A Methodology for Analyzing Power Consumption in Wireless Communication Systems

Mwelwa K. Chibesakunda and Pieter S. Kritzinger

CS03-28-00
December 15, 2003

Data Network Architecture
Department of Computer Science
University of Cape Town
Private Bag, RONDEBOSCH
7701 South Africa
e-mail: {Mchibesa, PSK}@cs.uct.ac.za

Abstract

We propose a methodology that allows software protocol implementors a means of analyzing power consumption in wireless communication systems. The energy-intensive nature of wireless communication has spurred much concern over how best systems can make the most use of this nonrenewable resource. The methodology involves describing the logical flow of protocol data units through the protocol layers based on their formal specification. Here after the state diagrams of the protocol layers are derived from this logical description. These diagrams provide the schema for the *continuous-time Markov chains* that enable the capturing of the protocol layer's behaviour. *Markov Reward Models* are then specified by defining reward structures for these *Markov chains* that allow for the modelling of power consumption as a reward. Modelling power consumption this way allows for the investigation of the *power factor*, *power level analysis*, and *power consumption comparison* of the the radio interface protocol layers.

1 Introduction

Power consumption has become a growing concern among designers of wireless communication systems. With increased mobility, the limited supply of battery power has become a known constraint to the continuous operation time of wireless devices. More so with the proliferation of wireless devices that can support not only voice and data but multimedia applications as well. Unless there are technological advances in battery technology, we have to consider other means of making the best use of the limited resource.

According to Jones *et al.* in [16], the sources of power consumption in a wireless device are as a result of two factors: *communication* and *computation.* Communication, in this regard, considers the power consumed by the radio transceiver, whereas computation is concerned with the protocol processing aspects. Our research confines itself to the computation factor because it has become increasingly obvious that software implementations offer an opportunity of improving power usage in mobile devices.

By considering the wireless device software, we have to have an understanding of the *formal specification* [23] of underlying protocol architecture. The schema of the layer protocol architecture, provides the basis for uncovering the low level functions and procedures associated that are responsible for the power consumption associated with software. This has been the basic premise of most research around energy concerns in wireless systems. We categorize research areas of this nature as, the 1) *investigation of energy consumption* [9, 10, 29], 2) *power saving strategies* [27, 17] and, 3) *energy efficient protocol designs* [28, 7, 5] of wireless communication systems. According to Stark *et al.* in [25], many approaches to energy efficient design attempt to optimize each protocol layer separately and may achieve global optimality only by coincidence. In addition, most investigations only analyze the power consumption of the device after the software has been uploaded. In a way, this becomes a paradox because the software may be correct and efficient (in terms of processor usage) without consideration of the power consumption behaviour of the protocol architecture. This is the fundamental motivation for our work. We propose a methodology that would allow protocol implementors of wireless communication systems an opportunity of analyzing the implementations of the software layers in terms of their power consumption. The methodology is presented in the next section. The rest of the paper provides the necessary theory for understanding *Markov Reward Models* and how they can be used to model power consumption. Hereafter, the power consumption experiments we conducted are presented and the results generated discussed.

2 Methodology

In the study of energy-efficient design of wireless communication systems, the challenge lies in the fact that the overall performance depends, in a coupled way, on the following subsystems [25]:

- *Antenna*,
- *Power Amplifier*,
- *Modulation*,
- *Error control coding*, and
- *Network architecture*

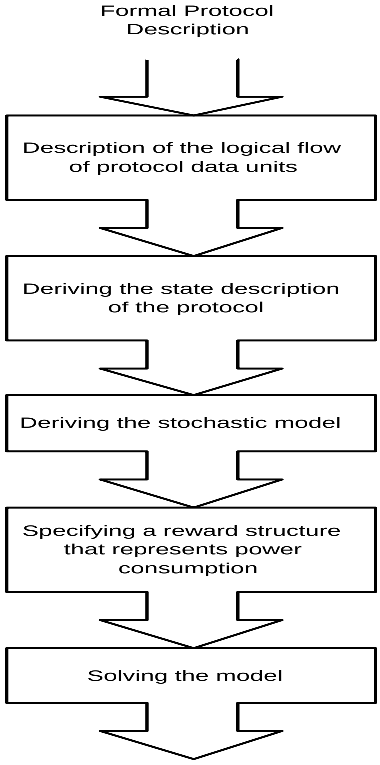

Figure 1: The Methodology Procedure in Outline

There is a considerable amount of power consumed during the operations of the aforementioned. Though a significant subsystem, the *network architecture* poses a more difficult challenge in determining the empirical data associated with their power consumption because of the way the different layers interface with the hardware. We propose a methodology for analyzing the power consumption of a wireless communication system based on the functions and operations documented in the formal specification of their radio interface protocols. The following discussion describes the different parts of our methodology illustrated in **Figure** 1.

Description of Logical Flow of Data Units

Using the formal specification of a protocol, we can derive the logical flow of *protocol data units* (PDUs) as they progress through the layered architecture. Whist the PDUs make their way from *source layer* to *destination layer* (served and servicing layers), there are processes performed on these data units. Processes vary from a combination of instructions executing different protocol functions, to a number of messages needed to transfer a given amount or type of information [26], among other things. Coupled together, these protocol processes contribute to power consumption.

Derivation of Protocol State Description

With an understanding of the processes associated with individual protocol layers, it can be deduced that behaviour of the protocol layer can be represented as a state transition graph where the vertices represent states and arcs represent state transitions. It should be pointed out here that this high level representation is protocol implementor dependent as the protocol specification documents do not provide hard and fast rules for this.

Derivation of Continuous-Time Markov Chain

Using the state transition graphs as our state space, we can derive a *Markov* representation of the protocol layer architecture [19]. We use the assumption from [14] that the behaviour of the real protocol system during a given period of time is characterized by the probability distributions of a stochastic process. The state transition diagrams in this case, indicate the evolution of the system in time and can be represented by the finite-state stochastic process, which characterizes the dynamics of the protocol system of interest.

Specification of Markov Reward Model

In order to analyze the power consumption of the protocols, we propose the use of *Markov Reward Models* (MRMs). MRMs are commonly used for the *performance*, *dependability* and *performability* analysis of computer and communication systems. The motivation for using MRMs is twofold:

1. The evolution of any protocol with finite memory can be modelled as a *Markov chain* [28]; and
2. The freedom to modify the reward structure allows the modeler to represent a wide variety of operating conditions [24].

2.1 Model Solution

Once we have the MRM, we are able to specify the stochastic model, generate the underlying MRM and solve it using the *performability* tool *Möbius* [8] developed by W.H. Sanders *et al.* of the *PERFORM*[1] research group at the University of Illinois. The tool allows us to solve different MRMs based on the specification of reward rates and rewards structures. In particular, the tool can be used to predict the following results over an interval of time:

1. Power consumption in a protocol state, and
2. Power consumption of the protocol layer.

In the next section, we provide the necessary theory of *Markov Reward Models* and how we apply them in our models.

3 Markov Reward Models

A *Markov Reward Model* (MRM) is a *continuous-time Markov chain* $\{X(t), t \geq 0\}$ with finite state space S, and a reward function r where $r : S \rightarrow \Re$. For each state $i \in S, r(i)$ represents the reward obtained per unit time in that state. This type of MRM is called a *rate-based* [22] MRM.

---

[1] PERFORM-Performability Engineering Research Group at the University of Illinois conducts research in the design and validation of dependable distributed systems.

Depending on their intended use, specifications of MRMs allow the modeler to assign different meanings to states, state-transitions, rates, and rewards. Analogous to the example provided by Trivedi and Haverkort in [13], every state signifies a particular function of a protocol layer. State transitions take place when the protocol layer moves from one state to another. The rates of transition are exponentially distribute based on the sojourn time spent in a particular state. The rewards associated with each state are chosen on basis of the measure of interest, i.e. power consumed for being in a particular state.

In [13], Trivedi and Haverkort also categorized four types of measures supported by MRMs and they are listed and discussed below:

- *Steady state measures*;
- *Transient measures*;
- *Cumulative measures*; and
- *Performability measures.*

Steady state measures express the long run gain per unit time of the system. They are computed from the steady state behaviour of the *Markov chain. Transient measures* (or instant-of-time measures) express the rate at which gain is received from the system at any particular time t. They are computed from the transient behaviour of the system. *Cumulative measures* express the overall gain that is received from a system over some finite time interval. They are computed when transient measures are integrated over a specified time interval $[0, t]$. Lastly, *Performability measures* are the a distribution of cumulative measures that express whether a prespecified gain y can be received from the system in some finite time interval $[0, t]$.

Of interest to us is the *steady state measure* because it allows to analyze the behaviour of a system during normal operation over a period of time. We use MRMs because they allows us the freedom to modify the reward structure to represent a variety of situations[24]: in this case, power consumption of the state space of the radio interface protocols of a wireless communication system. In the next section, we discuss how the aforementioned protocol stack was modelled.

4 Modelling Power Consumption as a Reward

This section introduces the concept of using MRMs to analyze power consumption in wireless communication systems. We explain how based on the type of reward structure specified, different power consumption characteristics can be modelled:

Protocol Behaviour

Formal specification documents provide information on the logical flow of protocol data units through a network architecture. The network architecture in this case represents the layered protocol stack. To the protocol implementor writing the program for each layer, the specification provides the *services* of the protocol, its *functions* and the *protocol data units* (PDU) exchanged between adjacent. With this information, the implementor is able to distinguish between the different states of a protocol whilst in execution. We understand this because while a program is in execution, different defined functions of the program will be called depending on what service the protocol is providing or receiving from its adjacent layer. Alternatively, the protocol implementor can use *Formal Description Techniques* (FDTs) [18] to perform a meta-execution of the protocol specification to evaluate the protocol. Most FDTs, for example SDL [15], are able to generate the program source code for protocol execution that can be used to explore the behaviour of the protocol during its execution.

For a protocol in execution, we can assume that it has continuous-time behaviour. We make this assumption because of the unpredictable nature of the wireless communication medium. A typical protocol layer can at any time be sending protocol data units, or be performing error correction due to retransmission. We are able to model this behaviour by using *Markov theory* [28]. Our approach to deriving the *Markov* chain is determined by the fact that whilst a layer is in an arbitrary protocol state (protocol function in execution), there is a amount of time associated with being in that state before the next state change. This is known as the *sojourn time.* We assume the distribution of the layer being in an arbitrary state is *exponential* and that subsequent state changes are determined by the present state: has a memoryless property.

The *Markov* chain that has the properties discussed thus far is the *continuous-time Markov chain.* The protocol processes in this case represent the state space S, where $S = \{i, i \in \mathbb{N}\}$. For a protocol with n states, we let q_{ij} be the *mean transition rate* from state i to state j and Q be the $n \times n$ generator matrix.

Power as a Reward

In order to model power consumption, we can make the following hypothesis based on protocol layer behaviour. Whist the protocol layer is in a particular state, we expect a certain amount of time to be spent in that state. In addition, while in that state, it is obvious that a certain amount of power to be used in order to complete that task. Alternately, if we couple the consumption of power by the different states, power consumption can be viewed as a measure which depends on the accumulated behaviour of all states of a protocol over the utilization interval of that protocol: Power is the reward associated with being in a state. These stochastic behaviors combined with the reward functions form *Markov reward models* [21]. For our purpose, power can be classified as an *interval-of-time* variable for the analysis of *Markov reward models.* The *accumulated reward* earned up in time is defined by [11] as follows:

Definition 1 *If* r_i *is the reward rate associated with the structure-state* i*: then the vector* $\boldsymbol{r}$ *defines the reward structure. A real-valued reward rate* r_i *is associated to each state* $i \in S$*. Assuming that the reward rates are non-negative and have zero absorbing states,* S_A*:*

$$\forall i \in S,\ (r_i \in \mathbb{N}, \mathbb{r}_{\mathbb{i}} \geq \ ) \ \wedge \ (\mathbb{i} \in \mathbb{S}_{\mathbb{A}} \Rightarrow \mathbb{r}_{\mathbb{i}} = \ ) \tag{1}$$

The accumulated reward earned up in time t *is defined by*

$$Y(t) = \int_0^t r_{X(\tau)} d\tau. \tag{2}$$

By interpreting rewards as energy levels, we have derived the following as our definition of power consumption in relation to rewards:

Definition 2 *Let* c_i*,* $i \in \mathbb{N}$ *represent power consumption as a reward in the Markov Reward Model (MRM).*

Though this definition of power consumption provides a way of predicting how the actual power of the device would be consumed by the different protocol states S, it is not sufficient in modelling power consumption of the layer because it does not incorporate parameters such as the the number of packets and their frame size, which would aid in providing a more realistic model. Alternatively, we can assume this to be a coefficient that is part of the overall consumption of the protocol layer. In [10], Feeney and Nilsson describe the energy consumed by the network interface when a host sends, receives or discards a packet under error free conditions by the following linear equation:

$$Energy = m \times size + b \tag{3}$$

The constant coefficients m and b are used to represent values for various operations i.e. packets discarded and packets received. They further state that there is a fixed component associated with device state changes and channel acquisition overhead and an incremental component which is proportional to the size of the packet. Since the protocol architecture is coupled with the wireless interface [25], we can intuitively assume that this description is analogous to the protocol layers as well. We categorize the aforementioned modelling of power consumption as the *Power Factor analysis.*

Another way of using the MRM framework is by using it as a pure (failure-free) performance model [12] to conveniently describe power consumption as a measure of interest. For our purpose, we can analyze the power utilization of the different protocol states. We assume that once the protocol layer is in operation, sending and receiving protocol data units, and the effects of the initial bias have been overcome, the network architecture will enter into steady state behaviour. What this implies is that the model exhibits regularity and predictability over its state space. When the model is in *steady state*, we denote the steady state probabilities i.e., π_i, for the probability that the model is in state i. The utilization, ρ, can be computed based on a binary reward assignment. This is a special MRM where if a particular resource is occupied in a given state, the reward rate 1 is assigned, otherwise reward rate 0 indicates the idleness of the resource. With reward structure $r_0 = 1$ and $r_i = 0$ for $i = 1, 2, 3, \ldots, n$ the utilization becomes:

$$\rho = \sum_{i \in S} r_i \pi_i = \pi_0 \tag{4}$$

This allows us to determine the probability vector π and therefore the fraction of time the protocol is in a particular state. This information proves useful when the protocol implementor wants to determine which of the protocol states the layer spends the most time in and what contribution, to the overall power consumption, that particular state makes. We categorize the aforementioned modelling of power consumption as the *Power Level Analysis.*

Lastly, MRMs can be used to capture the accumulated behaviour of a reward based on the reward specification defined for the model. In this case, if we specify a reward structure that describes the consumption of power by each protocol state, the combined *accumulated reward* of all the states of a specific layer can be used to compare the power consumption characteristics of the various protocol layers specified in a particular layered architecture. This information would allow the protocol implementor to distinguish which protocol layers of a particular implementation are power intensive. We categorize the

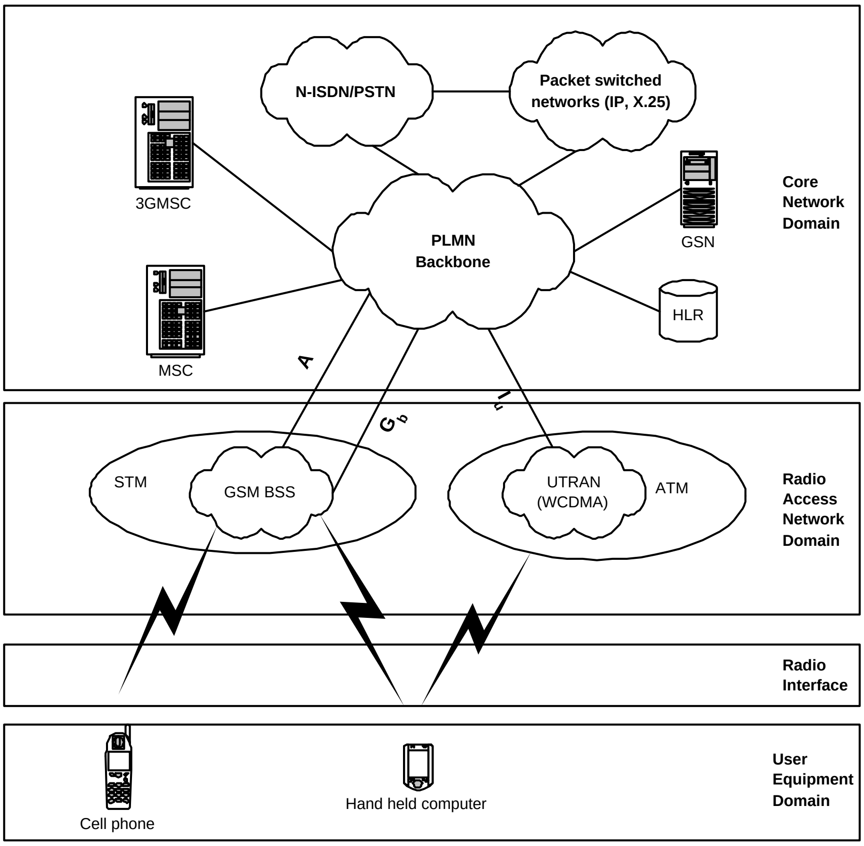

Figure 2: The UMTS Architecture

aforementioned modelling of power consumption as the *Power Consumption Comparison* analysis.

5 Power Consumption Experiments

In this section we provide an example of the application of our proposed methodology in the UMTS protocol stack[2].

In the reminder of this paper, we focus on the radio interface protocol in **Figure** 2. **Figure** 3 from [1] shows the protocol stack for this protocol.

Unfortunately, space does not allow us [6] to use the entire protocol stack to illustrate the application of our methodology. Instead, we choose the physical layer (PHY) [2] and derive the state transition diagram illustrated in **Figure** 4. Similarly for other layers, we used the formal specification documents [3] and [4] to derive the full state transition diagrams of the radio link protocols. Because of their dynamic high level formalism, the state diagrams were subsequently used as the *continuous-time Markov chains* that closely capture the *stochastic behaviour* of these protocol layers. These state diagrams we then used to derive the generator matrices for the *Markov chains.*

In the following subsections, we discuss the experiments we performed and the power consumption characteristics under investigation.

---

[2] UMTS- Universal Mobile Telecommunications System.

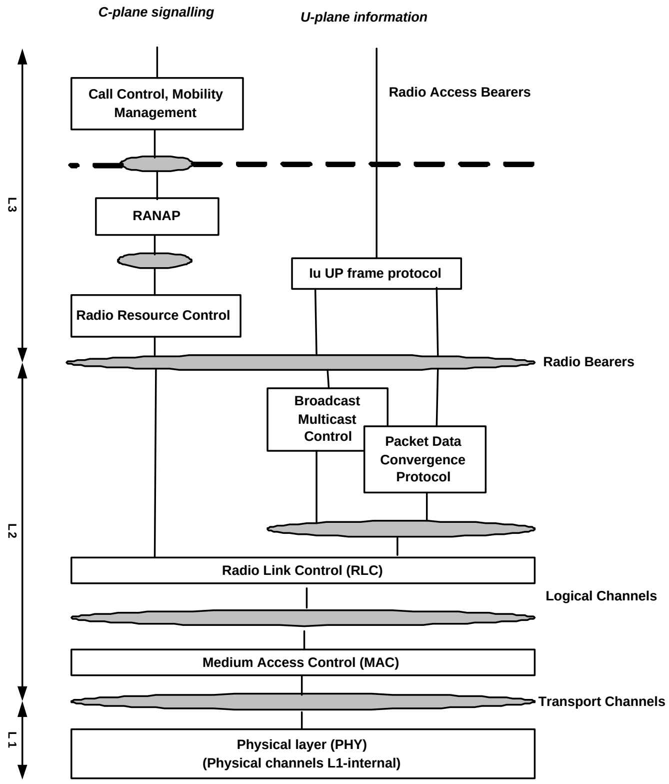

Figure 3: Architecture of the Radio Interface

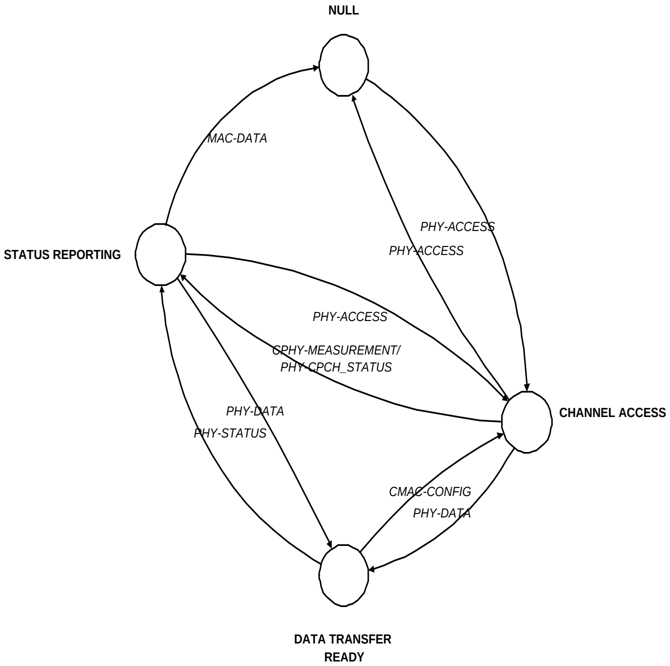

Figure 4: The State Model for the Physical Layer

Power Factor Analysis

For the *Power factor* analysis experiment, we specify two cases in order to observe how the power accumulation can vary during different transmission conditions. In the first case, we modelled a transmission condition in which the protocol spends more time performing measurements on the wireless channel. What this implies is that the expected transition rates into all states associated with channel condition measurements have higher values compared to other states. On the other hand, the second case models a situation where the protocol spends more time in data transfer states during transmission. Ultimately, a comparison can be made of the power factors associated with each of the states under these two conditions.

Power Level Analysis

For the *Power Level Analysis* we specify reward structures based on the MRM property of *utilization.* This type of MRM generates the state-based measure that corresponds to the amount of reward the model contributes to the overall reward in time. This allows us to understand the *power utilization* of each protocol state of a particular layer.

In order to analyze the protocol state utilization, we specified the following as the reward structure for the model: for each state $i \in S$, we set the power reward $c_i = 1$ while $c_j = 0, \forall j \neq i$. A *Markov reward model* of this type is known *availability model* [24]. For our analysis though, this provides an aid to understanding the *power utilization* of each protocol state. In addition, it allows us to determine which states would consume the most power for a particular protocol implementation

Power Consumption Comparison Analysis

For the *Power consumption comparison* analysis we specify reward structures for each of the layers that allows us to compare the accumulated power consumption behaviour of each protocol layer. We compare the accumulated amount of reward in time based on the premise that each protocol layer is expected to accumulate a certain fraction of the overall power consumed. For a particular protocol implementation, this allows the protocol implementor to predict which protocol layers consume the most energy during normal operation. The parameters used for the estimated contribution of reward that each protocol layer would make was based on the intuitive assumption of the expected consumption associated with being in a particular state. In practice, these parameters would be determined by the *protocol implementor* based on the desired layered architecture [26].

6 Results

The results presented are for the experiments modelled using the Möbius *Accumulated Reward Solver* (ARS) [20, 8]. The parameter values used for transition rates and power are based on intuitive estimates for purposes of illustrating the proposed values. Actual empirical data proved difficult to come by, though it is not impossible to acquire these values if the specific instrumentation tools are available.

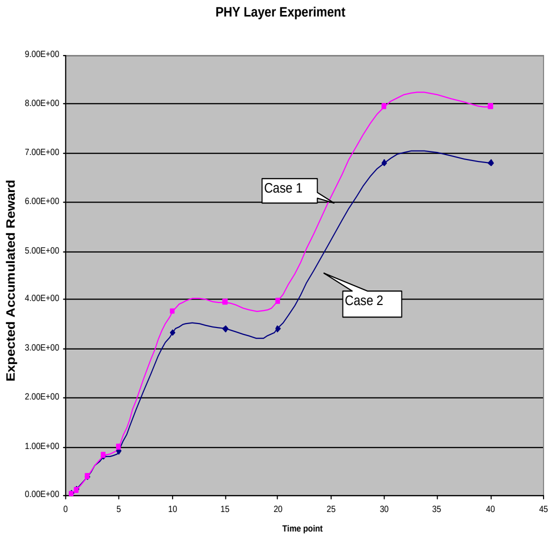

Figure 5: Expected Accumulation

Power Factor

Figures 5 and 6 illustrate the *expected* and *time-averaged accumulated reward* results of the PHY layer respectively. They compare the power consumption characteristics when more time is spent performing measurements and data transfer, in *Case 1* and *Case 2* respectively. These graphs show the initial transient behaviour and the subsequent steady-state behavior. Results of this sort can be very useful to the protocol implementor in understanding and visualizing the expected power consumption of the protocols under different operational conditions.

The expected accumulation results compare the expected accumulation of reward in time of the two experiment cases. It shows that for the parameter values assessed, the protocol layers consume more energy when they are performing measurements than when they are data transmitting.

The time-averaged accumulation results show that after a certain amount of time, the consumption stabilized for both cases. This indicates that the protocol layer has entered into a steady state. Using this result, we are able to determine which case consumes the most and the least amount of power.

Power Level

The results presented here compare how much power reward each state of the RLC layer protocol would contribute over a period of time. Figures 7 and 8 illustrate the combined expected accumulated reward and time-averaged accumulated reward results.

These results show which states (denoted by $Exp\ AR^3$ r1→ $r\mathbb{N}$) are expected to consume the most amount of power. In addition, the results illustrate the expected power consumption behaviour associated with each protocol state over a period of time. This is determined by the frequency with which are state is visited. Analogous to the *power factor* experiment, these results are largely dependant on the transition rates between the

---

[3] Exp AR denotes the Expected Accumulated Reward.

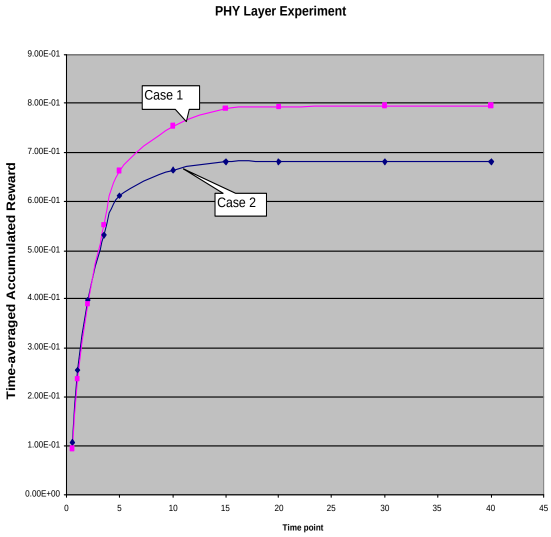

Figure 6: Time-Averaged Accumulation

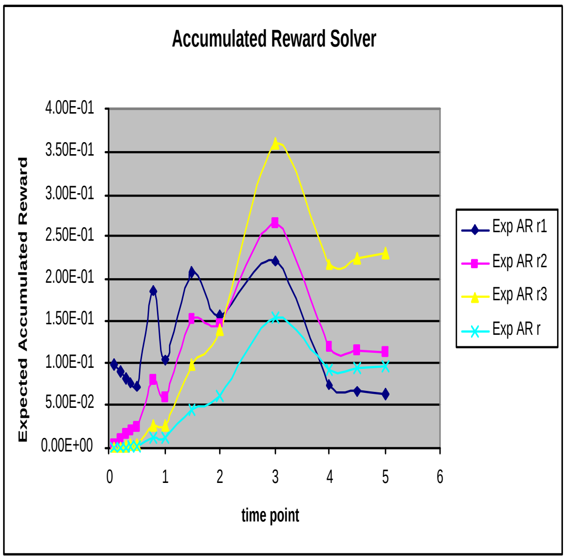

Figure 7: Expected Accumulated Reward

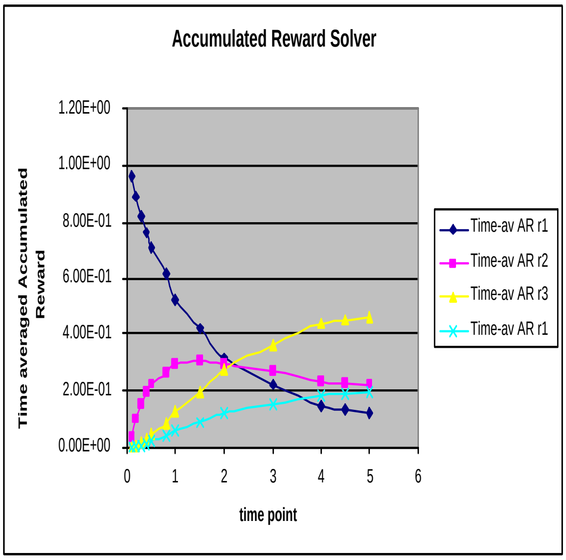

Figure 8: Time-Averaged Accumulated Reward

various states. For example, we can observe from **Figure** 7 that the state that returns reward $r1$ is more likely to be the most frequently visited state. Results of this sort can be very useful to the protocol implementor in determining which of the protocol states are power intensive based on a particular protocol layer implementation. At this stage, the implementor has an opportunity to address the implementation overhead caused by power intensive protocol states before the final protocol program code is uploaded on the wireless device.

Power Consumption Comparison

Figure 9 illustrates the power consumption comparison of specific protocol implementations of the RLC, MAC, and PHY layers. The graphs illustrate how a protocol implementor can determine and compare expected power consumption characteristics of part or the entire protocol architecture implementation. It provides insight into which protocol layers consume the most and the least amount of energy when the mobile device is in a steady state. In this particular case, the MAC layer is expected to consume the most, and the RLC layer consumes the least amount of power for this particular protocol implementation.

7 Conclusions

In this paper, we have proposed a methodology for analyzing the power consumption in wireless communication systems. We have discussed some of the short comings of present research work in the energy concerns in wireless communication systems. By incorporating a mechanism that allows protocol implementors an opportunity to analyze the power consumption of the protocol layer software before it is implemented, we can address the issue of power intensive software at a design stage. *Markov Reward Models* offers the type of modelling flexibility that protocol implementors can take advantage of,

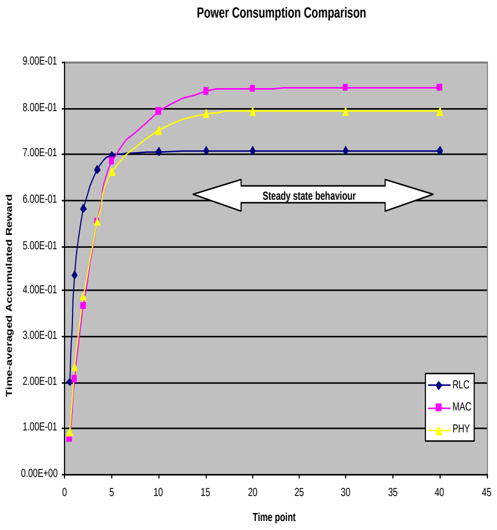

Figure 9: Compared Power Consumption Of Individual Protocol Layer Implementations

in order to model the different wireless communication scenarios associated with wireless communication.

Acknowledgments

The authors would like to thank Tod Courtney from the *PERFORM* Research group at University of Illinois for helping in understanding the Möbius Modelling tool, and Kruger Murray from Siemens Communications South Africa for the many insightful conversations.

References

[1] 3GPP TS 25.301 Universal Mobile Telecommunications System (UMTS) version 5.2.0 Release 5. Radio Interface Protocol Architecture. WWW page http://www.etsi.org, August 2002.

[2] 3GPP TS 25.302 Universal Mobile Telecommunications System (UMTS) version 5.3.0 Release 5. Services provided by the physical layer. WWW page http://www.etsi.org, December 2002.

[3] 3GPP TS 25.321 Universal Mobile Telecommunications System (UMTS) version 5.3.0 Release 5. Medium Access Control-MAC Protocol specification. WWW page http://www.etsi.org, December 2002.

[4] 3GPP TS 25.322 Universal Mobile Telecommunications System (UMTS) version 5.3.0 Release 5. Radio Link Control-RLC Protocol specification. WWW page http://www.etsi.org, December 2002.

[5] J.-C. Chen, K. M. Sivalingam, P. Agrawal, and S. Kishore. A Comparison of MAC Protocols for Wireless Local Networks Based on Battery Power Consumption. *IEEE: INFOCOM*, pages 150–157, 1998.

[6] M. K. Chibesakunda. A Methodology for Analyzing Power Consumption in Wireless Communication Systems. Masters by coursework and dissertation, University of Cape Town, DNA Group, Computer Science Depart, UCT, Rondebosch 7700, December 2003.

[7] A. Chockalingam and M. Zorzi. Energy Efficiency of Media Access Protocols for Mobile Data Networks. *IEEE Transactions on Communications*, 46(11):1418–1421, November 1998.

[8] G. Clark, T. Courtney, D. Daly, D. Deavours, S. Derisavi, J. M. Doyle, W. H. Sanders, and P. Webster. The Möbius Modeling Tool. Proceedings of the 9th International Workshop on Petri Nets and Performance Models, Septermber 11-14 2001.

[9] L. M. Feeney. Investigating the Energy Consumption of an IEEE 802.11 Network Interface. WWW page http://citeseer.nj.nec.com/325357.html, December 1999.

[10] L. M. Feeney and M. Nilsson. Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. *IEEE INFOCOM 2001 - The Conference on Computer Communications*, 0(1):1548–1557, April 2001.

[11] G. Ciardo and R. Marie and B. Sericola and K. S. Trivedi. Performability Analysis Using Semi-Markov Reward Processes. *IEEE Transactions on Computers*, C-39(10):1251–1264, October 1990.

[12] K. Goseva-Popstojanova and K. S. Trivedi. Stochastic Modeling Formalisms for Dependability, Performance and Performability. In *Performance Evaluation*, pages 403–422, 2000.

[13] B. R. Haverkort and K. S. Trivedi. Specification Techniques for Markov Reward Models. *Discrete-Event Dynamic Systems: Theory and Applications*, 0(3):219–247, 1993.

[14] J. Hillston. Constructing and Solving Markov Processes. WWW page http://www.dcs.ed.ac.uk/teaching/cs4/www/modsim/notes/note3.ps Accessed July 2003.

[15] D. Hogrefe, F. Belina, and A. Sarme. *SDL with applications from protocol specification.* Prentice Hall International, 1994.

[16] C. E. Jones, K. M. Sivalingam, P. Agrawal, and J.-C. Chen. A Survey of Energy Efficient Network Protocols for Wireless Networks. *Wireless Networks*, 7(4):343–358, 2001.

[17] R. Kravets and P. Krishnan. Application-driven power management for mobile communication. *Wireless Networks*, 6(4):263–277, 2000.

[18] P. Kritzinger. Communication Systems and Computer Networks- An Advance Course. Course notes in the Department of Computer Science, University of Cape Town, 2003.

[19] P. S. Kritzinger and F. Bause. *Stochastic Petri Nets - An Introduction to the Theory.* Vieweg Verlagsgesellschaft, 1996.

[20] PERFORM. *UltraSAN User's Manual.* The University of Illinois, Center for Reliable and High-Performance Computing Coordinated Science Laboratory University of Illinois at Urbana-Champaign, version 3.0 edition, 1994.

[21] M. A. Qureshi. *Construction and Solution of Markov Reward Models.* Doctor of phylosophy, University of Arizona, Department of Electrical and Computer Engineering, 1996.

[22] S. Racz and M. Telek. Performability Analysis of Markov Reward Models with Rate and Impulse reward. WWW page http://citeseer.nj.nec.com/acz99performability.html, 1999.

[23] H. Rudin. From formal protocol specification towards automated performance prediction. In *Workshop Protocol Specification, Testing, Verification, Rüschlikon, Switzerland*, pages 257–269, 1983.

[24] R. M. Smith, K. S. Trivedi, and A. V. Ramesh. Perfromability Analysis: Measures, an Algorithm, and a Case Study. *IEEE Transactions on Computers*, 37(4):406–417, April 1988.

[25] W. Stark, H. Wang, A. Worthen, S. Lafortune, and D. Teneketzis. Low-energy wireless communication network design. *IEEE Wireless Communications*, 9(4):60–72, August 2002.

[26] L. Svobodova. Implementing OSI Systems. *IEEE Journal on Selected Areas in Communications*, 7(7):1115–1130, September 1989.

[27] H. Woesner, Jean-Pierre, M. Schläger, and A. Wolisz. Power Saving Mechanisms in Emerging Standards for Woreless LANs: the MAC Level Perspective. *IEEE Personal Communication Systems*, June 1998.

[28] M. Zorzi and R. R. Rao. Energy-constrained error control for wireless channels. *IEEE Personal Communications*, 4(6):27–33, December 1997.

[29] M. Zorzi and R. R. Rao. Energy Efficiency of TCP in a Local Wireless Environment. *Mobile Networks and Applications*, 6:265–278, 2001. Dipartimento di Ingegneria, Universit di Ferrara, 44100 Ferrara, Italy.