

Pitfalls in Energy Consumption Evaluation Studies

Kostas Pentikousis

VTT Technical Research Centre of Finland
Kaitoväylä 1, FI-90571 Oulu, Finland
kostas.pentikousis@vtt.fi

Abstract—Energy consumption is fast emerging as a central item of research for the Future Internet (FI). In an environment where the majority of users access next generation networks using battery-powered devices, energy efficiency is of equal importance to other well-established metrics of overall system and protocol performance. Nevertheless, we still lack benchmarks of what constitutes energy efficient operation in mobile computing and networking. In particular, models that accurately quantify transport protocol energy consumption are in short supply. This paper introduces a blueprint for a configurable and extensible energy consumption model which takes battery depletion and recovery characteristics into consideration. We show that the non-linear nature of battery depletion cannot be ignored in energy consumption evaluation studies. Battery recovery effects should be accounted for as well. We explore the potential gains in different TCP variants using our model and illustrate that data-counting and time-counting are not sufficient methods for assessing the energy efficiency of transport protocols operating on devices with fixed-energy sources.

I. INTRODUCTION

Martin and Siewiorek [1] point out that for mobile computing devices, power consumption is as important a constraint as performance and cost. They observed that the relationship between power consumption and performance has not been studied systematically. In fact, the authors emphasized that “the techniques for reducing power have, so far, been ad hoc.” This goes double for the protocols in the TCP/IP protocol stack: we still do not have straightforward mechanisms for making tradeoffs between energy efficiency and performance.

At the lower end of the protocol stack, several power-efficient MAC/PHY layer protocols have been proposed and, in general, every effort is made to increase energy efficiency. For example, wireless LAN standards such as the IEEE 802.11 family of protocols, cellular technologies standardized by 3GPP and 3GPP2, as well as wireless MAN standards such as WiMAX, employ sophisticated rate adaptation schemes to achieve the highest possible raw bit rates at the most economical power level for a set level of interference. Nevertheless, the design of energy efficient MAC/PHY layer protocols is not coordinated with the design of power conserving transport or application layer protocols. As a result, many features that were meant to be power conserving prove either not to be so or, even worse, actually hurt performance. For example, Krashinsky and Balakrishnan [2] demonstrate that when the IEEE 802.11 power saving mode (PSV) is employed, users issuing web-like request/replies experience unjustified long delays. PSV, a MAC layer mechanism, was introduced without

considering the anticipated application workloads or transport layer mechanisms, and thus fails to deliver energy savings while maintaining reasonable performance.

Alternative protocol stacks designed for mobile devices, such as the Wireless Application Protocol (WAP) suite, failed to achieve widespread adoption. WAP 1.0 featured a power conserving stack, incompatible with the TCP/IP stack [3]. Internet connectivity can only be achieved through WAP gateways, and only for some Internet services. For example, WAP-enabled phones can access custom-made information resources (“WAP sites”), but cannot connect directly to web sites, FTP servers, databases, and so on. This proved a severe limitation [4]. Today, all-IP architectures appear to be at the center of emerging mobile and wireless standards such as System Architecture Evolution/Long Term Evolution (SAE/LTE) by 3GPP and Mobile WiMAX by the WiMAX Forum [5].

In particular with respect to end-to-end protocols, TCP remains the preferred transport protocol due to its ubiquity, rich set of features, and extensive programmer experience with it. TCP was not designed for battery-powered devices and does not take into account the battery state of charge (SOC), nor regulates its sending rate based on any information from the lower layers regarding power consumption. Still, TCP, in general, and TCP/ECN, in particular, is very frugal and keeps the overall wasted transmissions to a bare minimum, as we showed in previous work [6]. Alternative reliable transport protocols such as the Stream Control Transmission Protocol (SCTP) [7], support a *message-oriented* service model which may be more suitable for mobile applications than the *byte-stream* service model of TCP [8]. Even so, SCTP’s adoption in wireless stacks is limited. Introducing a different transport protocol, let alone a complete protocol stack as in the case of WAP 1.0, suffers from the lack of backwards compatibility. Modifying TCP mechanics to consider energy costs may be more promising as it is easier to deploy and does not require application rewriting. Significant energy gains may be achievable by taking into account the protocol transmission behavior. For example, mobile devices consume more power while transmitting and receiving data than when idling. Keeping the network interface “on” but idle places a high tax on energy consumption [9] and, therefore, the focus is shifted to minimizing the number and duration of idle periods [2].

Unfortunately, as mentioned earlier, we still lack methods and mechanisms for evaluating transport protocol energy efficiency. For instance, time-counting and data-counting are often

the only gauges for estimating energy efficiency. This paper argues that the non-linearity of battery depletion and recovery effects have to be accounted for as well. Thus, we introduce a blueprint for an energy consumption model which can be used to better assess the energy efficiency of transport protocols.

The rest of this paper is organized as follows. Section II overviews the methods used so far to quantify power efficiency. Sections III and IV introduce the basics of battery capacity depletion and the proposed energy consumption model. Section V reveals the pitfalls when using data- or time-counting as energy-efficiency gauges of transport protocols. Section VI concludes the paper and outlines future work items.

II. RELATED WORK

Simulation and modeling of batteries is not, in general, a new field. Motivated by the need to design well-performing and energy-efficient electric-only and hybrid-electric vehicles (HEV), electrical engineers and chemists have made considerable progress in modeling the electro-chemical, thermodynamic, and quantum effects that govern battery operation. On the other hand, computer scientists and engineers assess efficiency without introducing battery models in their studies but, instead, measure solely the energy consumed (often using indirect metrics). We feel that the results obtained in other disciplines can be of great use to designers of mobile computing systems. Network researchers, protocol designers, and application developers working in the mobile/portable battery-operated wireless devices area can benefit from understanding how batteries work and how certain decisions can affect overall system performance. We use the term “overall system performance” to highlight the importance of battery life alongside the usual performance metrics, such as, goodput.

McFarland and Wong study the energy efficiency of IEEE 802.11 and point out that transmitting and receiving at higher rates is more power consuming, but the energy consumed per megabyte transmitted at higher data rates is less in “typical” scenarios [10]. The caveat is that the network interface card should be switched off as often and for as long as possible. Their results provide insights for energy-conscious system design but do not account for battery capacity depletion and recovery. Indeed, few network protocol studies use battery capacity depletion models to evaluate the energy efficiency of existing and proposed transport protocols. Energy efficiency is determined by measuring power consumption *per se*, number of bytes transmitted and received, *average* energy spent per transaction, and total transfer time [2], [11]–[18]. These studies have added a lot to our understanding of the issues pertaining to TCP behavior and performance in an energy challenged environment, but are all based on the assumption that the power source is extremely large and SOC decreases linearly. In effect, the underlying assumption is that all packets incur the *same* toll on a battery. Other studies, *e.g.* [19], use finite capacity battery models, but use stochastic traffic patterns and the effects of reliable transport protocols are not considered.

Empirical studies with test beds [9], [11], [16]–[18] also provide insights on how real systems consume energy, but

have their own limitations. For example, Agrawal and Singh [11] explore the scenario where a user discharges the battery completely in one sitting while the only active components are the network interface and the CPU — all other components (secondary storage, display, *etc.*) are explicitly turned off. Their main metric is the amount of data TCP transfers until the battery is completely spent. This scenario does not take into account the battery recovery effects (Section III), and is not very typical since most users use mobile devices on a non-continuous basis. In addition, network usage coincides with CPU, disk and screen usage for laptops. PDAs and smartphones may be different in this respect, but they are also typically used on a non-continuous basis.

III. BATTERY DEPLETION MODELS

Battery discharge is governed by non-linear, complex phenomena. Peukert’s equation associates battery capacity, C , with a *constant* discharge current, I , and the total time needed to completely discharge the battery, T :

$$C = I^n T \quad (1)$$

Peukert’s number, n , is measured empirically and depends on the battery chemical and physical properties. Typically $n \in (1, 2)$; good lead-acid batteries have $n \leq 1.2$. Ideal batteries discharge linearly ($n = 1$), but are a theoretical concept.

The state of charge (SOC) provides a metric of the available capacity left in a battery with initial capacity C , at time t . $\text{SOC}(t) : [0, \infty) \rightarrow [0, 1]$ is defined as

$$\text{SOC}(t) = 1 - \frac{C_{used}(t)}{C} \quad (2)$$

When discharged at a constant current I , C_{used} can be approximated by $C_{used}(t) = I^n t$ and, thus, from Eq. (2)

$$\text{SOC}(t) = 1 - \frac{I^n t}{C}, \quad I = \text{constant} \quad (3)$$

Although Peukert’s equation models quite accurately the non-linearity of battery discharge, it does not capture another extremely important factor which is the battery recovery effect. When a battery is left to rest, chemical reactions partially restore lost capacity. Battery recovery is a phenomenon that needs special attention as it can be exploited to deliver larger battery lives. That said, battery recovery can occur only with intermittent use. To better illustrate and motivate this, consider the case where we want to calculate the battery life of a personal digital assistant (PDA) under different workloads.

Fig. 1 illustrates the results of a simple simulation study where current is drawn from a 1 Ah battery at different levels depending on the PDA workload. Examining the simulation data, we see that when the user shuts down the PDA for random intervals and continues from where she left off, the battery discharge duration can increase by more than 15%. Estimating the available battery life without accounting for the battery recovery effect and the usage patterns may yield unrealistic simulation results. Although the exact recovery behavior is battery-dependent, we believe that it is a critical

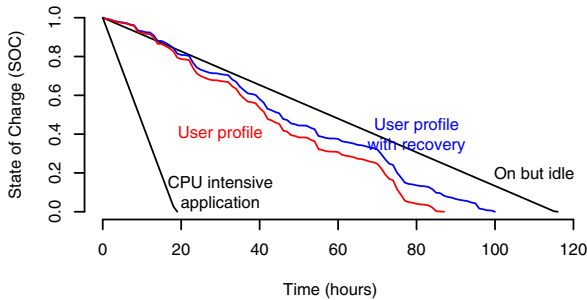


Fig. 1. PDA battery depletion simulation study

characteristic that protocol evaluations need to pay special attention to. What becomes clear from the above discussion is that “measuring” power consumption or, as it is also referred to as energy expenditure, in simulation studies cannot be based on assumptions about the linearity of fixed power sources.

We revisit Eq. (2) and introduce $C_{recovered}$, the battery capacity replenished due to the recovery effect. We can now redefine $SOC(t) : [0, \infty) \rightarrow [0, 1]$ to account for the recovery effect in addition to the non-linear behavior:

$$SOC(t) = 1 - \frac{C_{used}(t) - C_{recovered}(t)}{C} \quad (4)$$

To sum up, assessing energy costs in simulation studies cannot be based on the assumption that fixed-sized power sources discharge linearly. Even if one has good grounds to ignore the recovery effect, the characteristics of battery discharge need special attention. For example, the argument in [10] where a higher transmission rate justifies a higher current drawn from the battery as long as the network interface is switched off frequently, effectively assumes linearity with respect to battery depletion. The energy spent per megabyte metric calls for a battery model to be introduced as well.

IV. ENERGY CONSUMPTION MODEL

Several battery life estimation algorithms have been proposed based on electro-chemical models (including Peukert’s equation), equivalent circuit models, hydrodynamic models, finite element models, tabulated data models, and even neural networks; see for example [20]. Our goal is neither to propose a new battery depletion model nor develop a new battery life estimation algorithm. Instead, we design an extensible and configurable energy consumption model suitable for transport protocol simulation studies. The aim is to use this model as the basis for defining benchmarks for transport and application layer protocols. The model accounts for protocol usage patterns and features a pluggable battery depletion model. This way, protocol designers can take advantage of the essential characteristics of battery depletion behavior when designing new protocols, and advances in battery modeling can be incorporated in a straightforward manner.

The energy consumption model has the following states:

a) *SLEEP*: The model enters this state when no traffic is being transmitted or received *and* the protocol does not anticipate to send or receive any packets in the near term. This state corresponds to effectively shutting down the interface and is the most energy-conserving. This is the only state in which battery recovery effects can occur. We will refer to the current drawn from the battery while in this state as I_{sleep} . In certain occasions we shall use the notation $I_{sleep} \geq 0$ to mean that (a) recovery effects are not taken into account, and (b) that some (possibly minuscule) current is drawn even while in SLEEP. When $I_{sleep} < 0$ the battery recovery effect is simulated.

b) *IDLE*: The model enters this state when no traffic is sent or received, but the protocol is in active operation. This means that incoming packets may arrive at any time and that the protocol has data to send but does not do so for reasons not related with energy consumption. For instance, take the case of a TCP sender that has initiated the connection establishment three-way handshake by sending the first SYN. TCP is now waiting for the packet carrying the SYN/ACK to arrive and has data to send. The energy model remains in IDLE from the moment the SYN is transmitted till the SYN/ACK arrives.

In general, more power is consumed while the model is in the IDLE state than when it is in SLEEP. The energy consumption model dictates that a state transition from IDLE to SLEEP can occur only if the sojourn time in SLEEP will exceed a threshold value, denoted by $T_{recovery}$. If the protocol cannot guarantee that it can allow for the energy model to stay in SLEEP for a period $t \geq T_{recovery}$, and further preserve energy, then the model remains in IDLE. In this case, the protocol cannot take advantage of the battery recovery effects. An energy-efficient protocol will strive to avoid keeping the power model in IDLE for prolonged periods and, instead, will attempt to permit it to enter and remain in the SLEEP state as much as possible. We will refer to the current drawn from the battery while in this state as I_{idle} , which is always positive.

c) *TRX*: The model enters TRX only when the protocol transmits data. For full duplex communications, the protocol can simultaneously transmit and receive. The model will remain in TRX until the protocol completes the transmission or until data reception commences. If data transmission ends and nothing is being received, the model transitions to IDLE. The current drawn from the battery while transmitting packets is denoted by $I_{trx} > 0$.

d) *RCV*: Analogous to TRX, when the protocol starts receiving data the model enters the RCV state. When all data is received, the model transits to IDLE unless the protocol initiates a transmission. For full duplex communications, the protocol may initiate a transmission before the receipt of data is completed. In this case, the model enters the state we present next, TRV. The current drawn from the battery while the protocol is receiving packets is denoted by $I_{rcv} > 0$.

e) *TRV*: This state is entered only if the protocol is capable of transmitting and receiving simultaneously. Typically, the protocol starts transmitting (receiving), and before all packets are sent (received) it starts receiving (transmitting) packets. We denote the current drawn while in this state as $I_{trv} > 0$.

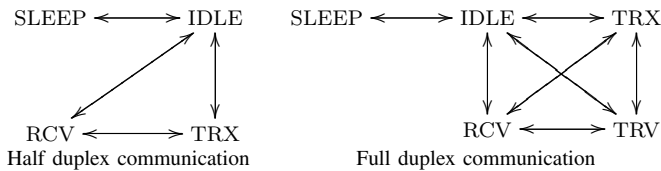


Fig. 2. Energy consumption model state transition diagram

As emphasized above, the model can enter the SLEEP state only if the state sojourn time $t \geq T_{recovery}$, which depends on the chemical and physical form of the battery, and often varies between batteries of the same type. In our model, state transitions do not incur any additional energy costs. In real systems, a transition from IDLE to SLEEP, and vice versa, is rarely cost-free and is usually coupled with shutting down and powering up costs. To capture these costs, the simulationist can increase the recovery threshold so that the model cannot enter the SLEEP state unless the sojourn is large enough to mitigate the shutting down/powering up costs. Moreover, when communication is half-duplex, additional costs may be associated with switching the transceiver from TRX to RCV and vice versa [9]. In these cases, an energy saving protocol will attempt to minimize the number of such state transitions.

Fig. 2 summarizes the state transitions described above. We implemented this model as a stand-alone module that takes ns-2 simulation output traces as input and calculates the energy profile for a given sender or receiver. The user specifies C , n , I_{sleep} , I_{idle} , I_{trx} , I_{trv} , and $T_{recovery}$. The following section presents characteristic results from a large set of TCP simulations using the proposed energy consumption model.

V. RESULTS

To portray the usage of the energy consumption model and highlight its advantages in assessing the energy profile of protocols, we conducted an extensive set of simulations using ns-2. We employed the topology shown in Fig. 3, which includes a subnetwork of $n+1$ identically configured download clients, one of which is designated as our “dedicated” battery-powered client. All clients are connected with point-to-point links to router A. We vary the client access network link capacities and propagation delays. The clients connect through router A to router B via a WAN-like link with capacity B Mb/s and propagation delay of d ms. We experimented with B set to 1.5 and 155.52, and d set to 2, 20, and 1000.

The topology includes two identically configured servers connected to router B by a 1 Gb/s point-to-point link with 100 ns propagation delay. Router B manages its interface queue towards A employing either Drop Tail (DT), Random Early Detection (RED) or RED with Explicit Congestion Notification (ECN). The queue size was varied, along the lines of the discussion in [6]. The rest of the links use DT and are over-provisioned to ensure no packet drops occur.

The dedicated client issues a single data transfer request at time $t = 10$ s for an application payload of f megabytes, where $f \in \{0.1, 1, 5, 20\}$. The remaining n clients issue a request

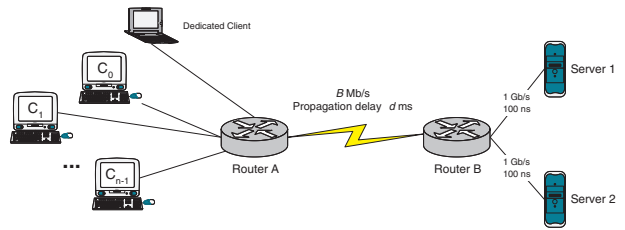


Fig. 3. The simulated network topology

for $10f$ MB at time $t = 0$ s. The data transfers occur between Server 1 and the set of n clients on the one hand, and Server 2 and the dedicated client, on the other. Different scenarios have the servers uniformly employ TCP Tahoe or Reno or NewReno with and without ECN. The clients use the delayed acknowledgements algorithm in half of the experiments and acknowledge every segment in the other half. The delayed ACK timer is set to 100 ms. The MSS is 1460 B, and the initial congestion window is set to 2 MSS.

Due to space restrictions, we present only some examples that illustrate two points. First, data- and time-counting can frequently lead to incorrect estimations about energy efficiency and are thus inappropriate gauges. Second, the proposed model can serve as a key tool in examining protocol behavior with respect to energy consumption in a consistent manner and has thus the potential to guide future protocol designs.

A. The data-counting pitfall

Consider the case of $n = 10$ clients connecting via a 11 Mb/s, 0.5 ms access link, $f = 0.1$ MB, $B = 1.5$ Mb/s, $d = 2$ ms, and the queue limit on router B is 64 packets (Fig. 3). Server 2 uses TCP Reno with ECN and the dedicated client does not delay ACKs (Reno/ECN/ACK). The goodput of the dedicated server/client pair is 83 kb/s, transmitting a total of 112 540 B, of which 4500 B (or 3 full-sized packets) are dropped at router B. The client sends a total of 2920 B in pure ACKs and receives 108 040 B, including the handshaking segments needed to establish the connection. A NewReno dedicated sender operating in a non-ECN network loses 3000 B while the client, which implements delayed ACKs, transmits 2040 B in pure ACKs (NewReno/noECN/delayedACK). If we simply use data-counting we would conclude that there is little difference in terms of energy consumption. However, the NewReno configuration achieves a goodput that is 81% better than the Reno scenario. By adopting time-counting as an energy consumption gauge, one would conclude that using NewReno with delayed ACKs can save energy *and* actually deliver significantly better goodput in a congested network.

Assume, however, that it is possible to relax the restriction of $T_{recovery} = \infty$, and set $T_{recovery} = 100$ ms. Shutting down the network interface for such durations is reasonable and is also employed by other schemes (such as [2]). We conservatively set $I_{sleep} = 0$ A, *i.e.* we do not simulate the battery recovery effect. After processing the simulation trace with our power model we find that the NewReno configuration

consumes 91% more battery capacity at the dedicated client than the Reno configuration. Close examination of the state transitions reveals that in the latter configuration there are periods sufficiently large for the SLEEP state to be visited and thus energy can be conserved. On the other hand, the “NewReno/noECN/delayedACK” configuration displays transmission patterns that prevent sojourn in SLEEP, which nearly doubles the total battery capacity consumed. In this case, neither data- nor time-counting provide accurate estimates.

B. The time-counting pitfall

Take $n = 25$ and the case of a Tahoe sender vs. a Reno sender (both over DT). The TCP Reno sender ends up attaining only 63% of Tahoe’s goodput. In both cases, the dedicated client implements delayed ACKs but ECN is not supported. By adopting time counting as a gauge for energy efficiency one would predict that in this scenario the Reno sender would consume approximately 60% more energy. Using our energy consumption model we find that indeed the TCP Reno sender consumes 61% more energy.

Consider, however, the case of $n = 25$ clients connecting via a 11 Mb/s, 0.5 ms access link, $f = 0.1$ MB, $B = 1.5$ Mb/s, $d = 20$ ms; the queue limit on router B is 64. The Reno/ECN dedicated sender finishes the download in 17.338 s while the Tahoe/DT sender completes the download in 17.38 s. Both clients use delayed ACKs. Taking into consideration that Reno/ECN transmitted only 40 bytes more than Tahoe, one may conclude that the energy consumed in both configurations should be about the same. As in the previous subsection, we set $T_{recovery} = 100$ ms and process the simulation output trace with the proposed energy model. We find that, in fact, Tahoe uses 55% more battery capacity than Reno. A study of the simulation trace and the model transitions reveals that transmission patterns play an important role in power consumption measured using our model. In this case, Reno experiences large periods of inactivity that can be taken advantage of to preserve battery capacity. This conclusion cannot be reached using the transfer size, duration, or a function that uses some arbitrarily weighted aggregate of the two.

VI. CONCLUSION AND FUTURE WORK

This paper presented a blueprint for an energy consumption model that can be used in simulation studies to measure the efficiency of different transport protocols. The model is extensible and configurable and captures the non-linearity of fixed-size energy sources as well as recovery effects, based on established research of battery behavior and modeling. We used our energy model to highlight certain pitfalls when assessing power efficiency using simulation. Overall, our experiments did not reveal that any particular TCP variant or configuration has an overwhelming or consistent advantage across the board. Note, nevertheless, that this investigation did not attempt to rank TCP versions or features *per se* but merely to scratch the surface of the potential of the proposed model.

Future work points towards two directions. On the one hand, we plan to define benchmark scenarios which can be used to

provide a leveled comparison field for different protocols. Such benchmark scenarios can also be used to evaluate protocol stacks other than the established TCP/IP suite. On the other hand, we aim at incorporating the insights gained using the model into energy conserving protocols particularly designed for battery-powered mobile devices.

ACKNOWLEDGMENT

This work was supported by the Commission of the European Union as part of the IST FP7 Integrated Project 4WARD under grant agreement no. 216041, and by TEKES as part of the Future Internet program of TIVIT (Finnish Strategic Centre for Science, Technology and Innovation in the field of ICT). The views expressed are solely those of the author and do not necessarily represent the views of VTT, the respective projects or sponsors.

REFERENCES

- [1] T. Martin and D. Siewiorek, “A power metric for mobile systems,” in *Proc. ISLPED*, Monterey, California, USA, August 1996, pp. 37–42.
- [2] R. Krashinsky and H. Balakrishnan, “Minimizing energy for wireless web access with bounded slowdown,” *Wireless Networks*, vol. 11, no. 1, pp. 135–148, 2005.
- [3] S. Mann, *Programming Applications with the Wireless Application Protocol: The Complete Developer’s Guide*. New York, NY: John Wiley & Sons, 1999.
- [4] K. Pentikousis, “TCP in wired-cum-wireless environments,” *IEEE Commun. Surveys Tuts.*, vol. 3, no. 4, pp. 2–14, Dec. 2000.
- [5] J. Pinola and K. Pentikousis, “Mobile WiMAX,” *The Internet Protocol Journal*, vol. 11, no. 2, pp. 19–35, June 2008.
- [6] K. Pentikousis and H. Badr, “An evaluation of TCP with Explicit Congestion Notification,” *Annals of Telecommunications*, vol. 59, no. 1-2, pp. 170–198, January-February 2004.
- [7] R. Stewart and C. Metz, “SCTP: New transport protocol for TCP/IP,” *IEEE Internet Comput.*, vol. 5, no. 6, pp. 64–69, 2001.
- [8] S. Fu and M. Atiquzzaman, “SCTP: State of the art in research, products, and technical challenges,” *IEEE Commun. Mag.*, vol. 42, no. 4, pp. 64–76, 2004.
- [9] M. Stemm and R. H. Katz, “Measuring and reducing energy consumption of network interfaces in hand-held devices,” *IEICE Transactions on Communications*, vol. E80-B, no. 8, pp. 1125–1131, 1997.
- [10] B. McFarland and M. Wong, “The family dynamics of 802.11,” *ACM Queue*, vol. 1, no. 3, pp. 28–38, May 2003.
- [11] S. Agrawal and S. Singh, “An experimental study of TCP’s energy consumption over a wireless link,” in *Proc. 4th European Personal Mobile Communications Conference*, Vienna, Austria, February 2001.
- [12] D. Qiao, S. Choi, A. Jain, and K. G. Shin, “MiSer: An optimal low-energy transmission strategy for IEEE 802.11 a/h,” in *Proc. MOBICOM*, San Diego, CA, USA, September 2003, pp. 161–175.
- [13] M. Zorzi and R. R. Rao, “Energy efficiency of TCP in a local wireless environment,” *MONET*, vol. 6, no. 3, pp. 265–278, 2001.
- [14] —, “Perspectives on the impact of error statistics on protocols for wireless networks,” *IEEE Personal Commun. Mag.*, vol. 6, no. 5, pp. 32–40, October 1999.
- [15] H. Singh, S. Saxena, and S. Singh, “Energy consumption of TCP in ad hoc networks,” *Wireless Networks*, vol. 10, no. 5, pp. 531–542, September 2004.
- [16] W. Bokyung and S. Singh, “Computational energy cost of TCP,” in *Proc. IEEE INFOCOM 2004*, vol. 2, Hong Kong, March 2004, pp. 785–795.
- [17] G. Anastasi, M. Conti, E. Gregori, and A. Passarella, “Performance comparison of power-saving strategies for mobile web access,” *Performance Evaluation*, vol. 53, no. 3-4, pp. 273–294, August 2003.
- [18] —, “A performance study of power-saving policies for Wi-Fi hotspots,” *Computer Networks*, vol. 45, no. 3, pp. 295–318, June 2004.
- [19] C.-F. Chiasserini and R. R. Rao, “Energy efficient battery management,” *IEEE J. Sel. Areas Commun.*, vol. 19, no. 7, pp. 1235–1245, July 2001.
- [20] A. Anbuky and P. Pascoe, “VLRA battery state-of-charge estimation in telecommunication power systems,” *IEEE Trans. Ind. Electron.*, vol. 47, no. 3, pp. 565–573, June 2000.