Towards the Simulation of Energy-Efficient Resilience Management

Cathryn Peoples, Gerard Parr
Faculty of Computing and Engineering
University of Ulster
Coleraine, Northern Ireland, United Kingdom
{c.peoples, gp.parr}@ulster.ac.uk

Alberto Schaeffer-Filho, Andreas Mauthe
School of Computing and Communications
Lancaster University
Lancaster, United Kingdom
{asf, andreas}@comp.lancs.ac.uk

ABSTRACT

Energy-awareness and resilience are becoming increasingly important in network research. So far, they have been mainly considered independently from each other, but it has become clear that there are important interdependencies. Resilience should be achieved in a manner which is energy-efficient, and energy-efficiency objectives should respect the networks' need to be prepared to observe and react against disruptive activity. Meeting these complementary and sometimes conflicting research objectives demands novel strategies to support energy-efficient resilience management. However, the effective evaluation of cross-cutting energy and resilience management aspects is difficult to achieve using the tool support currently available. In this paper, we explore a range of network simulation environments and assess their ability to meet our energy and resilience modelling objectives as a function of their technical capabilities. Furthermore, ways in which these tools can be extended based on previous related implementations are also considered.

Categories and Subject Descriptors

I.6.7 [**Simulation and Modeling**]: Simulation Support Systems; C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design

General Terms

Design, Experimentation

Keywords

Network simulation, policy, management, adaptation, energy-awareness, resilience, anomaly detection

1. INTRODUCTION

Resilience and energy-awareness are becoming increasingly important as next generation network management objectives, extending the more traditional International Organisation for Standardisation (ISO) Fault, Accounting, Configuration, Performance, and Security (FCAPS) requirements [6]. While each is important in its own right, they also impact on each other and must therefore be considered together. Network resilience describes the network's ability to maintain acceptable levels of service in situations where there are challenges to its normal operation [18]. Challenges may arise from operational problems, such as malicious attacks, natural disasters, or simply insufficient network capacity. Operational resilience should be achieved in a manner which is energy-efficient: when monitoring packet flows with the objective of identifying rogue users, for example, this procedure should be executed such that resource use and subsequently energy-efficiency is optimised. Conversely, in attempting to improve the energy-efficiency of network communications, resilience strategies may be seen as over-cautious monitoring exercises which may be turned off. There is therefore an important combined effort through the objectives of achieving resilience in an energy-efficient manner (and vice-versa) [19]. Meeting these complementary and sometimes conflicting research objectives demands novel strategies to support energy-efficient resilience management. However, it is not straightforward to evaluate the cross-cutting aspects of joint energy and resilience management strategies.

A range of strategies exist to evaluate the performance of network protocol solutions. Incorporation into a live network is an option that is generally not exploited due to its high costs, particularly at the earlier stages of experimentation. Constructing real testbeds for evaluating network operation is another option. However, this is expensive and requires significant time and effort. Network simulators permit the testing of network scenarios and protocols in a comparatively inexpensive manner. We are interested in simulating resilience and energy-efficiency strategies, which comprise a set of mechanisms whose behaviour can be adapted during run-time. For our research purposes this includes the exploration of the ability to achieve resilient and energy-aware network communications through the application of intelligent techniques to the protocol stack, and involves consideration of cross-layer communication and operational optimisation in relation to context attributes collected.

The investigation in this paper therefore involves identifying suitable environments to allow the modelling of resilience and energy-efficiency aspects of network communications. This is performed to enable the design and construction of research prototypes in optimal off-the-shelf software environments and allow evaluation of their impact on perfor-

SIMUTOOLS 2011, March 21-25, Barcelona, Spain

DOI 10.4108/icst.simutools.2011.245564

mance. In performing this investigation, it is acknowledged that there are modelling limitations associated with simulation software. However, it is demonstrated that those environments that are most extensible represent options that can best meet the requirements. The remainder of the paper is structured as follows: Section 2 discusses modelling capabilities of a selection of popular network simulators, and our functional energy and resilience research requirements are defined. Section 3 describes how these environments can be extended to achieve energy and resilience objectives. Section 4 discusses implementation examples from the literature in a range of software environments which have relevance for our research. Section 5 presents the concluding remarks.

2. NETWORK SIMULATORS: A CAPABILITIES AUDIT

In order to model energy-efficiency and resilience aspects we considered the most popular network simulators, including OMNeT++ [12], NS-2 [10], NS-3 [11], SSFNet [17] and Opnet [13]. These are briefly discussed in the following sections.

2.1 OMNeT++

OMNeT++ uses C++ to model network behaviour and NED (Network Description Language) to describe the network topology [20]. OMNeT++ has a modular, extensible, component-based architecture, in which multiple simple modules, e.g., protocols, can be combined into a compound module, e.g., host node. A number of simulation models are available, and of particular interest for our research is the INET framework, which includes a selection of protocols at the range of layers. At the transport layer, the protocols RTP, SCTP, TCP, and UDP are supported; the link layer supports Ethernet, 802.11, point-to-point, and radio links, and protocols supported at the network layer include ICMP, IPv4, IPv6, MPLS, and OSPF. OMNeT++ is reported to have a good overall performance [21].

2.2 NS-2

NS-2 uses C++ to implement node behaviour and OTcl to define and control each simulation scenario. The main source code in NS-2 is divided into modules, from routing to QoS scheduling, multicast capability, and queuing schemes. Protocol functions at each stack layer are represented in a similar modular approach. NS-2 has an extensive library of publicly available models, but fewer tools and infrastructure components such as support for hierarchical models. Performance results can be collected from NS trace files and post-processed to extract results. Collected attributes include times of packet sends and receives, and the node identification number from which the packet has been sent from and received at. However, NS-2 is reported to have poor scalability and large memory footprint when compared to other simulators [21].

2.3 NS-3

Simulations in NS-3 can be implemented in pure C++, or optionally combined with Python scripts. NS-3 is a major revision of NS-2, targeted for scalability, extensibility, modularity, emulation and clarity of design, and focusing on layers 2-4 of the protocol stack [5]. NS-3 currently lacks the extensive library of network models available for NS-2. One of the goals of NS-3 is to develop simulation tools that can be easily integrated with virtual machines, network testbeds and actual implementation code [5]. NS-3 has the ability to map standard APIs into the model, e.g., POSIX, enabling the emulation of real devices and applications. Compared to other simulators, NS-3 is reported to present good scalability and performance [21].

2.4 SSFNet

SSFNet is a standard for discrete-event simulation with implementations in C++ and Java. SSFNet can be used for modelling and simulation of Internet protocols at and above layer 3. Protocols are composed hierarchically to define network components, e.g., nodes and routers. SSFNet uses DML for topology description and configurations, which is a text-based format. DML is equivalent to OMNeT++'s NED, however, according to [20], DML lacks the same expressiveness and features to support large-scale models built from reusable components. In [9], the C++ implementation of SSFNet is reported to present almost the same speed when compared to NS-2, while using less memory. However, development of the SSFNet simulation framework and models was discontinued in 2004.

2.5 Opnet

Opnet is a commercial tool and the source code of the simulation kernel is not publicly available. Opnet is based on C, and it has an extensive implementation of protocols and applications, e.g., FTP, HTTP, print, remote login, and video models. It contains both generic node models e.g., switches and gateways, as well as vendor-specific node models, e.g., Cisco and 3Com routers. Opnet allows hierarchical modelling as in OMNeT++. Topologies are stored in binary format (in contrast with a text-based format as in SSFNet or OMNeT++). A process model, such as an algorithm or protocol, must be described as a state machine. If only built-in standard models are reused, Opnet's user-interface can be very intuitive. On the other hand, if new features have to be built, this representation can be difficult to abstract as well as being hard to debug, extend and validate [1].

2.6 Technical and Functional Requirements

Extension of a network simulator for the modelling of resilience and energy purposes places a number of requirements on the simulation environment:

- ***Library of network models***: the availability of a wide range of network models will permit the experimentation of energy-efficiency and resilience strategies at different network types, including ability to model hardware, infrastructure, and protocols in core, access, metro, and edge wired, mobile and satellite networks.
- ***Emulation ability***: will allow richer and more accurate simulations that can be built using off-the-shelf detection and remediation mechanisms for the evaluation of resilience and energy strategies, by plugging the simulator into live networks.
- ***Extensibility and generality***: we are interested in experimenting with resilience and energy-efficiency in layers 1-7 of the protocol stack. Thus simulators must be general and allow the modelling of communication networks, distributed systems and P2P networks.

Table 1: Comparison of Network Simulator Capabilities

Characteristic	NS-2	NS-3	OPNET	OMNeT++	SSFNet
Open source code	√	√	×	√	√
Emulation ability	√	√	√	√	×
Performance, scalability	×	√	√	√	×
Extensibility and generality	×	√	√	√	√
Language	C++/OTcl	C++/Python	C	C++	C++/Java

- ***Performance analysis***: the selected modelling environment should provide ability to assess application QoE on an end-to-end basis and evaluate the impact on key QoS attributes, including throughput, goodput, and node energy levels.

Table 1 compares the capabilities of the network simulators, in terms of more general technical and functional requirements. It indicates that, among the options that are open source and thus provide more opportunities for extension, NS-3 and OMNeT++ are equally suitable. These will permit the modelling of different network types using a range of application models. The next section will describe how these environments can be extended to achieve energy and resilience modelling objectives.

3. ENERGY-EFFICIENT RESILIENCE MANAGEMENT

Simulation of resilience and energy aspects have been addressed in the past in an independent fashion. However, solutions should be optimised in relation to each other, and a platform for the evaluation of resilience and energy requirements simultaneously is needed. Network resilience must be achieved in a manner which is energy-efficient, and at the same time, strategies for saving energy must not compromise the network's resilience properties. Energy and resilience are therefore cross-cutting aspects of network management. These aspects should be managed in relation to each other as strategies for achieving network resilience will lead to different energy costs, and energy-efficiency techniques may affect the ability to detect and remediate anomalies on the network (Figure 1).

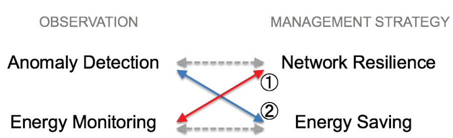

Figure 1: Cross-cutting aspects of energy-efficient resilience management

Resilience strategies will impact on energy usage in the network. This is the case when mechanisms such as traffic classifiers and packet filters are activated locally or at multiple locations of the network. Similarly, mechanisms for anomaly detection, including packet monitoring, filtering and sampling also impact on energy usage and, to improve efficiency of the network, these may be temporarily disabled or turned off. Effective management of these aspects in parallel is therefore an important research objective.

In this section, use and extension of simulation tools to achieve a common platform for *energy-efficient resilience management* are discussed. Our initial focus involves observing and understanding energy usage in the network as a function of the resilience mechanisms dynamically deployed (step 1 in Figure 1). As part of the ongoing research, it will be evaluated how energy-saving strategies can drive the selection and operation of resilience mechanisms without compromising the resilience properties (step 2 in Figure 1).

3.1 Energy Monitoring

Network simulators provide a range of energy-associated capabilities, which can be used to determine the overall energy consumption in simulated models:

- ***NS-2:*** energy model attributes are initialised in the `$ns node-config` definition (Figure 2). They are used to determine operational performance and network state during a wireless transmission. The power-saving mode of operation may be applied to nodes to adapt power consumption as a function of state changes. Wireless signal power is calculated using transmitter power, antenna gain, wavelength, system loss, and propagation distance between communicating nodes.

```
$ns node-config
...
-energyModel $opt(energymodel) \
-idlePower 1.0 \
-rxPower 1.0 \
-txPower 2.0 \
-sleepPower 0.001 \
-transitionPower 0.2 \
-transitionTime 0.005 \
-initialEnergy $opt(initialenergy)
```

Figure 2: $ns node-config definition

- ***NS-3:*** provides scripts relevant to node energy consumption. Within `YansWifiPhy.cc`, for example, the energy detection threshold is defined, representing the minimum acceptable wireless signal strength to allow detection at the Physical layer. Other attributes initialised within `YansWifiPhy.cc` include transmission gain, reception gain and the number of transmission power levels available.

- ***Opnet:*** supports a diverse range of physical layer modelling techniques, including wired, mobile, WiMAX and satellite links. Relevant to all wireless modules is the `ReceivedPower` attribute, which represents the average power of packets arriving at a receiver channel. This attribute is important for the ability to detect a wireless signal, which must remain above a pre-defined threshold to be detectable by the Physical layer. Also relevant for energy purposes are Opnet's models for radio receiver and transmitter gain.

- ***OMNeT++:*** its energy framework supports battery power modelling in wireless 802.11 networks, with the potential to model multiple sources of energy consumption at each node. Every time a packet is processed at the node, the battery capacity is updated by recalculating residual node capacity (Figure 3). This allows node capacity levels to be monitored and the battery consumption scheme can be adapted as required, according to node specific consumption rates. `BatteryStats.cc` is responsible for collecting and presenting statistical data concerning device batteries.

```
energy=devices[i].draw*voltage*(now-
    lastUpdateTime);
devices[i].accts[currentActivity]+=energy;
devices[i].times[currentActivity]+=(now-
    lastUpdateTime);
residualCapacity-=energy;
```

Figure 3: Calculating residual node energy

Such energy-associated capabilities in the simulation environments demonstrate a range of implementation options for this research. However, extensions to the default capabilities are also necessary given the fact that current energy capabilities are included for wireless network portions only. Such extensions could also include, for example, the measurement of carbon emissions from operation of wired networks.

Based on the energy-related information obtained from the simulation environment, a common platform is being developed to allow the evaluation of energy-efficient resilience management. This platform, based on a previously-developed Context-Aware Broker (CAB) algorithm [14] originally built in NS-2, includes the integration of energy-associated context attributes. The CAB algorithm is intended to demonstrate the ability to improve transmission sustainability and reliability in resource-constrained Delay-Tolerant Networks (DTNs). CAB stores information on the overall scenario, including the number and location of nodes, and Management Information Base (MIB) data is populated within the simulation environment (Figure 4). Based on the MIB data, dynamic calculations are performed by CAB during transmissions, thus enforcing intelligent decisions and generating reports that detail network performance statistics.

3.2 Resilience Management

Resilience strategies require the network to continuously detect and remediate the effects of operational challenges. Such strategies typically rely on a complex interplay between a number of detection and remediation mechanisms that are activated on demand, according to events observed in the network (as opposed to hardcoded protocols). Different combinations of resilience mechanisms have different associated costs in terms of energy consumption. In order to achieve *energy-efficient resilience management* it is therefore required to understand how the network can be managed in an energy optimal manner without compromising resilience.

We previously advocated the use of policies to specify the orchestration of detection and remediation mechanisms in such resilience strategies [16]. Policies can de-couple the hard-wired implementation of the resilience mechanisms from their management strategy, which is a desirable property considering the changing nature of challenges that the network may face over time. In order to understand how real policies dynamically affect the operation of the network, we are currently working on the integration between a standard network simulator and a policy management framework. This makes it possible to assess the impact of different resilience strategies on the operation and performance

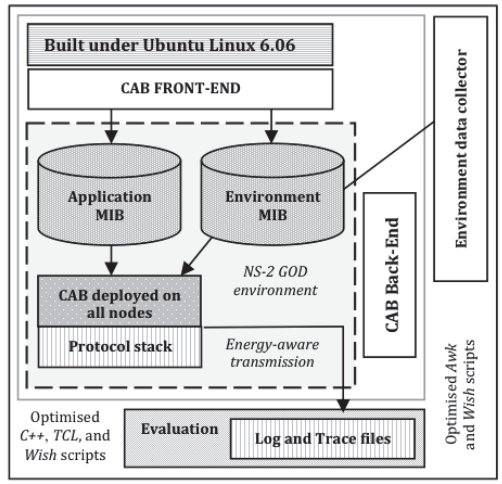

Figure 4: CAB implementation in NS-2

of simulated components. The preliminary prototype [15] is based on an integration between the SSFNet [17] network simulator and the Ponder2[1] policy framework.

```
adaptHigh := factory/ecapolicy create.
adaptHigh event: event/highUtil.
adaptHigh condition: [ :value | value >= 75].
adaptHigh action: [flowexporter
                    enable:60 rate:0.1.].
adaptHigh active: true.
```

Figure 5: Re-configuring flow exporter mechanism

Figure 5 shows a typical policy written in Ponder2 syntax, named `adaptHigh`. This policy is triggered by the event `highUtil`, and specifies that the `flow exporter` component in the simulated network should be enabled, if link utilisation is greater or equal to 75% (`value` is one of the parameters of the event). The event is generated by components in the simulation, for example, a `link monitor` component, and then published in the policy framework. `Flow exporter` is a proxy for an instrumented component running in the simulation. Management interfaces specify what functionality instrumented components in the simulation export to the policy framework. For the policy above, a custom `FlowExporter` class was implemented, whose management interface defines the method `public void enable(Long timeOut, Double samplingRate)`. This permits changing the operation of the simulated component dynamically. However, SSFNet is not suitable for modelling Physical and Link layer properties, and therefore has limited applicability for observing energy related aspects. We are therefore porting the implementation to OMNeT++, since it is capable of modelling energy attributes and is also one of the most popular simulators for research in communication networks.

Our goal is to understand the impact that resilience strategies have on energy consumption in the network, and conversely drive the deployment of resilience mechanisms as a function of energy saving goals. To achieve this we are capitalising on a range of energy related attributes and context information that can be collected in simulation scenarios and managed by the CAB algorithm described in the previous section. This toolset will enable the assessment of

[1] http://ponder2.net

the energy-efficiency of resilience mechanisms dynamically deployed in the network, through the use of simulations.

3.3 Case-study Scenario

A case-study scenario of a corporate network infrastructure under attack (e.g., a botnet raid) is used to illustrate the ideas proposed in this paper. Under normal circumstances it is assumed that the network infrastructure and equipment deployed can work in an energy-efficient mode (e.g., low-packet sampling rate and traffic analysis tools disabled). However, whenever an adverse condition, breach of security or malicious attack is identified, operational efficiency concerns have to take second place as the network resilience strategy takes priority until protection against the attack has occurred (e.g., enable deep packet inspection which will look for anomalies not only in the flow features but also in packet payloads). Table 2 summarises a number of reconfigurations applicable to this scenario when switching from an energy-efficient to a resilience strategy.

Table 2: Reconfigurations for Energy-saving and Resilience Strategies

Energy saving strategy	Resilience strategy
Low packet sampling rate in order to save resources and energy	Increased packet sampling rate to collect more evidence to combat the attack
Reduced transmit power in wireless network given acceptable signal/noise ratio	Resources dedicated to resolve attack as opposed to incurring delay and cost needed to influence the transmit power
Packet inspection and traffic classification tools disabled	Packet inspection and traffic classification tools enabled
Routing path configured taking into account energy cost and queuing delay	Path with lowest delay taken to protect victim more quickly irrespective of the energy cost

Energy efficiency can therefore be provisioned according to a set of performance metrics, which include: the ability to influence *packet-sampling rate*, the ability to influence *transmit power*, the ability to enable or disable *traffic processing tools*, and the ability to influence *routing path*. An ability to manipulate each of these parameters in different simulation environments will determine the suitability of each for our purposes. In this scenario, all equipment reconfiguration is performed with the aide of dynamically loaded policies. With respect to Table 1, those environments which are open source more readily lend themselves to applying policies for autonomic reconfiguration of, for example, the packet sampling rate and signal transmission power. As future work, we intend to monitor, quantify and establish a correlation between the levels of energy and resilience for each mode of operation.

4. LITERATURE REVIEW

In this section, strategies focusing on either energy-efficiency or resilience are presented, validating the ability to use simulation environments for these individual objectives. This also highlights the research gap that we hope to fill with an energy-efficient resilience management approach.

4.1 Energy-Efficiency Simulation

In [8], Opnet was used for modelling energy-efficiency aspects. An energy-aware process model was created in parallel with a routing protocol, adapted radio transmitter and receiver, and CSMA/CD MAC, mobility, packet filtering and topology management modules. Packet filtering modules control node flows as a function of its power state, dumping packets when asleep or with insufficient battery capacity. The topology management module enables wireless sensors to determine their transmission power to conserve energy yet achieve network connectivity. The energy module calculates energy consumed per packet, with node failure occurring when its residual energy reaches zero. The experimental results conform with the expectations in that energy is consumed more quickly in more densely populated areas. For areas of both dense and sparse population, linear consumption of energy is evident as transmission progresses over time. The application of node power-saving states is one which may be replicated and extended as part of our research.

In [7], a routing protocol with energy-efficiency objectives is evaluated in OMNeT++. The main functionality is incorporated in the network layer; energy-efficient decisions involve turning the transceiver off when being powered on is not strictly necessary. Integrated energy and statistics modules allow the evaluation of overall network behaviour and an adaptive energy-aware routing protocol balances load. At the MAC layer, shared access to the communication medium is also balanced for energy-efficiency. This strategy is also important for the purposes of our research, with the exploration of energy-efficient routing strategies potentially representing a component of our work.

4.2 Network Resilience Simulation

In [3], OMNeT++ was integrated with *Distack* [4] and used for evaluating distributed detection of network attacks, such as DDoS and worm propagations. *Distack* allows the implementation of detection mechanisms based on a combination of user-written libraries that perform basic functions such as packet inspection, filtering, sampling as well as several anomaly detection methods. Similarly, our resilience strategies also require the integration between off-the-shelf detection mechanisms and simulation environments.

In [2], a framework to simulate network attacks and challenges in NS-3 is presented. Challenges are classified accordingly to their *domain* (wired or wireless), *scope* (nodes, links, area) and *intention* (malicious or non-malicious). To simulate a specific challenge (e.g. a malicious attack or a large-scale disaster), a certain pattern of *links* and *nodes* is disabled during a given *interval*. The simulation of network challenges is also important for our research, and the framework proposed in [2] might be adapted to our needs.

In contrast to the work introduced in [3, 2], which uses simulations to evaluate the effects of challenges and attack scenarios, we are also interested in evaluating strategies for the remediation of the network. Our work requires not only the simulation of challenges and algorithms for their *detection*, but also the activation of mechanisms that will attempt the *remediation* of the effects of a challenge in the network, based on conditions observed during run-time in the simulation.

5. CONCLUSION AND FURTHER WORK

Core to this research is the development of solutions that tackle energy-efficiency and resilience requirements simultaneously. In order to devise management strategies that combine both aspects, a common test platform is required.

In this paper, candidate simulation environments possessing the building blocks to support our research are reviewed. Initial investigations indicate that NS-3 and OMNeT++ are equally suitable for our requirements. Base-lining experiments are being executed to determine the consistency of results across each, and modifications will be made if and when required. We have recently integrated a simulation environment with a policy framework, in order to allow policy-based decision-making based on conditions monitored within the simulation [15]. Based on our previous work, we are now involved in extending simulation environments with energy-efficient resilience mechanisms. In the future, these extensions can be made available to the research community as a consequence of our selection of open source environments.

To the best of our knowledge there is currently a lack of research approaches for the combined evaluation of energy-efficiency and resilience requirements. Our complementary objectives will enable the simulation of energy-efficient resilience strategies in one or more of the simulation environments explored in this paper. As part of further work, the incorporation of simulation scenarios with live, emulated networks is envisaged. This will assist in further exploring the suitability of energy-efficient resilience strategies once techniques have been developed, implemented, tested, and validated in selected software environments.

6. ACKNOWLEDGMENTS

This work has been supported by the India-UK Advanced Technology Centre in Next Generation Networking, funded by the UK EPSRC Digital Economy Programme and the Government of India Department of Science and Technology.

7. REFERENCES

[1] D. Cavin, Y. Sasson, and A. Schiper. On the accuracy of manet simulators. In *POMC '02: 2nd ACM international workshop on Principles of mobile computing*, pages 38–43, New York, NY, USA, 2002. ACM.

[2] E. K. Çetinkaya, D. Broyles, A. Dandekar, S. Srinivasan, and J. P. Sterbenz. A Comprehensive Framework to Simulate Network Attacks and Challenges. In *RNDM'10: 2nd International Workshop on Reliable Networks Design and Modeling*, Moscow, Russia, October 2010.

[3] T. Gamer and C. P. Mayer. Large-scale evaluation of distributed attack detection. In *Simutools '09*, pages 1–8, Brussels, Belgium, 2009. ICST.

[4] T. Gamer, C. P. Mayer, and M. Zitterbart. Distack – a framework for anomaly-based large-scale attack detection. In *SECURWARE '08: 2nd International Conference on Emerging Security Information, Systems and Technologies*, pages 34–40, Washington, DC, USA, 2008. IEEE CS.

[5] T. R. Henderson, S. Roy, S. Floyd, and G. F. Riley. ns-3 project goals. In *WNS2 '06: Proceeding from the 2006 workshop on ns-2: the IP network simulator*, page 13, New York, NY, USA, 2006. ACM.

[6] International Telecommunication Union, The International Telegraph and Telephone Consultative Committee. X.700 – data communication networks: Management framework for open systems interconnection (OSI) for CCIT applications, 1993.

[7] A. F. Kacso and R. Wismuller. A simulation framework for energy-aware wireless sensor network protocols. In *ICCCN '09: Proceedings of the 2009 Proceedings of 18th International Conference on Computer Communications and Networks*, pages 1–7, Washington, DC, USA, 2009. IEEE Computer Society.

[8] N. Krishnamurthi, S. J. Yang, and M. Seidman. Modular topology control and energy model for wireless ad hoc sensor networks. Technical report, Rochester Institute of Technology, New York, NY, USA, 2004.

[9] D. M. Nicol. Scalability of network simulators revisited. In *Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation Conference*, Orlando, FL, February 2003.

[10] NS-2 Website. The Network Simulator - NS-2. Available [Accessed: Sep. 2010]: http://www.isi.edu/nsnam/ns/.

[11] NS-3 Website. The NS-3 network simulator. Available [Accessed: Sep. 2010]: http://www.nsnam.org/.

[12] OMNeT++ Website. OMNeT++. Available [Accessed: Sep. 2010]: http://www.omnetpp.org/.

[13] OPNET Website. OPNET Modeler Accelerating Network R&D (Network Simulation). Available [Accessed: Sep. 2010]: http://www.opnet.com/solutions/network_rd/modeler.html.

[14] C. Peoples, G. Parr, B. Scotney, and A. Moore. Operational performance of the context-aware broker (CAB): A communication and management system for delay-tolerant networks (DTNs). In *SPACOMM '10: Proceedings of the 2010 Second International Conference on Advances in Satellite and Space Communications*, 2010.

[15] A. Schaeffer-Filho, P. Smith, and A. Mauthe. Policy-driven network simulation: a resilience case study. In *26th ACM Symposium on Applied Computing (SAC)*, Taichung, Taiwan, March 2011. ACM.

[16] P. Smith, A. Schaeffer-Filho, A. Ali, M. Scholler, N. Kheir, A. Mauthe, and D. Hutchison. Strategies for network resilience: Capitalising on policies. In *4th International Conference on Autonomous Infrastructure, Management and Security (AIMS)*, pages 118–122, Zurich, Switzerland, June 2010. LNCS.

[17] SSFNet Website. Modeling the Global Internet. Available [Accessed: Sep. 2010]: http://www.ssfnet.org/.

[18] J. P. G. Sterbenz, D. Hutchison, E. K. Çetinkaya, A. Jabbar, J. P. Rohrer, M. Schöller, and P. Smith. Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines. *Comput. Netw.*, 54(8):1245–1265, 2010.

[19] U.S. Environmental Protection Agency. Report to congress on server and data center energy efficiency public law 109-431. Technical report, U.S. Environmental Protection Agency, 2007.

[20] A. Varga and R. Hornig. An overview of the OMNeT++ simulation environment. In *Simutools '08*, pages 1–10, Brussels, Belgium, 2008. ICST.

[21] E. Weingärtner, H. vom Lehn, and K. Wehrle. A performance comparison of recent network simulators. In *ICC 2009: IEEE International Conference on Communications*, 2009.