-powered nodes to optimize node lifetime, with a battery model that allows recovery, but do not consider network behavior. Their assumptions include that energy consumption occurs at a constant rate and the time/energy spent on switching between modes is negligible and therefore can be ignored. Jongerden et al. (2010) consider the parameter settings for maximizing the battery lifetime, instead of network lifetime. They find that by using the recovery effect to gain more power for the battery has a significant effect on the lifetime. They also find that there is a natural bound on the lifetime that no scheduling protocol can surpass.

2.2 Möbius Modeling Tool

The modeling research conducted for this research was done using the Möbius (Deavours et al. 2002) modeling framework. Möbius was developed to integrate several modeling formalisms, as well as multiple solvers. It can support Stochastic Activity Networks (SANs), Performance Evaluation Process Algebras (PEPAs), Stochastic Petri Nets (SPNs), Markov processes and queuing networks, and allows a model to be composed of all the different types. It offers both simulation and numerical analysis of Markov processes. Submodels can be composed into larger models, with the ability to share data between submodels. We expand further on the terminology of Möbius, since this is the framework chosen for the research presented here.

In Möbius, the base unit is an atomic model. As a simple example, we consider a basic two-state node with two state variables, `Awake` and `Sleep`, and two states, μ_A and μ_S, that indicate if the system is up ($\mu_A(\mathtt{A},\mathtt{S}) = (1,0)$) or down ($\mu_S(\mathtt{A},\mathtt{S}) = (0,1)$). There are two events, going to sleep (`ToSleep`) and

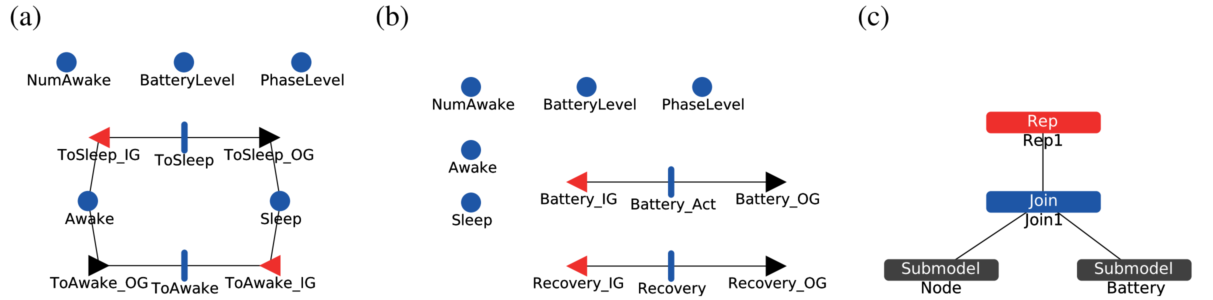

Figure 1: The Möbius atomic model of (a) a single sensor node, and (b) the battery, with (c) the Möbius composed model of multiple nodes under a *Rep/Join* composition.

waking up (`ToAwake`), where `ToSleep` can only occur in state μ_A (the unit is up) and `ToAwake` can only occur in state μ_S (the unit is down). Where these events can occur, we have a sleep transition rate of α_S, a wake transition rate of α_A, with $\alpha_A, \alpha_S > 0$, and the transition value to be the opposite state. That is, in state μ_A, the node transitions to the sleep state with rate α_S, and the new state becomes μ_S.

An atomic model within the SAN formalism used in Möbius is shown in Figure 1(a). In the SAN, the *place* primitive, pictured as a solid circle, represents the state of the model as an integer that can be modified as the state changes. For instance, the place `Awake` represents whether a unit is working (value equals 1) or not (value equals 0), with the opposite values for the place `Sleep`. A third state variable, `NumAwake`, is used by the full system to share the number of nodes, or submodels, in the system that are in the `Awake` state with all nodes in the system. It should be noted that in state μ_A, μ_A(`NumAwake`) has values between 1 and n (where n is the number of submodels in the system) while in state μ_S, μ_S(`NumAwake`) has values between 0 and $n-1$. Places can be of more complex data types such as user-defined structures or arrays in addition to primitive types like integer and floating point. The other places in the model are used in connection with the battery model, which is discussed later.

The transitions between states are controlled by the *activity* primitive, shown as a thick vertical line. Transition rates can be the rate of an exponential distribution that models the time the node stays in a state, but Möbius supports a variety of distributions that a modeler can chose from to describe time delays. Another SAN primitive, *output gates*, depicted by the right arrows, allows the modeler to also specify complex state changes, described through C++ code. In the example above, the transition from state `Awake` to `Sleep` performs a decrement of `Awake`, increment of `Sleep`, and decrement of `NumAwake`. All submodels in the full system of the same type will share the place `NumAwake` since they will all contribute to how many of the submodels are in the `Awake` state. These places can also be used to calculate measures of interest (e.g., how often and for how long a submodel is in the sleep mode). The final SAN primitive, *input gates*, depicted by the left arrows, allows conditions to be set for when an activity can fire. In this model, they are used to condition the transitions to occur such that the desired coverage percentage is preserved, as well as considering the state of the battery, as discussed later.

For models with a large number of places and transitions, it is useful to apply some structuring mechanism or composition operations for SANs. Here, we use a composition operation to construct a larger model with n submodels. Figure 1(c) shows how we make use of the Möbius *Rep/Join* composition, which provides a composition operator, *Rep*, to instantiate n copies of the submodel to build an aggregate model. The other composition operator, *Join*, connects two submodels without replication. All instances can share a user-given set of state variables/places, (e.g., `NumAwake`). We could then build this into a larger model through the use of another *Rep* or *Join* operator. Alternative composition methods are so called action synchronization and graph composition, see (Deavours et al. 2002) for details.

In Möbius, measures of interest (such as the probability of the submodel being in the `Awake` state) are defined through *reward* variables. We can define a performance variable, `numAwakeProb`, to return the value of the place `NumAwake` at either a specific moment in time (transient analysis) or on average (steady state analysis). Once the model is fully defined through the atomic model, composed model, and reward measurements, Möbius allows a modeler to solve for the reward measurements either through numerical analysis, with the accompanying state space generation, or discrete-event simulation. More information about all of these procedures and formalisms can be found in (Deavours et al. 2002) or through the program website (PERFORM 2011).

3 MODEL DESCRIPTION

The goal of this study is to examine the influence of the inclusion of various aspects of the battery into the model on the overall system behavior. We specifically consider the constraint of network coverage and the performance measure of network lifetime. We use a simple sensor node with the two modes as previously described, modeled in a network of 100 nodes.

3.1 Sensor Node

For this research, we expand the node introduced in the previous section to include the battery model to follow what is used in Ren et al. (2005). For the experiments done here, a node has a **period** for its `Awake`-`Sleep` cycle with an **active_ratio** for the fraction of the **period** spent in the `Awake` mode. Thus, the transition rate from the `Awake` mode is **period** $*$ **active_ratio**, which is modeled with an exponential distribution with a mean of the rate inverse and the transition rate from the `Sleep` mode is **period** $*$ **(1 - active_ratio)**, again modeled with an exponential distribution with a mean of the inverse of the rate. The exponential distribution is chosen because it guarantees that the rates will be positive and it allows numerical analysis to be done on the associated Continuous Time Markov Chain, if desired. The exponential distribution is a simple non-negative distribution In addition to modeling the transitions from a `Sleep` mode to an `Awake` mode, we also want to model the battery level. The battery model is shown in Figure 1(b). The *place* `BatteryLevel` holds the battery level, which is depleted only when the node is in the `Awake` mode (controlled by the *input gate* `Battery_IG`) at a rate of **energy_rate** (controlled by the *activity* `Battery_Act`). This means that given an initial battery level of **max_battery**, if the node never goes to sleep, the battery will be depleted in **max_battery / energy_rate** time units. This time is extended by the amount of time the node spends in the sleep mode. The *output gate* `Battery_OG` ensures that when the battery is depleted, the node automatically goes to the `Sleep` mode. In the node model, the transition from `Sleep` to `Awake` is conditioned first on the battery level being greater than 0 (controlled by the *input gate* `ToAwake_IG`), so a fully depleted battery becomes an absorbing state. Once all nodes have reached this state, the system is in a final, failed state. The transition from `Awake` to `Sleep` is conditioned on the requirement that at least the coverage percentage of nodes remains awake (controlled by the *input gate* `ToSleep_IG`). That is, if the coverage requirement is not met, then the node remains awake for another time period (as defined by the transition rate). Similarly, the transition from `Sleep` to `Awake` is further conditioned on not needing more than the coverage percentage of nodes awake (controlled by the *input gate* `ToAwake_IG`). That is, if the coverage requirement is met, the node will remain in the sleep mode. The *place* `PhaseLevel` tracks the phase of the node, which relates to the time the node spends in the `Awake` mode.

Jongerden et al. (2010) show that when modeling batteries, including the *recovery effect*, or the phenomenon that batteries recharge when not in use or when in low usage, affects the lifetime of a battery. According to Jongerden et al. (2010), the simplest recovery model is to allow a constant recovery effect, but further detail can be added to reflect the fact that the weaker the battery, the less recovery occurs. In the model used here, the recovery activity and gates control battery recovery, which can only occur in the sleep mode if the battery has not been fully depleted. That is, the *input gate* `Recovery_IG` puts the condition on the `Recovery` activity so the activity is only enabled when `Sleep` equals 1 and `BatteryLevel` is not zero, since power is needed to make power, and `BatteryLevel` is less than the maximum charge, since this is the limit on the amount of power that the battery can hold. The recovery rate is also an

Table 1: Average network lifetime for nodes with no battery recovery, simple battery recovery, and phase-based battery recovery for increasing required coverage percentages.

Cov. %	No	Simple	Phase
60	64.22	65.58	76.76
70	58.93	60.02	68.36
80	54.27	54.73	60.76
90	50.43	50.52	54.04
100	46.08	45.92	46.49

exponential distribution with either the constant mean or with the state-dependent value. Both the phase level and battery recovery aspects of the model are used in later experiments.

Since the node and the battery models are defined in separate atomic models, to form a complete node we must join the two parts of the unit. In connecting a node model and a battery model through the *Join* operator, all places in each model are shared since this is information both parts of the unit need to access and change. For instance, when the battery activity fires, the `BatteryLevel` must be decremented, but if that puts the level at 0, the battery has died, so the node must be put to sleep and the total number of nodes that are awake must also be decremented, so the places `Awake`, `NumAwake`, and `Sleep` are all given new values.

3.2 Sensor Network

The simplest way to specify the network is with the *Rep/Join* operator as in Figure 1(c). The *Join* operator connects the two parts of the node as described above. The *Rep* operator then holds the information on the number instances of sensor nodes. For an initial state of the network, we have chosen to compare when all nodes start in the `Awake` mode, all nodes start in the `Sleep` mode, or having exactly the coverage percentage of the nodes start in the `Awake` mode. This simple model ignores spatial aspects, ignores communication and assumes that the only dependence across nodes is the current level of coverage.

4 RESULTS

The following results are calculated through simulation, run 1000 times to gain a 95% confidence interval. The maximum battery level for each node set at 30 with an **energy_rate** of 1, while the **period** is 1.1 with an **active_ratio** of 0.5.

4.1 Coverage Percentage under Different Initial Conditions

Figure 2 show the network lifetime for the various starting conditions. First, Figure 2(a) shows results from setting all nodes to start in the `Awake` mode, while Figure 2(b) comes from experiments where all nodes start in the `Sleep` mode. We also tested the initial condition where some nodes start in the `Awake` and the rest in the `Sleep` mode, based on the desired coverage percentage. In the first two cases, it can be seen that the network lifetime is the same regardless of the initial condition of the nodes in the network. The results for the mixed starting mode are not included here to reduce the redundant figures. Since the initial state of the network does not affect the overall network affect, for all future simulations, we use the initial state of all nodes being in the `Awake` mode. In Table 1, the first column lists the expected value for the lifetime of the network for the chosen coverage percentages with no battery recovery. We use these as base values for comparison to the models that include more information.

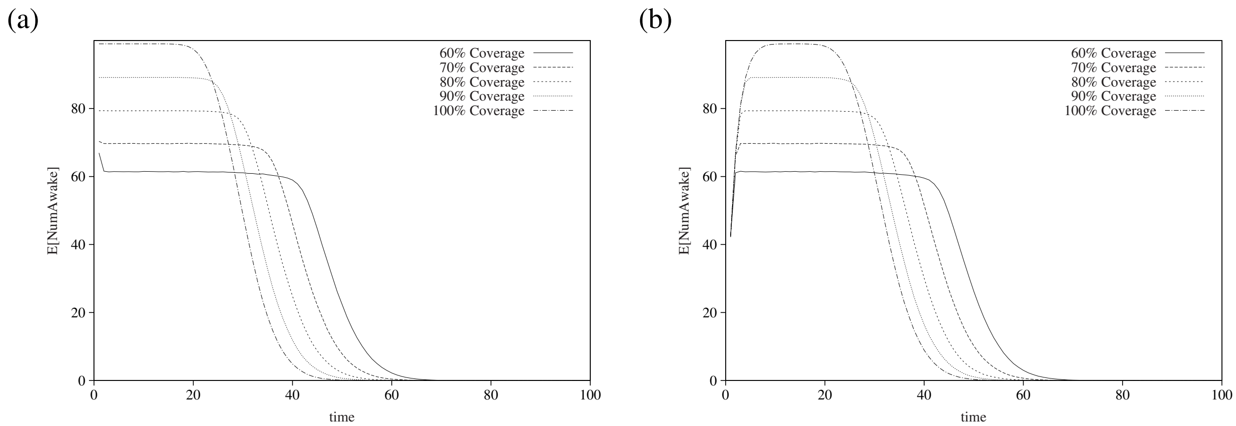

Figure 2: Expected number of nodes in the `Awake` mode given the starting condition that all nodes start in (a) the `Awake` mode or (b) the `Sleep` mode.

4.2 Coverage Percentage with Battery Recovery

Figure 3(a) shows the results for experiments on a network of nodes that incorporate battery recovery, measured by the expected number of nodes in the `Awake` mode. The results shown are for the simplest recovery effect, which shows a slight increase in the network lifetime. The second column of Table 1 lists the expected network lifetime for the chosen coverage percentages with this simple recovery. The increase in the network lifetime is less than 2% over the model with no recovery, so the inclusion of a simple battery recovery model does not change the outcome of the network study. One point of interest that can be seen in these results, and all others, is that with a higher required coverage percentage, more nodes will spend all or nearly all of their time in the `Awake` mode, so there is little or no recovery and the lifetime of the network is dependent on the energy level of a node.

Figure 3(b) shows the results for experiments that incorporate a more realistic battery recovery. As before, battery recovery occurs only when the node is in the `Sleep` mode, but the amount of recovery is dependent on the phase of the node p, and the current battery level, b. The phase of a node is defined by the amount of time spent in the `Awake` mode, as measured by the energy spent by the battery. Thus, as the battery is depleted, the phase increases. When the battery is nearly full, the phase level will be low. The longer a node remains in the `Awake` mode, the more the battery is depleted and the higher the phase level. Once the node goes to the `Sleep` mode, the phase and battery levels determine the recovery amount, r, as defined by (Chiasserini and Rao 2001):

$$r = e^{g_N(N-b)-g_C(p)}$$

where N is the maximum battery level and g_N and g_C are parameters dependent on the recovery properties of the battery. The expected lifetime of the network for the chosen percentages are listed in the third column of Table 1. For coverage percentage of 80, there is a 13% change in the calculated network lifetime, which shows that this simple inclusion can have a large impact on the study results.

4.3 Coverage Percentage with Battery Recovery and Modified Sleep Schedule

Modeling with the more refined battery model introduces the desire to schedule the transition to the `Sleep` mode based on the time spent in the `Awake` mode. That is, the longer a node stays awake, the more likely it is to transition. Therefore, we condition the `Awake-Sleep` transition on either the coverage percentage being met (as what has been done for the above results) or by the phase level reaching a specified maximum value. Given a maximum battery level of 30, we examine the network behavior for a maximum phase level of 5, 10, 15, and 20. Higher phase level results are not included here because when the maximum phase

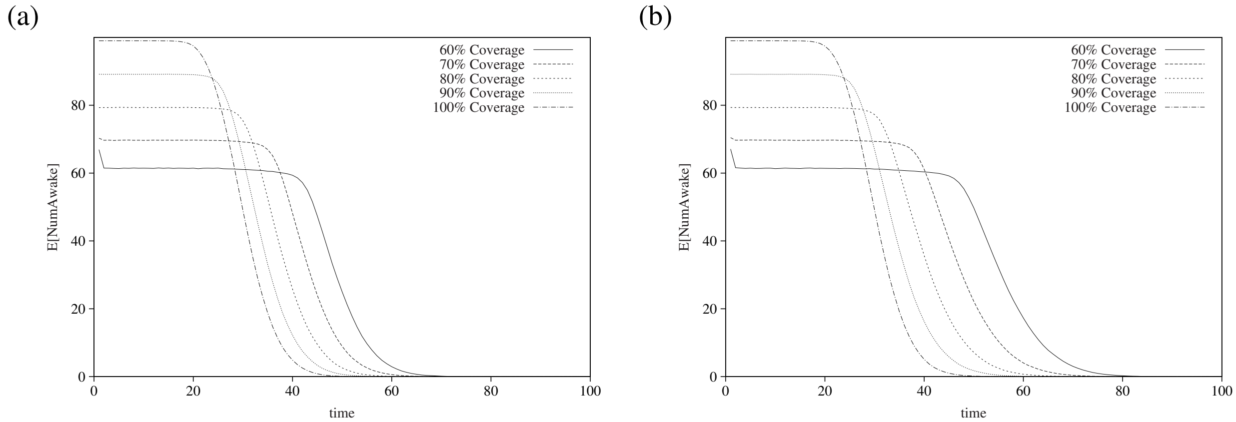

Figure 3: Expected number of nodes in the `Awake` mode given the starting condition that all nodes start in the `Awake` mode and allowing battery recovery during the `Sleep` mode (a) at a constant rate (simple recovery) or (b) at a rate dependent on the phase and battery level.

Table 2: Average network lifetime for nodes with phase-based battery recovery and phase-driven sleep schedule for various maximum phase levels and increasing required coverage percentages.

Cov. %	Max = 5	Max = 10	Max = 15	Max = 20
60	93.55	87.72	85.33	84.83
70	82.59	75.92	74.29	73.31
80	72.31	65.76	64.22	63.69
90	68.85	57.63	55.66	55.94
100	68.05	55.11	50.29	49.51

level nears the maximum battery level, the network lifetime values are within 2% of each other. Figure 4 shows the network behavior through the expected number of awake nodes under the chosen coverage percentages, while Table 2 lists the expected network lifetime for the other four maximum phase levels examined.. Maintaining a low phase-level, or shortening the amount of time spent in the `Awake` mode, which allows for greater battery recovery, increases the expected network lifetime by nearly 40% over the simple recovery model. This is because with a low maximum phase, more state switching is required, which means more time is spent in the `Sleep` mode, which means more recovery time. So, we find again, that a simple inclusion of additional information in the battery model can have a large impact on the study results. We also find that with a high coverage percentage ($> 90\%$), nodes will quickly reach maximum phase level, and so we see in the results a quick drop in the expected number of nodes in the `Awake` mode. However, the system then tries to increase the number of `Awake` nodes to meet the percentage requirements, so we see a brief rise in this number, until nodes begin to lose all power, the recovery amount becomes smaller, and the network descends to a failed state.

4.4 Coverage Percentage with Greater Variance

Figure 5 shows the results for experiments on modifying the variance of the transitions between the `Awake` and `Sleep` modes on the expected number of nodes in the `Awake` mode. High variance in a sensor network becomes interesting when there are unreliable links between the sensors (Halgamuge et al. 2009). The variance for the exponential distribution is 3.3, with a mean of 1.8181. To increase the variance in the transition rates between the two modes, a hyper-exponential distribution was used to increase the variance by factors of approximately 100 and 1000, which leads to a high index of dispersion. For both variance

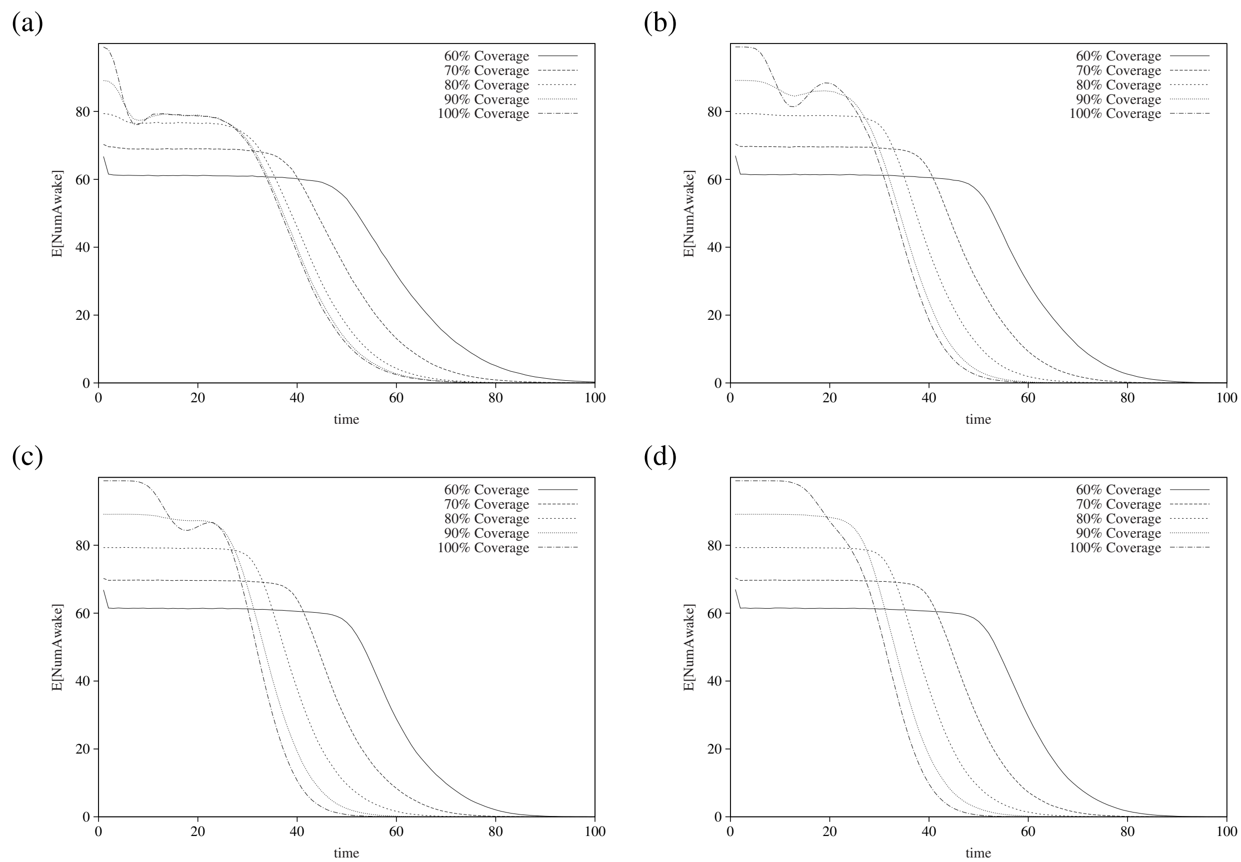

Figure 4: Expected number of nodes in the `Awake` mode given the starting condition that all nodes start in the `Awake` mode, allowing battery recovery during the `Sleep` mode, and scheduling sleep when the phase level reaches (a) 5 or (b) 10 or (c) 15 or (d) 20).

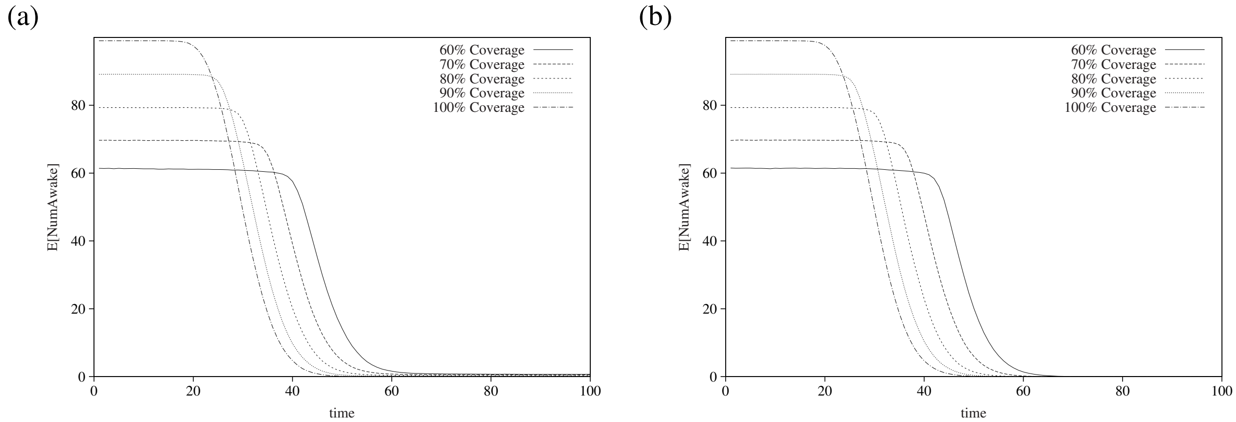

Figure 5: Expected number of nodes in the `Awake` mode given the starting condition that all nodes start in the `Awake` mode and allowing greater variance, (a) times 100 and (b) times 1000, on the mean time to sleep.

changes, the behavior is similar to the original experiments. This suggests that even under extreme variance, the network is stable in its behavior.

5 CONCLUSIONS

The objective of this paper was to explore the sensitivity of the network lifetime of a wireless sensor network to the inclusion of different modeling aspects. By combining the different parts of a WSN, and comparing the expected network lifetime, we conclude that simple model behaviors can be included and have a large impact on the network lifetime. Specifically, when modeling a WSN, while a common assumption is to use a simple battery model that has only energy depletion, the inclusion of battery recovery can lead to as much as a 40% increase in the expected network lifetime. Another observation we have made is that there seems to be two phases to the network behavior. The first phase is while the required coverage percentage can be met and the second phase is the time while the percentage cannot be met, but not all nodes have completely lost power. This is where the impact of the different kinds of models can be seen best. The length of the first phase is dependent on both the coverage percentage and the recovery that can occur while the second phase is dependent on the amount of energy the nodes have left once some nodes begin to die, which is also dependent on the ability of the nodes to have energy recovery during the first phase.

REFERENCES

Ailamaki, A., C. Faloutos, P. S. Fischbeck, M. J. Small, and J. VanBriesen. 2003, December. "An environmental sensor network to determine drinking water quality and security". *SIGMOD Record* 32:47–52.

Akyildiz, I. F., D. Pompili, and T. Melodia. 2005. "Underwater acoustic sensor networks: research challenges". *Ad Hoc Networks* 3 (3): 257 – 279.

Arora, A., P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Demirbas, M. Gouda, Y. Choi, T. Herman, S. Kulkarni, U. Arumugam, M. Nesterenko, A. Vora, and M. Miyashita. 2004. "A line in the sand: a wireless sensor network for target detection, classification, and tracking". *Computer Networks* 46 (5): 605 – 634. Military Communications Systems and Technologies.

Cao, Q., T. Abdelzaher, T. He, and J. Stankovic. 2005. "Towards optimal sleep scheduling in sensor networks for rare-event detection". In *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*, IPSN '05: IEEE Press.

Chiasserini, C.-F., and M. Garetto. 2004. "Modeling the performance of wireless sensor networks". *IEEE INFOCOM* 1:231–242.

Chiasserini, C.-F., and R. R. Rao. 2001, July. "Energy efficient battery management". *IEEE Journal on Selected Areas in Communications* 19 (7): 1235 –1245.

Deavours, D. D., G. Clark, T. Courtney, D. Daly, S. Derisavi, J. M. Doyle, W. H. Sanders, and P. G. Webster. 2002. "The Möbius framework and its implementation". *IEEE Transactions on Software Engineering* 28 (10): 956–969.

Douglas, C. C., R. A. Lodder, R. E. Ewing, Y. Efendiev, G. Qin, J. Coen, M. Kritz, J. D. Beezley, J. Mandel, M. Iskandarani, A. Vodacek, and G. Haase. 2006, December. "DDDAS approaches to wildland fire modeling and contaminant tracking". In *Proceedings of the 2006 Winter Simulation Conference*, edited by L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, WSC '06, 2117–2124. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Halgamuge, M. N., T.-K. Chan, and P. Mendis. 2009. "Experimental study of link quality distribution in sensor network deployment for building environment". *EJSE Special Issue: Sensor Network on Building Monitoring: from theory to real application*.

Jongerden, M., A. Mereacre, H. Bohnenkamp, B. Haverkort, and J.-P. Katoen. 2010, August. "Computing optimal schedules of battery usage in embedded systems". *IEEE Transactions on Industrial Informatics* 6 (3): 276–286.

Law, A. M., and W. D. Kelton. 2000. *Simulation Modeling and Analysis*. McGraw-Hill Higher Education.

Lu, M., J. Wu, M. Cardei, and M. Li. 2009, April. "Energy-efficient connected coverage of discrete targets in wireless sensor networks". *International Journal of Ad Hoc and Ubiquitous Computing* 4:137–147.

Niyato, D., E. Hossain, and A. Fallahi. 2007, February. "Sleep and wakeup strategies in solar-powered wireless sensor/mesh networks: performance analysis and optimization". *IEEE Transactions on Mobile Computing* 6:221–236.

PERFORM 2011. "The Möbius tool". www.mobius.uiuc.edu.

Ren, S., Q. Li, H. Wang, X. Chen, and X. Zhang. 2005. "Analyzing object detection quality under probabilistic coverage in sensor networks". In *Quality of Service IWQoS 2005*, edited by H. de Meer and N. Bhatti, Volume 3552 of *Lecture Notes in Computer Science*, 107–122. Springer Berlin / Heidelberg.

Xu, N., S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin. 2004. "A wireless sensor network for structural monitoring". In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, SenSys '04, 13–24.

AUTHOR BIOGRAPHIES

RUTH LAMPRECHT is an Ph.D. candidate in Computer Science at the College of William and Mary and an instructor at Virginia Union University in the Department of Computer Information Science / Computer Science. Her research interests include simulation, numerical analysis, and optimization. Her web page can be found via http://www.cs.wm.edu/∼rlampy and her email address is rlampy@cs.wm.edu.

PETER KEMPER is an Associate Professor in the Department of Computer Science at the College of William and Mary (previously TU Dortmund and TU Dresden, Germany). His research interests include modeling techniques and tools for performance, performability and dependability analysis of systems. He contributed to analysis techniques for the numerical analysis of Markov chains, model checking stochastic models, techniques for simulation optimization. His web page can be found via http://www.cs.wm.edu/∼kemper and his email address is kemper@cs.wm.edu.