Journal of Computer Networks and Communications

Energy-Efficient Wireless Communications with Future Networks and Diverse Devices

Guest Editors: Runhua Chen, Lingjia Liu, Krishna Sayana, and Hongxiang Li

Energy-Efficient Wireless Communications with Future Networks and Diverse Devices

Journal of Computer Networks and Communications

Energy-Efficient Wireless Communications with Future Networks and Diverse Devices

Guest Editors: Runhua Chen, Lingjia Liu, Krishna Sayana, and Hongxiang Li

Editorial Board

Contents

Hindawi Publishing Corporation
Journal of Computer Networks and Communications
Volume 2013, Article ID 897029, 2 pages
http://dx.doi.org/10.1155/2013/897029

Editorial

Energy-Efficient Wireless Communications with Future Networks and Diverse Devices

Runhua Chen,[1] Lingjia Liu,[2] Krishna Sayana,[3] and Hongxiang Li[4]

[1] *Wireless Base Station Infrastructure, Texas Instruments Incorporated, Dallas, TX 75243, USA*
[2] *Department of Electrical Engineering and Computer Science, The University of Kansas, Lawrence, KS 66045, USA*
[3] *Standards and Research Lab, Motorola Mobility Incorporated, Libertyville, IL 60048, USA*
[4] *Department of Electrical and Computer Engineering, University of Louisville, Louisville, KY 40292, USA*

Correspondence should be addressed to Runhua Chen; r-chen@ti.com

Received 11 February 2013; Accepted 11 February 2013

Telecommunication networks are responsible for approximately 2% of the global energy consumption. As data traffic continues to grow due to the proliferation of smart phone devices, reducing greenhouse emission by improving the network energy efficiency is becoming increasingly important. Such improvements may be achieved by advanced system design at both the network and the mobile terminal sides, although the majority of savings may come from the infrastructure. For instance, one promising technique for cellular network power reduction is through advanced radio resource management (RRM) to dynamically switch ON or OFF a basestation based on daily traffic variation. By switching OFF lightly loaded basestation and offloading the traffic to neighboring basestations, significant energy reduction may be possible without compromising the network performance.

Improving energy efficiency also needs to consider future networks shifting towards more diversified services. Exemplary application scenarios include smart metering, wireless sensor monitoring, and machine-to-machine communication. These scenarios introduce new traffic behaviors in the network, and due to the smaller packet sizes and the extremely large number of devices, improving energy efficiency of the network while reducing device cost is vital. There is a need for energy-efficient solutions to enable new smart and energy-efficient devices to simultaneously share the network with relatively fewer numbers of conventional devices that have much larger data and energy consumption demands.

The goal of this special issue is to give a comprehensive overview of the ongoing research on energy-efficient wireless communication for future networks and devices, in areas including energy-efficient communication, smart metering, energy harvesting, energy conservation with network coordination, and machine-type communication. We hope that this will motivate interested researchers to explore and propose new ideas in these growingly important areas. The special issue begins with a paper coauthored by A. Shareef and Y. Zhu discussing a novel evaluation platform for assessing energy efficiency of wireless sensor networks. Such a model is important as it allows for quick and accurate evaluations of different energy consumption algorithms, protocols, and system designs. In particular, the article proposes a Petri net model-based platform, which is analytically and numerically shown to outperform the Markov model and programmed simulation in terms of evaluation accuracy, with low construct and test simplicity.

In the article coauthored by A. Chehri and H. Moutfah, the researchers go into more details on improving the energy consumption efficiency of wireless sensor networks. As sensor nodes are usually powered by batteries or other low-energy power resources, optimizing the system performance with power constraint is of practical importance. For that end, the authors propose a green routing protocol relying on adaptive modulation and power control, which is shown to improve the energy efficiency of sensor networks without compromising the QoS constraint of delay and error rate.

The paper coauthored by G. Chen et al. investigates a new routing algorithm for wireless sensor networks (WSNs) based on principal component analysis (PCA). By disclosing the connection between PCA and K-means, the authors design a clustering algorithm which efficiently develops a clustering structure in WSNs. Moreover, as a compression method, the paper demonstrates that PCA technique can also be used in data aggregation for WSNs. It establishes the explicit procedure of PCA-guided routing algorithm for WSNs by incorporating PCA technique into both the data aggregating and routing processes. The advantages of the proposed algorithm are demonstrated by theoretical analyses and simulation results. The simulation results show that the PCA-guided routing algorithm significantly reduces the energy consumption, prolongs the lifetime of network, and improves network throughput.

In the following article, A. Silva et al. study new power management techniques to extend as much as possible the lifetime of primary cells (nonrechargeable batteries). By assuming low duty-cycle applications, three power-management techniques are combined in a novel way to provide an efficient energy solution for wireless sensor networks nodes or similar communication devices powered by primary cells. Accordingly, a customized node is designed, and long-term experiments in laboratory and outdoors are realized. Simulated and empirical results show that the battery lifetime can be drastically enhanced. Unattended nodes deployed in outdoors under extreme temperatures, buried sensors (underground communication), and nodes embedded in the structure of buildings, bridges, and roads are some of the target scenarios for this work. Some of the provided guidelines can be used to extend the battery lifetime of communication devices in general.

M. N. Alam et al. investigate the energy consumption for the IEEE 802.11s link specific Power Saving Mode (PSM) for peer link operation. The study is further extended to a multihop network consisting of eight STAs. They conclude that at the cost of increased packet delay, the IEEE 802.11s PSM operation not only provides significant energy savings, but also provides almost the same throughput that the active mode operation offers. For a large network, the energy saving could be as high as eighty percent when compared with active mode operation. In the presence of hidden node especially, the PSM can perform much better than an active mode, if the nodes avoid simultaneous operation. Further it is also shown that, by switching to active mode, receiving STAs can reduce the link delay considerably, which points to a clear tradeoff of delay, throughput and energy consumption. The analysis provided can be used to investigate the critical parameters in PSM operation in an 802.11 network.

The paper coauthored by K. Dhondge et al. introduces an energy-efficient collaborative and opportunistic positioning system (ECOPS) for heterogeneous mobile devices. In particular, ECOPS-facilitates mobile devices with estimated locations using Wi-Fi in collaboration with a few available GPS broadcasting devices, in order to achieve high-energy efficiency and accuracy within available energy budget constraints. ECOPS estimates the location using heterogeneous positioning services and the combination methods including a received signal strength indicator, 2D trilateration, and available power measurement of mobile devices. The evaluation conducted by the authors shows that ECOPS significantly reduces energy consumption and achieves good accuracy of a location.

Energy consumption reduction in the receiver chain by limiting the use of the equalizer is investigated by S. Bourbia et al. This goal is achieved by making the radio receiver aware of its environment and being able to take decision to turn on or off the equalizer according to its necessity or not. When the equalizer is off, the computational complexity is reduced, and the rate of reduction depends on the percentage of time during which the component is disabled. In order to adapt the use of the equalizer, the authors developed a decision-making technique that provides to the receiver the capacities of awareness and adaptability to the state of its environment. A technique based on a statistical modeling of the environment is introduced by defining two metrics as channel quality indicators to evaluate the effects of the intersymbol interferences and the channel fading. The statistical modeling technique allows the authors to take into account the impact of the uncertainties of the estimated metrics on the decision-making process.

Acknowledgments

The guest editors wish to take the opportunity to thank all the authors who have submitted their papers. We also wish to extend our thanks to the anonymous reviewers for their high-quality reviews which tremendously helped improve both the content and presentation of the papers in this special issue. Last but not least, we would like to thank the editorial board members for their valuable and insightful comments and guidance through the preparation of this special issue.

Runhua Chen
Lingjia Liu
Krishna Sayana
Hongxiang Li

Hindawi Publishing Corporation
Journal of Computer Networks and Communications
Volume 2013, Article ID 136213, 13 pages
http://dx.doi.org/10.1155/2013/136213

Research Article

ECOPS: Energy-Efficient Collaborative Opportunistic Positioning for Heterogeneous Mobile Devices

Kaustubh Dhondge,[1] Hyungbae Park,[1] Baek-Young Choi,[1] and Sejun Song[2]

[1] University of Missouri-Kansas City, 546 Flarsheim Hall, 5110 Rockhill Road, Kansas City, MO 64110, USA
[2] Texas A&M University, Fermier Hall 008, 3367 TAMU, College Station, TX 77843, USA

Correspondence should be addressed to Kaustubh Dhondge; kaustubh.dhondge@mail.umkc.edu

Received 16 June 2012; Revised 12 October 2012; Accepted 19 November 2012

Academic Editor: Lingjia Liu

The fast growing popularity of smartphones and tablets enables us to use various intelligent mobile applications. As many of those applications require position information, smart mobile devices provide positioning methods such as Global Positioning System (GPS), WiFi-based positioning system (WPS), or Cell-ID-based positioning service. However, those positioning methods have different characteristics of energy-efficiency, accuracy, and service availability. In this paper, we present an Energy-Efficient Collaborative and Opportunistic Positioning System (ECOPS) for heterogeneous mobile devices. ECOPS facilitates a collaborative environment where many mobile devices can opportunistically receive position information over energy-efficient and prevalent WiFi, broadcasted from a few other devices in the communication range. The position-broadcasting devices in ECOPS have sufficient battery power and up-to-date location information obtained from accurate but energy-inefficient GPS. A position receiver in ECOPS estimates its location using a combination of methods including received signal strength indicators and 2D trilateration. Our field experiments show that ECOPS significantly reduces the total energy consumption of devices while achieving an acceptable level of location accuracy. ECOPS can be especially useful for unique resource scarce, infrastructureless, and mission critical scenarios such as battlefields, border patrol, mountaineering expeditions, and disaster area assistance.

1. Introduction

Smart mobile devices such as smartphones and tablets are rapidly becoming prevalent in our lives. They have spurred a paradigm shift from traditional restricted phone applications to intelligent mobile applications such as location-based, context-aware, and situation-aware services. For example, a social-network-based traffic information system [1] allows each mobile user to report and use real-time traffic information, in addition to the archived traffic information from the US Department of Transportation.

As many of those application services require position information, smart mobile devices provide various positioning services via Global Positioning System (GPS) [2], WiFi-based positioning system (WPS) [3], or Cell-ID Positioning [4]. Being dedicated equipment for positioning, GPS becomes available for many smart devices as an additional feature and is considered to be an accurate and preferred method for location-based services (LBSs) [5, 6]. However, its high energy consumption, due to the Time To First Fix (TTFF), becomes a significant drawback. WPS approximates a position from the location information of a nearby wireless access point (AP) that is stored in the database. Its energy efficiency is much better than GPS, and the accuracy is moderate. As WiFi is a de facto standard in wireless local area network (WLAN) communication, it is broadly available on most smart devices. However, the service is limited to indoor or urban areas where the access points are densely populated. Cell-ID Positioning provides an approximate location from the serving cell tower, where a cell area range is around 100 ~ 500 m in urban areas, but it can span up to 10 Km for rural areas. Although this is the most power saving approach, due to a large error range caused by the coarse cell tower density, Cell-ID Positioning cannot offer the utility of most LBS applications. In addition, mobile devices such as the WiFi version of tablets are not fully equipped with 3G/4G data chips at the present time even though 3G and 4G wireless networks provide enough bandwidth to enable

Table 1: Characterization of various positioning methods.

Positioning method	Accuracy	Energy efficiency	Equipment availability	Service limitations
GPS	High (within 10 m)	Low	Low	Indoor and canyon
WPS	Moderate (within 50 m)	Moderate	High	Coarse AP density area
Cell-ID Positioning	Low (within 5 Km)	High	Moderate	Rural area

explicit support for real-time LBS. We have summarized the characteristics of positioning methods ([7]) in Table 1.

Energy efficiency while maintaining required accuracy for the given service limitations is one of the most critical issues in mobile devices, due to limited battery life and the high energy consumption of applications. As different positioning methods available on a mobile device have different characteristics with respect to accuracy, energy-efficiency, and service availability, there have been several proposals for dynamic selection of a positioning method on an individual device. For example, [8] uses an accelerometer for movement detection to power cycle GPS, if the device is not mobile. However, the effectiveness of most of the existing heuristics is limited by equipment constraints or service availability, as the applications choose a preferred positioning method that is available within an individual device.

In this paper, we propose ECOPS to facilitate a WiFi hotspot mode [9] or WiFi Direct mode [10] based approximation in collaboration with a few available GPS broadcasting devices under budget constraints. ECOPS is a collaborative positioning method between WiFi and GPS mobile devices, in addition to a positioning method selection heuristic within a mobile device. It can achieve moderate accuracy with low energy usage. Although there is a previous collaborative work [8] that pairs two devices via Bluetooth to save GPS power cycle, the approach needs both GPS and Bluetooth on both devices. Instead, ECOPS supports heterogeneous methods among mobile devices. There are many mobile devices including the majority of current tablets that only support a basic wireless communication method which is WiFi. The WiFi-only device can obtain position information from a GPS device with ECOPS. This proposed system can operate opportunistically, where each device can resolve the location via various available methods including trilateration [11] with three GPS broadcasting devices and a received signal strength indicator (RSSI) [12] or approximation with geomagnetic sensors [13] and a single GPS device without requiring any WiFi AP.

We implemented ECOPS using Android-powered mobile devices such as smartphones and tablets. The evaluation results show that ECOPS significantly saves the total energy consumption of the devices while achieving a good level of location accuracy. In addition, it enables constrained devices to enjoy location-based services that would otherwise not be possible.

The rest of the paper is organized as follows. Potential application scenarios are described in Section 2. Section 3 discusses the existing and state-of-the-art techniques. A detailed explanation of the proposed system is presented in Section 4. The performance evaluations and experimental scenarios are explained in Section 5. Finally, we conclude the paper in Section 6.

2. Application Scenarios

While security and social incentive issues are not in the scope of this paper, the proposed opportunistic and collaborative positioning scheme can be especially useful for unique resource scarce and mission critical applications. Such examples include border patrol, battlefields, mountaineering expeditions, and disaster area assistance.

For example, suppose a team of border patrol officers is searching for an illegal immigrant in the border area. In some areas of rigid terrain, GPS and cellular signals can be lost in a canyon. Some projects [14] employ a low-altitude tethered aerostat to set up a temporary WiFi hotspot. To help with positioning, a few officers stay at the top of the valley to relay their GPS position information to the officers searching down in the valley. Such a collaborative positioning is a natural application scenario of ECOPS.

In a battlefield scenario, when a platoon is air-dropped into a war zone, it is nearly impossible to find WPS services in the surroundings. Even with the availability of technology like LANdroids [15] to provide a network in such conditions, it is not a simple task. Also, it is crucial for soldiers to have accurate location information in the battlefield. In such a scenario, the capabilities of ECOPS can be exploited to maintain accurate location information while reducing overall energy consumption. Although one may not have a strong incentive to take a lead and offer location information for others, such concerns are lifted immediately if a leadership hierarchy preexists in the application scenario. For instance, when a platoon is being deployed in a battlefield, the platoon leader chooses to be the primary location broadcaster using ECOPS along with a few others at the top of the hierarchy. The other soldiers in the unit are able to estimate their location information based on the geocoordinates they receive from their unit's command. This will result in fewer devices from the unit querying satellites for location information and reduce the overall energy consumption. Extending the lifetime of devices during the operation is a mission critical parameter as the duration of an operation is not fixed and often tends to be longer than expected. Under the Battlefield Air Targeting Man Aided Knowledge (BATMAN) [16] project, the United States Air Force is actively seeking to equip their soldiers with modern Android-powered smartphones to obtain accurate location information with high energy efficiency. Modified versions of Android [17, 18] enable the desired level of security for military use. Such projects can benefit greatly by ECOPS.

Another scenario where ECOPS can be extremely useful is during natural calamities. In such cases, emergency

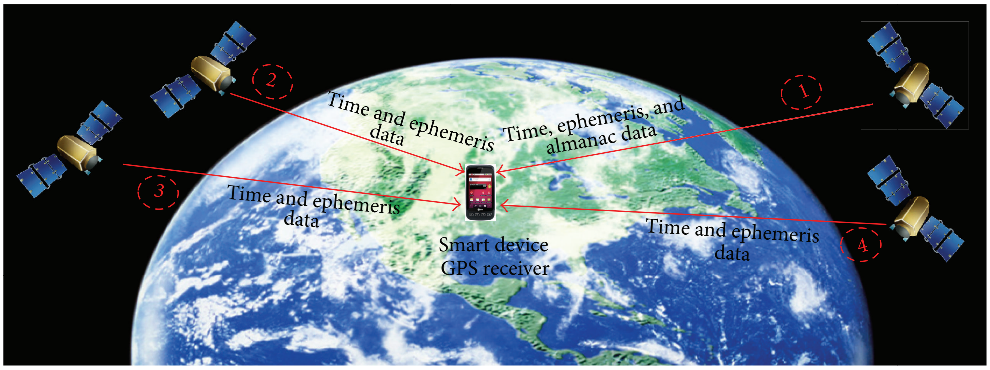


Figure 1: Illustration of Global Positioning System.

responders who are involved in search and rescue missions can host an ECOPS-based location broadcasting service over WiFi Direct. As they move around the area, victims can use their smart devices to either request assistance or transmit their locations.

3. Related Work

Positioning schemes on mobile devices have been a long standing topic of exploration. This resulted in three main positioning techniques using either the information provided by the GPS, WPS, or Cell-ID Positioning. Also, there have been several research proposals for specific environments.

3.1. Global Positioning System (GPS). The Global Positioning System (GPS) is a satellite navigation system that provides location and time information anywhere on earth with four or more GPS satellite signals. It is originally deployed and maintained by the United States government and is now freely accessible to anyone [2]. The GPS provides very high level of accuracy, but suffers from a high TTFF due to the large distance between the GPS receiver and serving satellites. This problem has been somewhat addressed by the use of assisted GPS (aGPS) that relies on the cellular or internet infrastructure to get a faster lock on the serving satellites while obtaining precise time information from the network.

Within the navigation message continuously broadcasted by each of the satellites in the constellation, the GPS receiver looks for three important pieces of data as illustrated in Figure 1. The first piece of data consists of the GPS date and time information. It additionally also consists of the health statistics of the satellite. The ephemeris data forms the second important piece and allows the GPS receiver to calculate the position of the satellite and is broadcasted every 30 seconds. The ephemeris data is valid for no longer than four hours. The third important piece is the almanac data which provides approximate information concerning the rest of the satellites. This data is transmitted over 12.5 minutes and is valid for a maximum of 180 days. The almanac data can be obtained from any satellite, and it enables the GPS receiver to determine which particular satellite to search for next. As the signal from the selected satellite becomes directly available, the GPS receiver then downloads the second important data, that is, the ephemeris data. It is absolutely necessary that the GPS receiver has the satellite's complete copy of the ephemeris data to determine its position. In case the signal is lost in the middle of acquiring this data, the GPS receiver will have to discard whatever data was downloaded and start searching for a new satellite signal.

Once the GPS receiver has ephemeris data directly from three or more satellites, it can carry out various methods to accurately determine its own location. These methods involve and are not restricted to 3D trilateration, Bancroft's method, and multidimensional Newton-Raphson calculations. Due to the high propagation delays, getting the ephemeris and almanac data can take up to 15 minutes for a device just out of the factory, and then around ~20 seconds after the initial configuration. To expedite this process, some GPS receivers can use multiple channels for faster fixes. Another strategy is to obtain the ephemeris and almanac data from a faster network like the cellular network or the internet as in the case of a GPS.

3.2. WiFi-Based Positioning System (WPS). WPS maintains an extensive database of WiFi access points (APs) along with their geographic locations [19, 20]. This information has to be collected painstakingly over a large duration of time and is vulnerable to changes in the location of APs or the discontinuation of their service. The information of APs-SSID and geographic location can be collected manually or in a more automated way by retrieving the GPS location of smart devices connected to the AP and associating that information with the AP. Once such a large and dedicated database is ready and a device is in the vicinity of an AP, or several APs, it can provide the RSSI values and the SSID of the APs to the WPS servers. The WPS servers, based on proprietary techniques, apply filtering approaches and trilateration techniques to this data and determine the accurate location of the smart device. This geographic location information is then relayed back to the smart device which can exploit it for various LBSs. The illustration of WPS is depicted in Figure 2.

While the WPS service approaches work well in terms of energy efficiency [21–23], they are not globally available for users. A solution leveraging the existing infrastructure, such as APs without requiring any specialized infrastructures for localization, has been proposed in [24]. However, since this

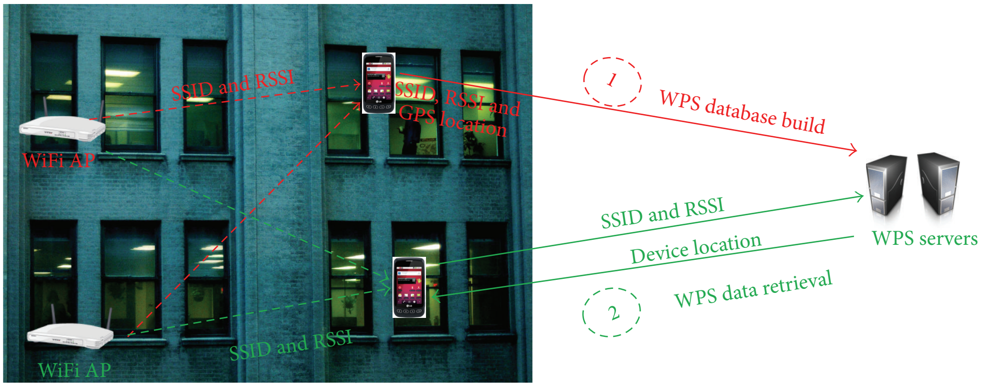


Figure 2: Illustration of WiFi positioning system.

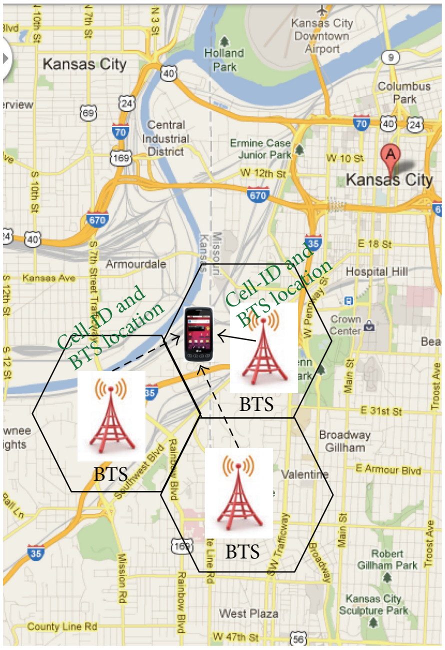


Figure 3: Illustration of Cell-ID Positioning.

localization scheme is limited to the indoors and still relies on infrastructure, such as APs, it cannot be useful outdoors where the WiFi signals are neither dense enough nor covered.

3.3. Cell-ID Positioning. In Cell-ID Positioning, a mobile device obtains its position from the geographic location of its associated base transreceiver station (BTS), with an error range proportional to the signal strength within a cell. The mobile device can estimate its location as the BTS periodically broadcasts its Cell-ID along with its location. Once this information is available to the mobile device, it can use the location of the BTS as its own location with the error calculated using the propagation model. Another technique that may be used for cell phones to estimate their location is to observe the delay in receiving a special message broadcasted by the BTS from the time it was transmitted. This information is used by the mobile device to estimate its distance from the BTS.

Note that a cell size can be very large especially in rural areas and highways where the density of cellular towers is very low. One cellular tower is often capable for serving up to 5 Km radius. As this large cell size leads to a significant error range, other nearby cell tower signals may be used in order to improve the accuracy [4, 25]. Such approaches also exploit the fading phenomenon independently or along with predictive techniques to improve the accuracy of Cell-ID Positioning. However, the accuracy is still limited as the propagation model needed for the trilateration does not work well, due to complex signal fading behavior over long distances. The illustration of Cell-ID Positioning is depicted in Figure 3.

3.4. Recent Research Proposals. While most of commercial approaches heavily depend on infrastructures [3, 26], or use extra high-end sensors and exploit the available information from an *individual* device [26–28], research proposals mostly aim to improve the positioning accuracy or energy efficiency through algorithmic approaches [8, 23, 24, 29–33].

The work in [31–33] attempt to learn a known location from a training phase for a better location accuracy.

```
check the residual power ($p_r$);
if GPS-equipped device & $p_r \geq p_{\min}$ then
    device becomes PB and activates WiFi hotspot;
    executes CollaborativePB();
else if non-GPS device || $p_r < p_{\min}$ then
    device becomes PR and executes CollaborativePR();
end if
```

Algorithm 1: ECOPS: Initial procedure deciding whether a device becomes either a PB or a PR.

The authors of [33] employ indoor positioning, and perform fingerprinting and training of the known space using multiple sensors in a smartphone such as WiFi radio, cellular communications radio, accelerometer, and magnetometer. In order to improve Cell-ID location accuracy in low-end cell phones where neighboring cell tower information is not available, [32] uses RSSI from only the associated cell tower and leverages the signal strength history to estimate the location. The Cell-ID Aided Positioning System (CAPS) [31] relies on the continuous mobility and position history of a user to obtain better location accuracy over a basic cell tower-based approach. It uses Cell-ID sequence matching to estimate current position based on the history of Cell-ID and GPS position sequences that match the current Cell-ID sequence. CAPS assumes that the user moves on the same routes repeatedly and has the same cellular chip and infrastructure availability.

A few studies address energy efficiency of smartphones using power duty cycling techniques [7, 8, 31] that use a combination of the basic positioning techniques in a smartphone. The authors of [7] use different positioning schemes depending on the condition, for the purpose of target tracking. In the scheme, energy-efficient but inaccurate Cell-ID Positioning or WPS is used when the target is distant, while accurate but energy-inefficient GPS is used when the target is close. The rate-adaptive positioning system (RAPS) [8] uses built-in sensors in a smartphone to determine if the phone has moved beyond a certain threshold and decides whether to turn on the GPS or not. RAPS also stores the space-time history of the user's movements to estimate how to yield high energy efficiency. Another idea presented by the authors involves a Bluetooth-based position synchronization (BPS) in which devices share their location information over a Bluetooth connection. While a Bluetooth connection consumes less power as compared to a WiFi ad hoc, it also limits the range of communication to less than 10 m. Our work has advantages over the basic BPS technique in several aspects. Not only does a WiFi ad hoc mode give us a better range, but we have also taken into account the heterogeneity amongst the devices in terms of availability of a GPS chip or cellular connection. We expect all the devices to have at least a WiFi module present onboard. In BPS, once location information is obtained from a neighboring device, only a fixed error range of 10 m (e.g., same as the range of a typical Bluetooth) is associated with that information. However, in ECOPS we exploit the RSSI values of the connection to determine the accurate distance between the two devices, and when three or more location transmitting devices are available, the trilateration technique achieves pinpoint locating capabilities.

The work in [34] proposed to use minimal auxiliary sound hardware for acoustic ranging in order to improve the accuracy. The acoustic ranging technique estimates the distance among peer phones, then maps their locations jointly against a WiFi signature map subject to ranging constraints. It is a WPS augmentation technique to improve the accuracy over a pure WPS.

Our approach is unique in that we use a collaborative approach rather than focusing on the information in an individual device and do not rely on any special hardware or infrastructure such as WPS or Cell-ID Positioning. Note that we only use a small amount of GPS information and the WiFi ad hoc mode of mobile nodes. ECOPS is specifically aimed at resource constrained environments such as battlefields where GPS is the only available positioning infrastructure, and WiFi ad hoc mode is readily available in most mobile devices while allowing good network range (up to 100 m). Beside controlling energy usage and location accuracy, we allow to use heterogeneous mobile device types.

4. ECOPS Approach

In this section, we discuss ECOPS algorithms in detail. Figure 4 shows an ECOPS deployment example. It consists of mobile devices with heterogeneous positioning methods available such as GPS, WiFi, and Cell-ID. These devices virtually establish an ad hoc network using WiFi to build a collaborative positioning environment. In the established ECOPS ad hoc network, a device may function as either a position broadcaster (PB) or a position receiver (PR). A GPS-equipped device with sufficient battery life and up-to-date location information becomes a candidate for a PB. Other devices with no GPS that need current location information will become PRs.

Three algorithms are presented for the overall operation of the ECOPS. Algorithm 1 describes the initial procedure deciding whether a device becomes a PB or a PR. After the initial decision, Algorithms 2 and 3 depict how the devices in ECOPS collaboratively maintain their most updated location information as a PB or a PR, respectively. For GPS-equipped devices, the role of the devices can be changed during their operation according to their residual energy level (i.e., PB ↔ PR). As illustrated in Algorithm 1, a device, once it starts ECOPS operation, will check the time elapsed since the device got the location information (t_e) and residual power

```
while $p_r \geq p_{\min}$ do
    listen to connection request from a PR;
    wait for location request from a PR;
    if PR requests then
        check the time elapsed since the device got location information ($t_e$)
        calculate $I_{\text{decay}}$;
        if $I_{\text{decay}} < \alpha$ then
            update its GPS location information;
            $t_e = 0$;
        end if
    end if
    broadcast current GPS location information;
    check the residual power ($p_r$);
end while
execute CollaborativePR();
```

Algorithm 2: ECOPS: *CollaborativePB*().

```
while non-GPS device || (GPS-equipped device & $p_r < p_{\min}$) do
    sleep until the device needs to update location information
    numofPBs = 0;
    check the list of available PBs;
    make connection to each PB and request GPS location information sequentially;
    calculate the distance to each PB using the obtained RSSI value;
    set numofPBs to the number of the detected PBs
    if numofPBs == 1|| one of PBs is within the near field threshold ($\beta$ meters) then
        use the received GPS location information immediately without trilateration;
        continue;
    end if
    if numofPBs == 2 then
        calculate two possible locations (PR) and get the middle location between the two possible locations;
    end if
    if numofPBs >= 3 then
        select three PBs randomly and calculate the its current location (PR) with the GPS coordinates and distance information of the selected PBs;
    end if
    if GPS-equipped device then
        check the residual power ($p_r$);
    end if
end while
execute CollaborativePB();
```

Algorithm 3: ECOPS: *CollaborativePR*().

(p_r) to see if it is qualified for being a PB. Since we are looking for the devices that have the most recent location information with enough residual power, the device with the conditions such as $p_r \geq p_{\min}$ and $I_{\text{decay}} \geq \alpha$ can be a PB, where $p_{\min}$ is the minimum residual energy that a PB has to maintain and I_{decay} is the level of the validity with respect to time for the location information, defined by

$$I_{\text{decay}} = 100 \times \left(1 - \frac{t_e}{t_d}\right), \qquad (1)$$

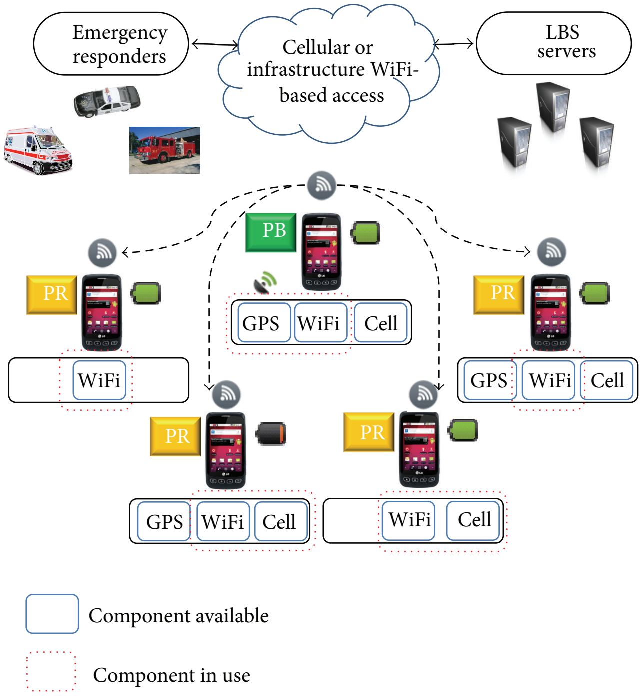


FIGURE 4: ECOPS deployment example.

where t_d is the maximum time in which the location information is considered to be valid.

If a device is equipped with a GPS receiver and satisfies the $p_{\min}$, it can be a PB. Once it becomes a PB, it will start its WiFi hotspot mode and serve the most up-to-date location information to a PR when a PR requests the location information. An Android device cannot use the WiFi Internet service while it is in the WiFi hotspot mode. However, with (Android 4.0), WiFi Direct technology can be used for PBs. In a PB mode with WiFi Direct, the users can enjoy their WiFi Internet service and provide the most up-to-date location information simultaneously. The device without a GPS receiver will automatically be a PR once it enters ECOPS, and then search for PBs around it.

As shown in Algorithm 2, once the device enters the PB mode, it plays a role of the PB while it satisfies the $p_{\min}$ constraint. The threshold α is a system parameter that can be varied according to the requirement of applications. The tradeoff between location accuracy and energy consumption can be adjustable using α. An application requiring high accuracy will select a small amount of α, but a high value of α is used for applications requiring low energy consumption. The PB will check I_{decay} to see if the current location information is adequate (e.g., $I_{\text{decay}} \geq \alpha$) before broadcasting it.

In Algorithm 3, a PR will collect the possible number of GPS coordinates and corresponding RSSI values and apply opportunistic localization as illustrated in Figure 5. If a PR finds a PB within the threshold distance (β meters), then a PR uses the GPS coordinate from a PB as is. The parameter β is controllable and users of ECOPS can set it according to their preference. Once a PR estimates its location, it can become a PB. However, we do not use those cases in our experiments to avoid the additional errors that will be induced from PBs and focus on the PR's accuracy.

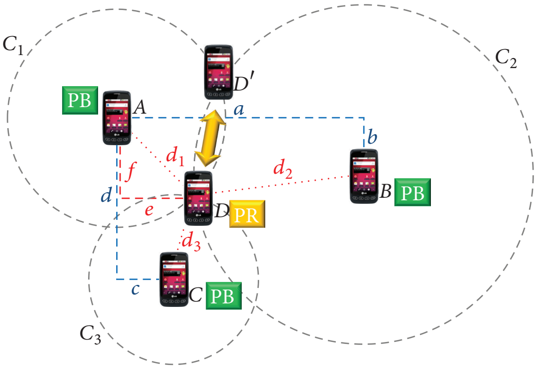


FIGURE 5: 2D trilateration.

ECOPS is opportunistic, meaning that getting the most updated location via trilateration is not limited by the number of available PBs. Supposing that there is only one GPS broadcaster in Figure 5, say node C, then the center of an error range of the circle C_3 will become the PR's approximated position. Another possible situation is when there are two PBs, say nodes A and B; then the middle point of two possible points, D and D', is selected as an approximated PR location. The accuracy of the estimated location will range from one point where the two circles intersect in the best case to the diameter of the smaller circle in the worst case, respectively. In order to get the most accurate location information for a PR, we need at least three PBs to provide their location information obtained from the GPS receiver along with the RSSI values, so that we can build an absolute coordinate system from the relative coordinate system. For example, in Figure 5, we calculate the distance parameters a, b, c, and d using the algorithm described in [35] in order to obtain the values of e and f. We convert the obtained distances e and f into the unit of the GPS coordinates to get the final calculated GPS coordinate. The distance between two GPS coordinates, $(\text{lat}_1, \text{lng}_1)$ of e and $(\text{lat}_2, \text{lng}_2)$ of f, is computed using the haversine formula that gives a spherical distance between two points from their longitudes and latitudes [35, 36]. The formula is described in (2) and the formula for the value of dist is shown in (3):

$$\begin{aligned} F_{\text{dist}}&(\text{lat}_1, \text{lng}_1, \text{lat}_2, \text{lng}_2) \\ &= \text{rad}\,2\ \deg\left(a\cos\left(\text{dist}\right)\right) \times 60 \times 1.1515 \\ &\quad \times 1.609344, \end{aligned} \tag{2}$$

$$\begin{aligned} \text{dist} = {}& \sin\left(\deg 2\,\text{rad}\left(\text{lat}_1\right)\right) \times \sin\left(\deg 2\,\text{rad}\left(\text{lat}_2\right)\right) \\ &+ \cos\left(\deg 2\,\text{rad}\left(\text{lat}_1\right)\right) \times \cos\left(\deg 2\,\text{rad}\left(\text{lat}_2\right)\right) \\ &\times \cos\left(\deg 2\,\text{rad}\left(\text{lng}_1 - \text{lng}_2\right)\right). \end{aligned} \tag{3}$$

Thus, we can calculate relative coordinates (a, b, c, and d) as follows:

$$a = F_{\text{dist}}\left(A_{\text{lat}}, A_{\text{lng}}, B_{\text{lat}}, A_{\text{lng}}\right),$$
$$b = F_{\text{dist}}\left(A_{\text{lat}}, A_{\text{lng}}, A_{\text{lat}}, B_{\text{lng}}\right),$$

$$c = F_{\text{dist}}\left(A_{\text{lat}}, A_{\text{lng}}, C_{\text{lat}}, A_{\text{lng}}\right),$$
$$d = F_{\text{dist}}\left(A_{\text{lat}}, A_{\text{lng}}, A_{\text{lat}}, C_{\text{lng}}\right). \tag{4}$$

The distances (d_1, d_2, and d_3) between node D and other nodes (A, B, and C) can be derived from the measured RSSI values of node D, using the formula from the path loss propagation model [37]:

$$\text{RSSI} = -\left(10n \cdot \log_{10} d + \mathscr{L}\right), \tag{5}$$

where RSSI is the received power which is a function of the distance between the transmitter and the receiver (T-R), n is the signal propagation constant (also called propagation exponent), d is the T-R separation distance in meters, and $\mathscr{L}$ is the system loss factor. Based on (5), we derived the distance (d) between two devices using the average RSSI value as follows:

$$d = 10^{(-\text{RSSI}-\mathscr{L})/10n}. \tag{6}$$

Now, the three circles in Figure 5 can be represented by the following, respectively:

$$C_1 : X^2 + Y^2 = {d_1}^2, \tag{7}$$

$$C_2 : (X-a)^2 + (Y-b)^2 = {d_2}^2, \tag{8}$$

$$C_3 : (X-c)^2 + (Y-d)^2 = {d_3}^2, \tag{9}$$

where the location of node A is set to $(0, 0)$.

We obtain the relative coordinate of PR node D from node A, (e, f), by calculating the point where these three circles intersect. In other words, we want to calculate the coordinate values of X and Y that simultaneously satisfy (7), (8), and (9). We first extend (8) and (9) as follows:

$$\begin{aligned} C_2 &: X^2 - 2aX + a^2 + Y^2 - 2bY + b^2 = {d_2}^2, \\ C_3 &: X^2 - 2cX + c^2 + Y^2 - 2dY + d^2 = {d_3}^2. \end{aligned} \tag{10}$$

By applying (7) into (10), the circles C_2 and C_3 can be written as

$$\begin{aligned} C_2 &: {d_1}^2 - 2aX + a^2 - 2bY + b^2 = {d_2}^2, \\ C_3 &: {d_1}^2 - 2cX + c^2 - 2dY + d^2 = {d_3}^2. \end{aligned} \tag{11}$$

Finally, the node D's coordinate that satisfies the three circles is attained by replacing X and Y with e and f, respectively, in (11). We can formulate the equations in terms of e and f as follows:

$$\begin{aligned} e &= \frac{d\left(D_{\text{vec}\,1}\right) - b\left(D_{\text{vec}\,2}\right)}{2\left(bc - ad\right)}, \\ f &= \frac{c\left(D_{\text{vec}\,1}\right) - a\left(D_{\text{vec}\,2}\right)}{2\left(ad - bc\right)}, \end{aligned} \tag{12}$$

where $D_{\text{vec}\,1} = {d_2}^2 - {d_1}^2 - a^2 - b^2$ and $D_{\text{vec}\,2} = {d_3}^2 - {d_1}^2 - c^2 - d^2$.

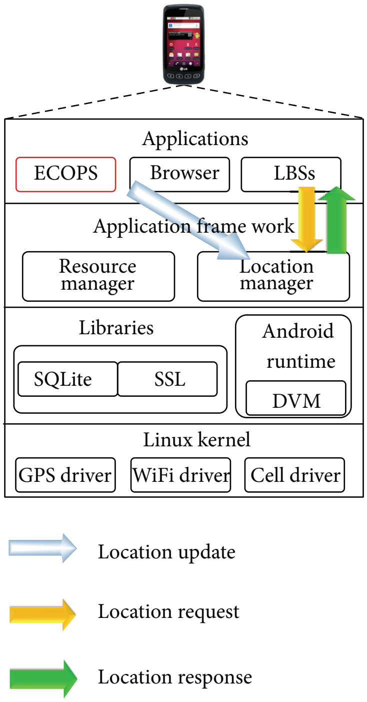


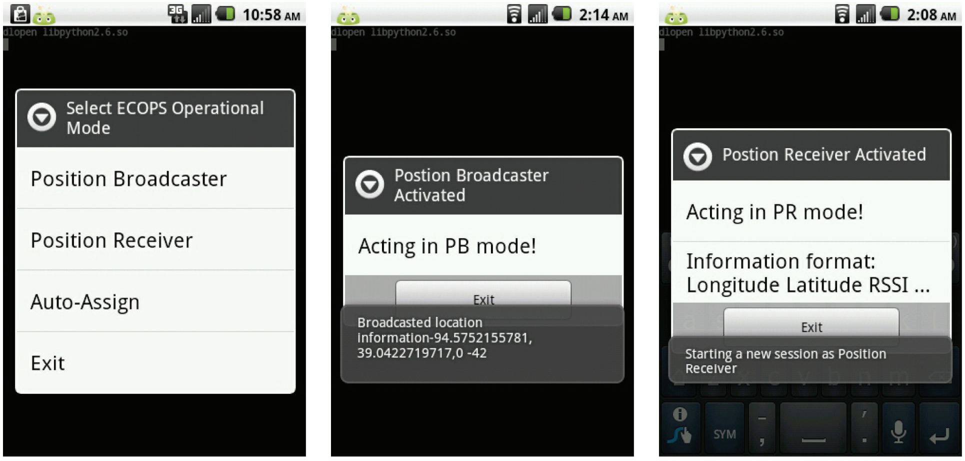
Figure 6: Android module architecture.

We have implemented ECOPS as an Android application for a feasibility test and analysis. Figure 7 shows the screenshots of the ECOPS application. Figure 7(a) displays the main screen that allows users to manually select an ECOPS device option either in PB mode or PR mode, or to request the selection automatically based on various parameters such as the remaining energy and sensor availabilities. Figure 7(b) presents a PB screen that lists the broadcasting location information. Figure 7(c) shows a PR screen that lists the received information and measured distance using the RSSI value. Although the current implementation is on an application level, as illustrated in Figure 6, it is still capable of making the received location information available to other application services. It will eventually be implemented within the application framework so that other applications can use the ECOPS services via APIs.

5. Experiments

In this section, we present the evaluation results of ECOPS in terms of energy efficiency and location accuracy. We have implemented an ECOPS Android application and used several Android smartphones including Samsung Galaxy Nexus S running Android version 2.3.6 and two LG Optimus V running Android version 2.2.2. We have turned the GPS off on some of the devices to mimic heterogeneous devices.

5.1. Validation of Smartphone GPS Accuracy and Propagation Model. As a first step, we test the accuracy of commodity GPS receivers on smartphones since they are not dedicated devices like the navigation devices for positioning. We have measured the accuracy of smartphones' GPS, by walking around



10:58 AM

dlopen libpython2.6.so

Select ECOPS Operational Mode

Position Broadcaster

Position Receiver

Auto-Assign

Exit

2:14 AM

dlopen libpython2.6.so

Postion Broadcaster Activated

Acting in PB mode!

Exit

Broadcasted location information-94.5752155781, 39.0422719717,0 -42

2:08 AM

dlopen libpython2.6.so

Postion Receiver Activated

Acting in PR mode!

Information format: Longitude Latitude RSSI ...

Exit

Starting a new session as Position Receiver

(a) Main menu

(b) PB mode

(c) PR mode

Figure 7: ECOPS screenshots.

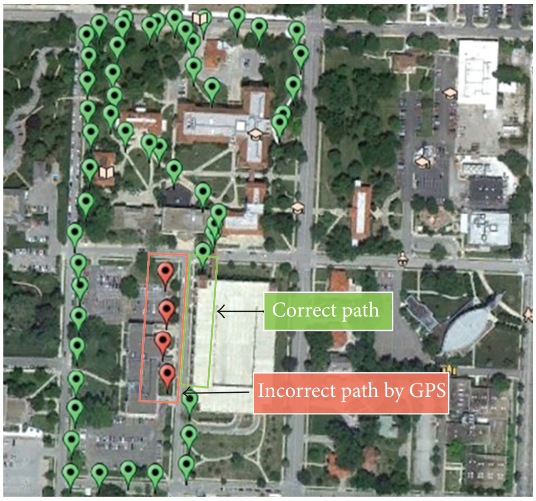


Figure 8: GPS trace obtained by smartphone.

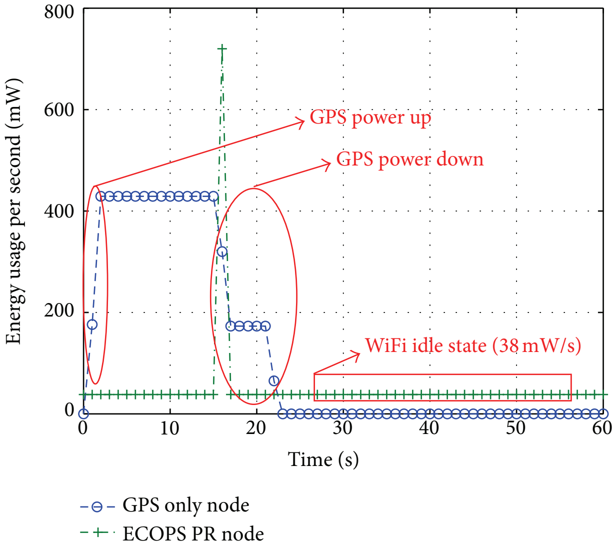


Figure 9: Energy usage: GPS versus ECOPS PR.

the Kansas City area while carrying three smartphones. As shown in Figure 8, the GPS collected locations are presented accurately except for a little error between tall buildings (~10 m).

Next, we validate the *path* loss model for correlation of the distance between a WiFi signal emitter and receiver with the measured RSSI values at the signal receiver for both indoor and outdoor environments. We compared the measured RSSI value with the theoretical RSSI value from the *path* loss model. As RSSI values often vary at each time of measurement for a given location, we used an averaged RSSI value with multiple samples (e.g., 1,000 samples within a few seconds). In Figures 10 and 11 we compare the measured RSSI with the theoretical RSSI while varying the distance between PB and PR; both inside a building and in outdoor environments are shown, respectively. The theoretical RSSI values are derived from (5). The dotted blue line shows the measured average RSSI values, and the solid green line represents the theoretical RSSI values at the corresponding distances. The system loss factor value (A) is set to 30. For the indoor environment, since we measured the RSSI between two devices while they were in the line-of-sight, we set the system loss factor (n) to 0.6. For the outdoor environment, we used $n = 1.9$. As evidenced in the figures, we observe that the *path* loss model works well for us in estimating the distance.

5.2. Energy Consumption. We now compare the total energy consumption of ECOPS devices to that of devices with GPS only scheme in various settings. First, we compare the energy consumption of a node that is ECOPS PR with a node using only GPS at per second granularity as illustrated in Figure 9. This power consumption profiling was done using PowerTutor [38]. The GPS uses 429 mW/s continuously once it is powered up and takes several seconds to power down which adds up to the energy consumption. Meanwhile, the WiFi module once powered up uses 720 mW/s in an

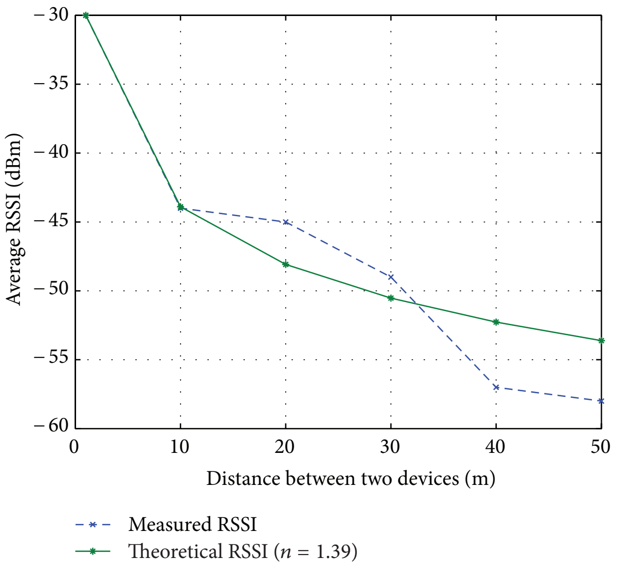


Figure 10: Measured RSSI (avg. of 1,000 samples) at various indoor spots.

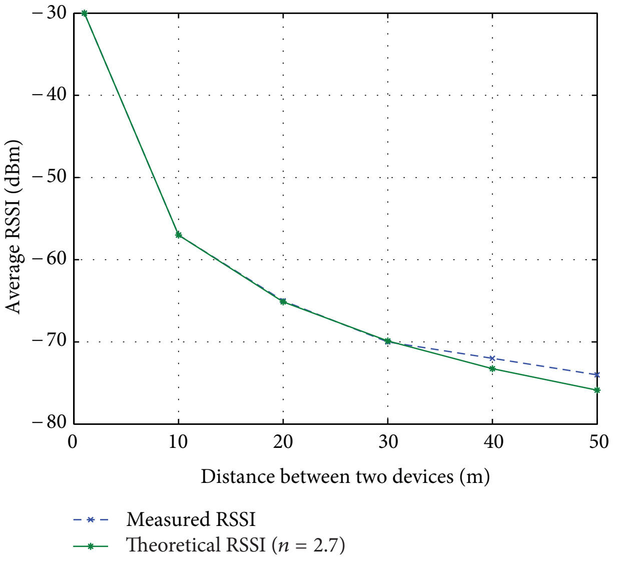


Figure 11: Measured RSSI (avg. of 1,000 samples) at various outdoor spots.

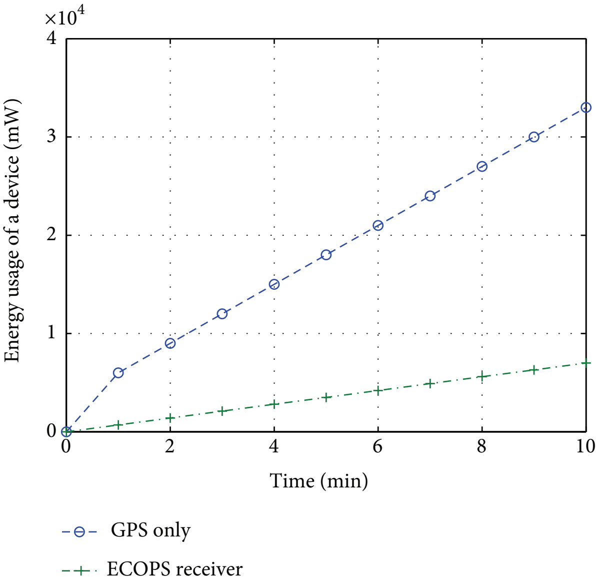


Figure 12: Comparison of individual node energy consumption: GPS versus ECOPS PR.

active state and 38 mW/s when in an idle state. During the experiment, for the same operational time of one minute, an ECOPS PR node uses only 3000 mW of energy in total whereas the GPS-only node uses 7432 mW of total energy. This clearly shows that an ECOPS PR is more energy efficient than a GPS-only node. These values are for an LG Optimus V model in particular, and similar for most smartphones.

Next, we contrast the energy consumption of a node that is ECOPS PR with a node using only GPS while varying the operational time with 1 minute increments as illustrated in Figure 12. We do this experiment to analyze the effectiveness of ECOPS over a duration of time. It shows that ECOPS is increasingly energy efficient with the elapsed time over the GPS only scheme.

In Figure 13, we compare the energy consumption of nodes that are ECOPS PR with nodes using only GPS while varying the number of devices in the network. We do this experiment to analyze the energy efficiency of ECOPS as the number of devices in the network scales. Note that for the ECOPS PR scheme, the PRs receive GPS data from three PBs and their energy consumption is accounted for in the results. The energy efficiency of ECOPS compared with the GPS only scheme is clear from Figure 13 and becomes increased substantially as the number of devices in the network scales.

5.3. Location Accuracy. In this section, we evaluate the location accuracy of ECOPS as compared to that of GPS, WPS, and Cell-ID Positioning. We tested ECOPS in a soccer field using four smartphones for the accuracy measurements. The soccer field was chosen, so that we have a clear and unhindered view of the sky, and in turn the experimental results are not influenced by the GPS position errors, and the ECOPS errors are precisely measured. We turned on GPS for three devices and turned it off for a device that acted as the PR. In order to measure the location accuracy, as illustrated in Figure 14, we placed the PR device at the center of the area and moved the other PB devices around multiple locations within the soccer field. The PR device computed its location using the measured RSSI values, and GPS coordinates from the PBs, and the trilateration technique described in Section 4.

We have moved the PBs to various places around the PR and recorded the PR's computed locations. As shown in Figure 15, we observed that ECOPS achieves a minimum error range of 2.32 m and a maximum error range of 33.31 m. While from Figure 16 we can observe that nearly 60% of these locations are within a 10 m error range and less than 10% have an error range greater than 15 m. The results represent that

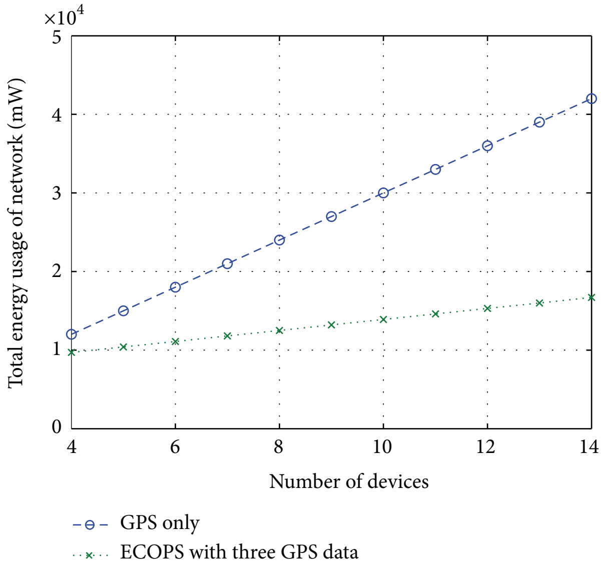


FIGURE 13: Comparison of total energy consumption of nodes (1 min): GPS versus ECOPS.

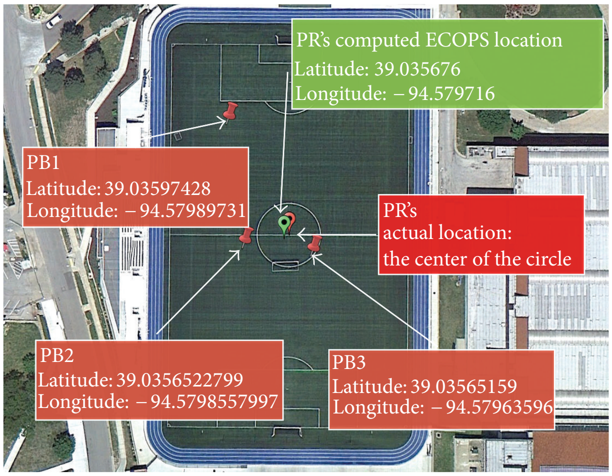


FIGURE 14: An example of field experiments.

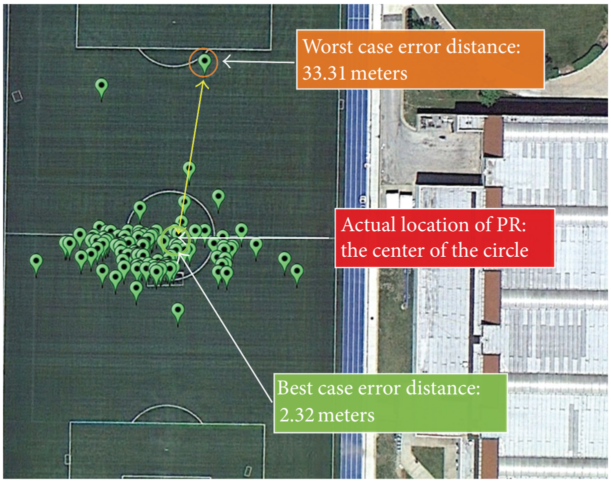


FIGURE 15: Experiment results: points calculated with three GPS coordinates and RSSI values.

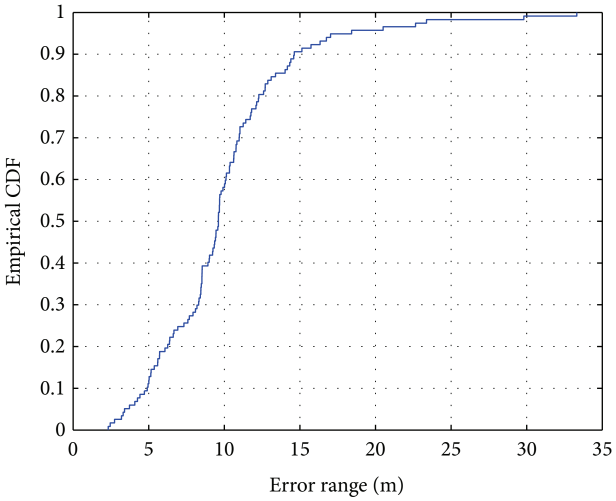


FIGURE 16: Distribution in error range for location estimated by PR.

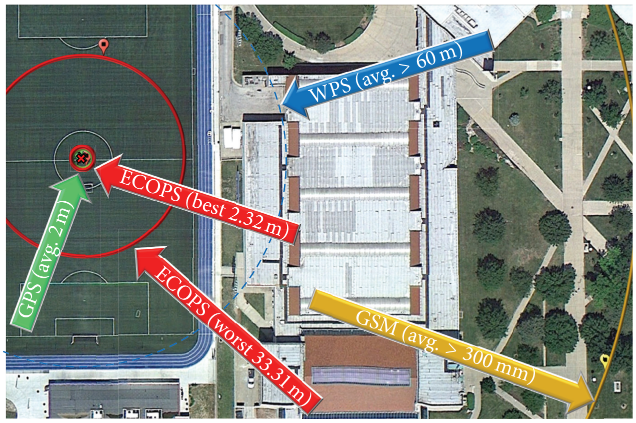


FIGURE 17: Accuracy comparison: ECOPS, GPS, WPS, and GSM-based positioning.

ECOPS can achieve a higher location accuracy than WPS while using less energy than GPS receivers. Also, note that the error ranges we observe here are an amalgamation of the general GPS receiver error from the PBs and the distance measurement error from the RSSI values.

Finally, we compare the errors of different positioning methods in Figure 17. As before, a smartphone that needs positioning is located at the center of soccer field. The network positioning API in Android obtains the location information either from WPS or Cell-ID Positioning. In order to ensure the Cell-ID Positioning in Android, the smartphone acquired the location from network positioning API while turning off the WiFi signal. The location information received was off by almost 300 m. Together with the location, it also suggested its own estimated error range of 1,280 m associated with it. Clearly, such information is too inaccurate to be used in most of LBS application scenarios. As for WPS, the smartphone obtained the location from Android network positioning API while turning off cellular signal. Note that WPS is not typically available in outdoor environment. Thus, we used an average WPS error from what we experimented at multiple locations in Kansas City area where WPS is available and found it to be 60 m. It is the dotted blue line in Figure 17. While the GPS-based location information proved to be the most accurate with an error range of about 2 m, ECOPS achieved the accuracy ranging from 2.32 m to 33.31 m. This is better than the performance of a WPS and fairly close to GPS accuracy while saving energy costs. This encourages us to comment that even when WPS service might be available, using ECOPS will facilitate a smartphone to receive more accurate location information at the same energy cost.

6. Conclusion

We have presented an Energy Efficient Collaborative Opportunistic Positioning System (ECOPS) for heterogeneous mobile devices. Unlike existing approaches that are seeking the best available positioning method from an individual device, ECOPS facilitates collaborative environments among a set of mobile devices, and thus mobile devices benefit from their neighboring devices. ECOPS supports heterogeneous devices to maximize energy-efficiency, as a device with only WiFi can collaborate with a few available GPS broadcasting devices via WiFi hotspot mode or WiFi Direct-based approximation. A beneficiary device may use one or more locations' information from neighbors opportunistically, depending on their availability. Furthermore, each device improves the received location accuracy via various available methods including trilateration or approximation with geomagnetic sensors. We have implemented an ECOPS prototype application on Android 2.3.6 and 2.2.2 and have tested it with various types of Android mobile devices. The results show that ECOPS provides accuracy within 10 m for nearly 60% of the location estimates, and within 15 m for more than 90% of them. ECOPS also offers significantly more energy efficiency than a GPS-only scheme, while overcoming various service limitations.

Future work includes fine-tuning the accuracy by opportunistically exploiting other available sensors on mobile devices. We also plan to consider strategies for various types of LBSs to improve their utility and energy efficiency.

References

[1] E. Kwon, S. Song, D. Seo, and I. Jung, "Agent-based on-line traffic condition and information analysis system for wireless V2I communication," in *Proceedings of the 2nd International Conference on Ubiquitous and Future Networks (ICUFN '10)*, pp. 360–365, Jeju Island, Republic of Korea, June 2010.

[2] "Global positioning system standard positioning service performance standard," http://www.gps.gov/technical/ps/2008-SPS-performance-standard.pdf.

[3] "Skyhook wireless," http://www.skyhookwireless.com/.

[4] E. Trevisani and A. Vitaletti, "Cell-ID location technique, limits and benefits: an experimental study," in *6th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA '04)*, pp. 51–60, December 2004.

[5] "Google latitude," 2012, http://www.google.com/latitude.

[6] "Foursquare," https://foursquare.com/.

[7] U. Bareth and A. Kupper, "Energy-efficient position tracking in proactive location-based services for smartphone environments," in *Proceedings of the 35th IEEE Annual Computer Software and Applications Conference (COMPSAC '11)*, pp. 516–521, Munich, Germany, July 2011.

[8] J. Paek, J. Kim, and R. Govindan, "Energy-efficient rate-adaptive GPS-based Positioning for smartphones," in *Proceedings of the 8th Annual International Conference on Mobile Systems, Applications and Services (MobiSys '10)*, pp. 299–314, New York, NY, USA, June 2010.

[9] "Android 2. 2 platform highlights—portable hotspot," http://developer.android.com/about/versions/android-2.2-highlights.html.

[10] "Wi-Fi direct," http://developer.android.com/guide/topics/wireless/wifip2p.html.

[11] Y. Zhang, L. Yang, and J. Chen, *RFID and Sensor Networks: Architectures, Protocols, Security and Integrations*, CRC Press, New York, NY, USA, 2009.

[12] T. Rappaport, *Wireless Communications Principles and Practice*, Prentice Hall, New York, NY, USA, 1996.

[13] "3-Axis geomagnetic sensor for electronic compass," http://developer.android.com/guide/topics/connectivity/wifip2p.html.

[14] M. Mohorcic, D. Grace, G. Kandus, and T. Tozer, "Broadband communications from aerial platform networks," in *Proceedings of the 13th IST Mobile and Wireless Communications Summit*, pp. 257–261, 2004.

[15] "LANdroids robot," 2012, http://www.irobot.com/gi/research/AdvancedPlatforms/LANdroidsRobot.

[16] "The air force has its own batman," 2012, http://money.cnn com/video/technology/2012/05/18/t-ts-wpafb-batman.cnnmoney/.

[17] "U.S. government, military to get secure Android phones," 2012, http://www.cnn.com/2012/02/03/tech/mobile/government-android-phones/index.html.

[18] National Security Agency(NSA), "Nsa-grade security enhanced android source released," 2012, http://www.androidauthority.com/nsa-grade-securityenhanced-android-source-released-45532/.

[19] "Google maps location based services," http://support.google.com/maps/bin/answer.py?hl=en&answer=1725632.

[20] "Apple location based services," http://www.apple.com/privacy/.

[21] B. N. Schilit, A. LaMarca, G. Borriello et al., "Challenge: ubiquitous location-aware computing and the "place lab" initiative," in *Proceedings of the 1st ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots (WMASH '03)*, pp. 29–35, September 2003.

[22] W. Ho, A. Smailagic, D. P. Siewiorek, and C. Faloutsos, "An adaptive two-phase approach to WiFi location sensing," in *Proceedings of the 4th Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops '06)*, pp. 452–456, Pisa, Italy, March 2006.

[23] D. Kelly, R. Behant, R. Villingt, and S. McLoone, "Computationally tractable location estimation on WiFi enabled mobile

phones," in *Proceedings of the IET Irish Signals and Systems Conference (ISSC)*, 2009.

[24] K. Chintalapudi, A. P. Iyer, and V. N. Padmanabhan, "Indoor localization without the pain," in *Proceedings of the Mobile Computing and Networking (MobiCom '10)*, pp. 173–184, 2010.

[25] "Blackberry locate service," https://developer.blackberry.com/devzone/develop/platform_services/platform_locate.html.

[26] "Ekahau real time location system (RTLS)," http://www.ekahau.com/products/real-time-location-system/overview.html.

[27] "Cisco AeroScout location-based services solution," http://www.cisco.com/web/strategy/docs/manufacturing/Aeroscout--Cisco-Brochure.pdf.

[28] "Broadcom: new location architecture withBCM4752," http://www.broadcom.com/products/features/GNSS.php.

[29] F. Schrooyen, I. Baert, S. Truijen et al., "Real time location system over WiFi in a healthcare environment," *Journal on Information Technology in Healthcare*, vol. 4, no. 6, pp. 401–416, 2006.

[30] A. Ofstad, E. Nicholas, R. Szcodronski, and R. R. Choudhury, "AAMPL: accelerometer augmented mobile phone localization," in *Proceedings of the 1st ACM International Workshop on Mobile Entity Localization and Tracking in GPS-Less Environments (MELT '08)*, pp. 13–18, New York, NY, USA, September 2008.

[31] J. Paek, K. H. Kim, J. P. Singh, and R. Govindan, "Energy-efficient positioning for smartphones using cell-ID sequence matching," in *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services (MobiSys '11)*, pp. 293–306, New York, NY, USA, July 2011.

[32] M. Ibrahim and M. Youssef, "A hidden Markov model for localization using low-end GSM cell phones," in *IEEE International Conference on Communications (ICC '11)*, pp. 1–5, Kyoto, Japan, 2011.

[33] E. Martin, O. Vinyals, G. Friedland, and R. Bajcsy, "Precise indoor localization using smart phones," in *Proceedings of the 18th ACM International Conference on Multimedia ACM Multimedia 2010 (MM '10)*, pp. 787–790, October 2010.

[34] H. Liu, Y. Guan, J. Yang et al., "Push the Limit of WiFi based Localization for Smartphones," in *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking (Mobicom '12)*, pp. 305–316, 2012.

[35] "Calculating distance between two points given longitude/latitude," http://www.freevbcode.com/ShowCode.asp?ID=5532.

[36] R. W. Sinnott, "Virtues of the Haversine," *Sky and Telescope*, vol. 68, no. 2, article 158, 1984.

[37] K. Aamodt, "Applicationnote AN042 (Rev.1.0)," pp. 7-8, 2006.

[38] "PowerTutor," http://ziyang.eecs.umich.edu/projects/powertutor/.

Hindawi Publishing Corporation
Journal of Computer Networks and Communications
Volume 2013, Article ID 794202, 15 pages
http://dx.doi.org/10.1155/2013/794202

Research Article

Equalizer's Use Limitation for Complexity Reduction in a Green Radio Receiver

Salma Bourbia,[1] Daniel Le Guennec,[1] Jacques Palicot,[1] Khaled Grati,[2] and Adel Ghazel[2]

[1] *IETR, Supélec, Campus de Rennes Avenue de la Boulaie, CS 47601, 35576 Cesson-Sévigné Cedex, France*
[2] *SUPCOM, University of Carthage, Cité Technologique des Communications, El Ghazala 2083 Tunis, Tunisia*

Correspondence should be addressed to Salma Bourbia; salma.bourbia@supelec.fr

Received 15 June 2012; Accepted 22 November 2012

Academic Editor: Lingjia Liu

This work is about reducing energy consumption in the receiver chain by limiting the use of the equalizer. It is to make the radio receiver aware of its environment and able to take decision to turn on or off the equalizer according to its necessity or not. When the equalizer is off, the computational complexity is reduced and the rate of reduction depends on the percentage of time during which this component is disabled. In order to achieve this scenario of adapting the use of the equalizer, we need to develop a decision-making technique that provides the receiver with the capacities of awareness and adaptability to the state of its environment. For this, we improve a technique based on a statistical modeling of the environment by defining two metrics as channel quality indicators to evaluate the effect of the intersymbol interferences and the channel fading. The statistical modeling technique allows to take into account the impact of the uncertainties of the estimated metrics on the decision making.

1. Introduction

The requirements in wireless communications are increasingly growing in terms of services and transmission's quality. Technological advances in this sense, either in the nodes or in the base stations, have improved the quality of service offered to the users of the wireless communication networks. However, these advances are not without side effects; with this development and with the growing number of mobile phones, the wireless communications consume more and more energy causing so an increase in CO_2 emissions. In front of this phenomenon, the solutions that make communications less consuming in terms of energy are needed for the preservation of the environment. With this aim, the concept of green communications, or green radio, allows to develop a radio equipment that consumes less energy. One of the solutions to obtain a green radio is to make it able to adapt dynamically to the environment and to use this adaptability in order to reduce energy consumption [1]. In fact, according to the state of its environment, the radio receiver with cognitive capacities can take some decisions of reconfigurations, and so it can select the reconfigurations that need less amount of energy than others.

Within this context, we focus in this paper on reducing energy consumption in the radio receiver through its adaptation to its environment. In particular, we are looking at the equalizer component and we seek to limit its use in order to reduce energy consumption. The idea is to make the radio receiver capable of being aware of its environment and deciding to turn off the equalizer when it is not useful and to turn it on as soon as it becomes necessary, so that the receiver performance does not degrade. The fact to get rid of this component for a period of time of the communication should reduce energy consumption in the receiver chain. So we try also to determine from what period of time, of disabling the equalizer, can we save energy.

The reminder of this paper is organized as follows. Section 2 describes some previous works that were looking for the same reason of switching off the equalizer. In Section 3 we develop the decision-making method for deciding to turn off the equalizer. In Section 4 we present the results of the computational complexity reduction and other results of simulation.

2. Previous Works

In the literature, the idea of limiting the use of the equalizer during a period of time of the communication is not new; some previous research works have focused on this for different reasons other than the energy consumption reduction. In fact, we notice this in the work [2] where the authors tried to adapt the use of the equalizer to the state of the channel in order to avoid the presence of this component in an environment without delay spread distortion as it can cause sensitivity degradation. The proposed solution in this work consists of two coherent detection algorithms; the first contains a decision feedback equalizer and the second is without any equalizer. A detector selection algorithm is used in order to dynamically select the corresponding detector depending on whether delay spread distortion is present or not. For this the correlation of the two kinds of detectors is measured and the selection algorithm chooses the one that have the greater correlation compared to a threshold. The problem of this invention is that the equalizer is always used, but also there is an additional stage of a detector without equalizer; this makes the computational complexity of the solution very high. In [3], the authors proposed a radio receiver with a selectively disabled equalizer. The aim of this work is to develop a technique for selectively and automatically disabling the equalizer in particular situations in order to avoid a prejudicial impact of the equalization in these cases. In particular, the discussed case is to disable the equalizer as long as a spurious signal is present for the FM radio receiver. For this, an equalizer controller compares the RF demodulated signal with a predetermined distribution for the received signal in order to determine if a spurious signal is present or not and so to disable or to keep the equalizer. The energy consumption aspect was not considered in this work since it was not the essential purpose of the invention.

The purpose of reducing the energy consumption started to be highlighted in [4]. In this work, the equalizer is powered off for reduced power consumption when there is small intersymbol interference and it is powered on for better signal reception. The idea is based on the analysis of the received signal quality after demodulation, by inserting a decision circuit at the receiver that compares the quality of the demodulated signal to a predetermined threshold. Basing on a couple of integrator and a low-pass filter, the decision error estimation of the signal quality is compared to a threshold and the result allows to decide if the quality of the signal is poor or good. According to this decision, the equalizer is powered on or off. In determining the quality of the received signal, the authors do not specify whether the degradation of the signal is due to the intersymbol interference or to other phenomena, but they are relying on the total degradation. For the same reason of reducing energy consumption, [5] proposes a new method, called conditional equalization, which minimizes the power consumption by equalizing only when it is most needed. At each time slot, the receiver decides if the equalization is needed or not by defining two criteria in order to distinguish between weakly dispersive channels and dispersive channels in the case of a two-path Rayleigh channel. The first criterion (C1) is based on the paths' energies; in fact, the receiver decides to not equalize the received signal only if the energy of one path is far greater than the total energy of the other paths. This criterion is simple to implement and so the additional computational complexity is small [6]. The second criterion (C2) is based on the intersymbol interference contribution in the degradation of the signal. For this, the determination of the path having the greatest energy is needed, and the receiver decides to not equalize only if the projection of the other paths on the axis defined by the greatest energy path is far smaller. This means that the effect of the intersymbol interference is negligible. In this work, the authors assumed that the channel coefficients are perfectly estimated and known; in addition to this, they did not study the impact of the variation of the channel on the decision system. The reduction of the energy consumption by disabling the equalizer was also enhanced in [7] where the authors developed a receiver with hybrid equalizer and RAKE receiver. The receiver enables or disables the equalizer according to the performance of the system for RAKE only and for RAKE with the equalizer. A measure for the channel quality allows to switch between the mode RAKE only and the mode RAKE with the equalizer. For this, two evaluations are taken into consideration: the first is the measure of the Signal to Interference plus Noise Ratio (SINR) and the second evaluation determines the speed of the channel. Finally we mention the work [8] which proposes a technique for selecting between an equalizing demodulator and a nonequalizing demodulator in a receiver.

Although the previous works had the objective of reducing the energy consumption or the computational complexity, none has tackled this notion, not even the existence of such a reduction related to their solutions. In our work, we present precisely some results concerning the rate of the complexity reduction when we limit the use of the equalizer in the receiver chain and also the maximum gains that can be achieved. Moreover, our work will be distinguished from the earlier works described above by the decision-making technique. In fact, we develop a technique that takes into account the uncertainty of the channel quality indicators' values used to decide to disable or keep the equalizer as they are estimated unlike the other techniques that ignore this aspect by assuming a perfect estimate of the metrics [5, 6] or by not considering the impact of these uncertainties on the decision [7, 8].

3. A Decision-Making Method to Adapt the Use of the Equalizer

Our idea is to statistically model and characterize the radio environment by using some techniques of the statistical inference like statistical estimation [9] and statistical detection [10]. We define for this some metrics in order to evaluate the quality of the channel. Basing on these channel indicators and on their statistical characteristics, we can evaluate the current state of the channel and then decide if the equalizer is necessary in this case or not. The use of the statistical aspect of the channel quality indicators allows to take into account the estimation errors of these metrics in the decision-making system which affect the decision performance. In the following paragraphs, we present the decision-making method based on the statistical modeling of the environment for the scenario of driving the equalizer in the receiver chain.

3.1. Definition of the Channel Quality Indicators. The first step to do is to define the channel quality indicators that are necessary to evaluate the state of the channel and then to choose between a use of the equalizer or not. We consider for this a multipath channel from length of L with an added Gaussian noise. The received signal is then expressed as in

$$y[k] = \sum_{i=0}^{L-1} h_i \cdot s[k-i] + b[k], \quad (1)$$

with s: the transmitted signal, y: the received signal, b: the added Gaussian noise, $h = (h_0, \ldots, h_{L-1})$: the coefficients of the multi-path channel.

We assume that the interferences, which the signal suffers from, are limited to the intersymbol interferences caused by delay dispersion of the multiple paths of the channel. At the receiver, the signal can be deteriorated by the noise and by the inter-symbol interferences. In order to distinguish between the two sources of signal degradation, we are defining two independent radio metrics. The first that we denote SNR_p is the signal-to-noise ratio without taking into account the effect of the inter-symbol interferences and the second that we denote ISI is the power of the inter-symbol interferences. For this we will consider two cases of the channel: the Rician channel and the Rayleigh channel.

3.1.1. In the Rician Channel. Since the Rician channel contains a direct path, we can express the received signal as in

$$y[k] = h_0 s[k] + \sum_{i=1}^{L-1} h_i \cdot s[k-i] + b[k]. \quad (2)$$

SNR_p evaluates the level of the degradation of the received signal by the noise and the fading without considering the interferences. This means that we consider only the fading of the signal observed in the direct path h_0. So in this case, SNR_p is defined in

$$\mathrm{SNR}_p = \frac{\|h_0\|^2 \cdot \sigma_s^2}{\sigma_b^2}, \quad (3)$$

with σ_s^2 and σ_b^2 are, respectively, the variance of the input signal and the variance of the noise. Concerning the second metric ISI, we are interested in the term $\sum_{i=1}^{L-1} h_i \cdot s[k-i]$ of the relation (2), which describes the phenomenon of the inter-symbol interferences. From this expression, we can define a new metric ISI to represent the power of the inter-symbol interferences as described in (4). We assume that the input symbols $s[k]$ follow an uniform distribution with mean $E\{s[k]\} = 0$ as follows:

$$\mathrm{ISI} = E\left\{\left(\sum_{i=1}^{L-1} h_i \cdot s[k-i]\right)^2\right\}, \quad (4)$$

where $E\{\}$ denotes the expectation operation.

Assuming that the transmitted symbols $s[k]$ are identically distributed and independent, the evaluation of the expression (4) leads to the following relation

$$\mathrm{ISI} = \sum_{i=1}^{L-1} \|h_i\|^2 \cdot E\left\{[s[k-i]]^2\right\}. \quad (5)$$

3.1.2. In the Rayleigh Channel. The Rayleigh channel does not contain a direct path, so in order to define the metric SNR_p we consider the path that has the highest energy, which means the highest signal to noise ratio. We denote this path $h_{i\max}$ and then the metric SNR_p is expressed in

$$\mathrm{SNR}_p = \frac{\|h_{i\max}\|^2 \sigma_s^2}{\sigma_b^2}, \quad (6)$$

with $\mathrm{SNR}_p > (\mathrm{SNR}_i)_{(i=0,\ldots,L-1), i \neq i\max,} \mathrm{SNR}_i = (\|h_i\|^2 \sigma_s^2 / \sigma_b^2)_{i \neq i\max}$ is the signal to noise ratio of the ith path.

The rest of the paths $(h_i)_{i \neq i\max}$ are adding the inter-symbol interferences, so we can define the metric ISI as follows:

$$\mathrm{ISI} = \sum_{(i=0, i \neq i\max)}^{L-1} \|h_i\|^2 \cdot E\left\{[s[k-i]]^2\right\}. \quad (7)$$

As a conclusion, if we consider that the transmitted signal is normalized ($\sigma_s^2 = 1$), then our two channel quality indicators are expressed as follows for the Rician case and the Rayleigh case:

$$\begin{gathered} \mathrm{SNR}_p = \frac{\|h_0\|^2}{\sigma_b^2}_{\text{Rice}}, \\ \mathrm{SNR}_p = \frac{\|h_{i\max}\|^2}{\sigma_b^2}_{\text{Rayleigh}}, \\ \mathrm{ISI} = \sum_{i=1}^{L-1} \|h_i\|^2_{\text{Rice}}, \\ \mathrm{ISI} = \sum_{(i=0, i \neq i\max)}^{L-1} \|h_i\|^2_{\text{Rayleigh}}. \end{gathered} \quad (8)$$

In the next paragraphs, we are focusing on the estimation of these metrics derived from the channel estimation, then we present our decision system based on the statistical modeling of the environment.

3.2. Estimation of the Channel Quality Indicators. Since the metrics ISI and SNR_p are based on the channel coefficients $(h_i)_{i=0,\ldots,L-1}$, their estimation needs an algorithm for channel estimation. Many techniques for channel estimation are proposed in the literature; we can distinguish blind techniques, techniques with pilot symbols, adaptive techniques, and nonadaptive techniques. In our case we search for one technique that provides a less estimation error and a less computational complexity. For this, we propose a comparison between some techniques of estimation and we select the one that presents the best compromise between the less mean-squared error of the estimation and the less computational complexity expressed in terms of the number of multiplication operations in the algorithm.

LS (Least-Squared) Channel Estimation [11]. It is a nonadaptive algorithm that is based on a known training sequence from the transmitted signal. For each received frame, a training matrix M is defined as follows:

$$M = \begin{pmatrix} m_L & \cdots & m_0 \\ \vdots & \ddots & \vdots \\ m_{P+L-1} & \cdots & m_{P-1} \end{pmatrix}, \tag{9}$$

where P is the number of the pilot symbols.

The LS estimation is a minimization of the quadratic error:

$$\begin{aligned} \underline{\hat{h}}_{\text{LS}} &= \operatorname{argmin}\|y - Mh\|^2 \\ &= \left(M^H M\right)^{-1} M^H y, \end{aligned} \tag{10}$$

where $()^H$ and $()^{-1}$ are, respectively, the hermitien and the inverse matrices and $\underline{h}$ is the vector of the channel coefficients.

Technique by Intercorrelation. It is an estimation method that we have developed by inspiring from [12]. Its idea is based on the intercorrelations between the transmitted pilot symbols, with delay, and the received symbols. In fact, the intercorrelation product between the received symbol $y[k]$ and the transmitted symbol $s[k-i]$ delayed by $i = (0, \ldots, L-1)$ is expressed as

$$\begin{aligned} & E\left\{s[k-i]^* \cdot y[k]\right\} \\ &\quad = E\left\{s[k-i]^* \left(\sum_{i=0}^{L-1} h_i \cdot s[k-i] + b[k]\right)\right\} = h_i. \end{aligned} \tag{11}$$

In practice $E\{s[k-i]^* \cdot y[k]\}$ is approximated with a finite number of symbols that represent the training sequence; this provides an estimated value of the coefficients, that is, $\hat{h}_i$.

LMS (Least Mean-Squared) Channel Estimation. It is an adaptive algorithm that is based on the stochastic gradient. Assuming e the error of the adaptive filter at the instant time n such as $e(n) = y(n) - \underline{\hat{h}}_{\text{LMS}}^T(n)\underline{s}(n)$, the update of the channel coefficients estimation is expressed in

$$\underline{\hat{h}}_{\text{LMS}}(n+1) = \underline{\hat{h}}_{\text{LMS}}(n) + \mu e(n)\, \underline{s}^H(n). \tag{12}$$

The parameter μ represents the step size and has an influence on the convergence of the algorithm.

All these techniques are with pilot symbols, which causes a bad impact on the useful throughput, unlike the blind techniques that are without pilot symbols. As one of these blind methods, we can mention the constant modulus algorithm.

Constant Modulus Algorithm [13, 14]. It is also an adaptive algorithm but blind, which tries to minimize the cost function $J\{(|V_n|^2 - C)^2\}$ with $C = E[s]^4/E[s]^2$, $V_n = \underline{\hat{h}}_{\text{CMA}}(n)\underline{Y}(n)$, $Y_n = [y(n), \ldots, y(n-L)]$.

Then the updated channel coefficient estimation is expressed in

$$\underline{\hat{h}}_{\text{CMA}}(n+1) = \underline{\hat{h}}_{\text{CMA}}(n) - \mu \underline{Y}(n) V_n \left(|V_n|^2 - C\right). \tag{13}$$

Then from the obtained values of $\hat{h}_i$, we can deduce the estimation of our metrics ISI and SNR_p as follows:

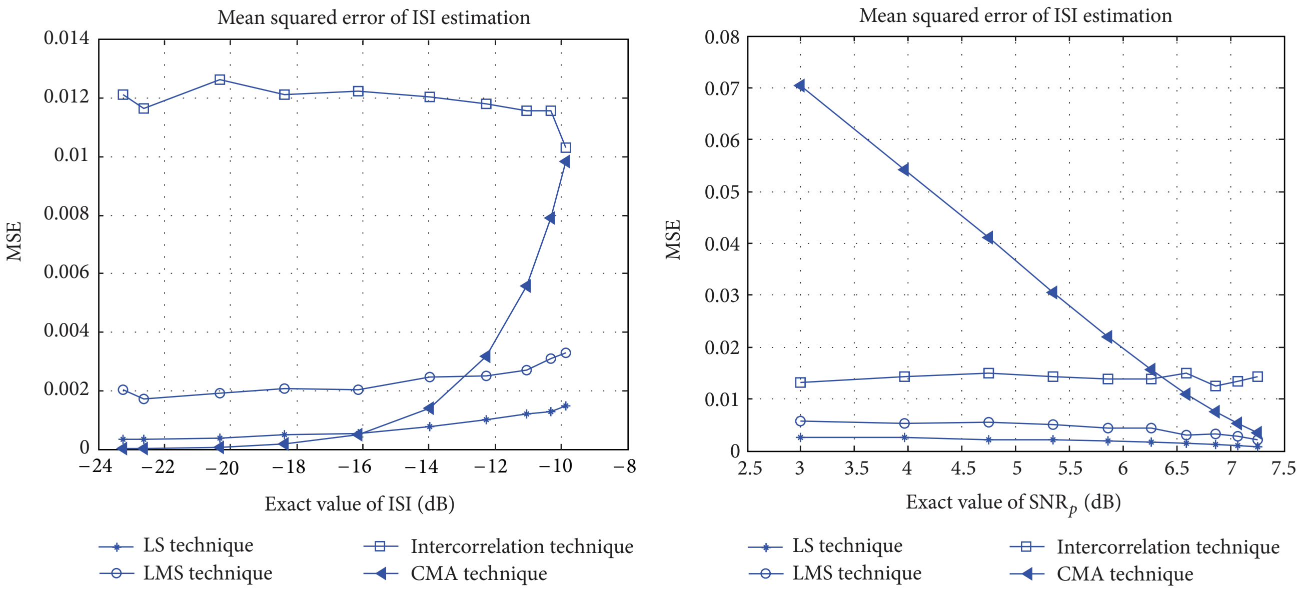
$$\begin{aligned} \widehat{\text{SNR}}_p &= {\frac{\|\hat{h}_0\|^2}{\sigma_b^2}}_{\text{Rice}}, \\ \widehat{\text{SNR}}_p &= {\frac{\|\hat{h}_{i\max}\|^2}{\sigma_b^2}}_{\text{Rayleigh}}, \\ \widehat{\text{ISI}} &= \sum_{i=1}^{L-1} \|\hat{h}_i\|^2_{\text{Rice}}, \\ \widehat{\text{ISI}} &= \sum_{(i=0, i \neq i\max)}^{L-1} \|\hat{h}_i\|^2_{\text{Rayleigh}}. \end{aligned} \tag{14}$$

In order to select the best channel estimation algorithm for us, we dress, in Figures 1 and 2, the mean-squared Error curves of the estimation of ISI and SNR_p by using the methods LS, LMS, CMA, and inter-correlation. Since we are in an environment variable in time, we will study the effect of the variation, of the inter-symbol interferences, and of the fading, on the estimates of ISI and SNR_p. For this, we draw the curves $\text{MSE}(\widehat{\text{ISI}}) = f(\text{ISI})$, $\text{MSE}(\widehat{\text{ISI}}) = f(\text{SNR}_p)$, and $\text{MSE}(\widehat{\text{SNR}}_p) = f(\text{ISI})$, $\text{MSE}(\widehat{\text{SNR}}_p) = f(\text{SNR}_p)$. The curves of simulation are not depending on the type of the channel. In another side, as we search also for algorithms with minimum computational complexity, we determine the equations of the complexity of the estimation of ISI and SNR_p with each technique. For this, we present in Tables 1 and 2 the computational complexity CC of the estimation operations that we define as the number of the operations of multiplication. Basing on the two criteria (MSE and complexity), we will choose the channel estimation algorithm.

We denote by

(i) $\widehat{\text{Metric}}^{\text{LS}}$, $\widehat{\text{Metric}}^{\text{LMS}}$, $\widehat{\text{Metric}}^{\text{CMA}}$, and $\widehat{\text{Metric}}^{\text{inter}}$: the estimated value of the metric Metric by using the channel estimation algorithm LS, LMS, CMA, and inter-correlation. Metric can be ISI or SNR_p,

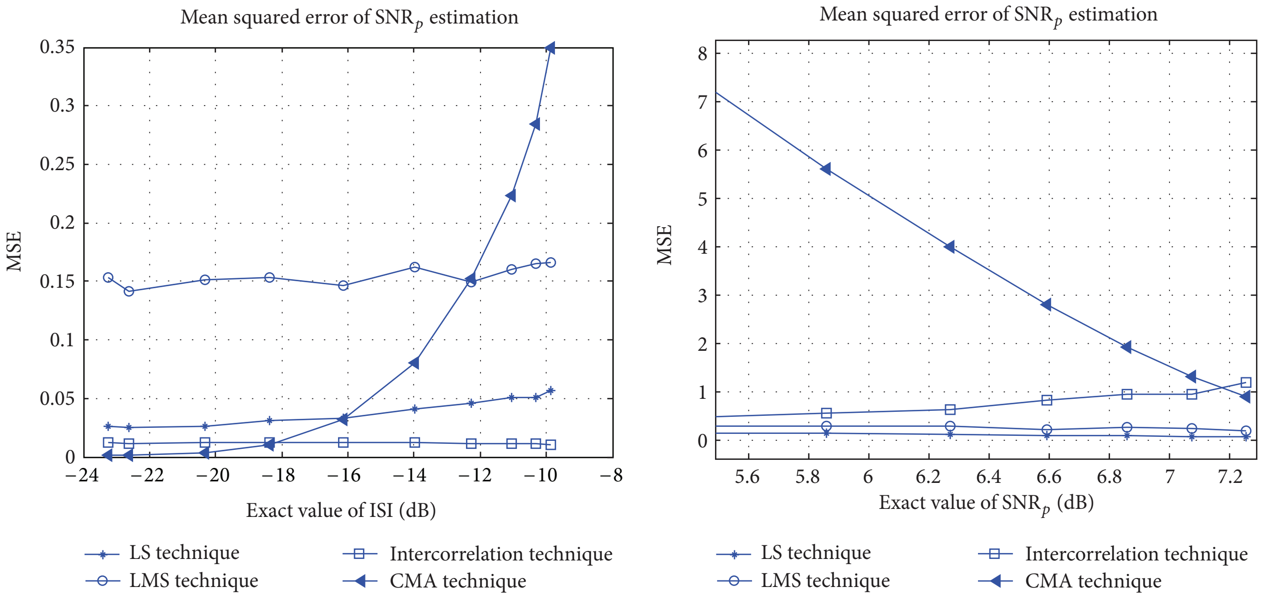
(ii) P: the number of the pilot symbols for the nonblind algorithms,



(a) $\text{MSE}(\widehat{\text{ISI}}) = f(\text{ISI})$

(b) $\text{MSE}(\widehat{\text{ISI}}) = f(\text{SNR}_p)$

Figure 1: Mean squared error of $\widehat{\text{ISI}}$.



(a) $\text{MSE}(\widehat{\text{SNR}}_p) = f(\text{ISI})$

(b) $\text{MSE}(\widehat{\text{SNR}}_p) = f(\text{SNR}_p)$

Figure 2: Mean squared error of $\widehat{\text{SNR}}_p$.

(iii) N: the number of the received symbols considered in the estimation for the blind algorithm,

(iv) L: the length of the channel.

In particular, for the values $P = 20$, $N = 20$, and $L = 5$, we found $\text{CC}_{\text{ISI}}^{\text{LMS}} = 224$ operations, while $\text{CC}_{\text{ISI}}^{\text{CMA}} = 1184$ operations, $\text{CC}_{\text{ISI}}^{\text{LS}} = 13104$ operations, and $\text{CC}_{\text{ISI}}^{\text{inter}} = 79$ operations.

It is true that, unlike the algorithms with pilot symbols, the blind algorithm CMA allows a throughput more interesting. But this technique is very sensitive to both the intersymbol interferences and the channel fading. In fact according to the curves of MSE in Figures 1(a) and 2(a), it is clear that the higher the value of ISI, the higher the mean-squared error of estimation with CMA technique. Similarly, we notice from Figures 1(b) and 2(b) that the mean squared error of this technique increases while the SNR$_p$ is degraded.

Table 1: Computational complexity of ISI estimation.

Method of ISI estimation	Computational complexity
$\widehat{\text{ISI}}^{\text{LS}}$	$\text{CC}_{\text{ISI}}^{\text{LS}} = 2PL^2 + PL + L + (L-1)L^2L! - 1$
$\widehat{\text{ISI}}^{\text{LMS}}$	$\text{CC}_{\text{ISI}}^{\text{LMS}} = 2LP + P + L - 1$
$\widehat{\text{ISI}}^{\text{CMA}}$	$\text{CC}_{\text{ISI}}^{\text{CMA}} = N(2L^2 + L + 4) + L - 1$
$\widehat{\text{ISI}}^{\text{inter}}$	$\text{CC}_{\text{ISI}}^{\text{inter}} = 2L - 1 + \sum_{i=1}^{L-1}(P-i)$

Table 2: Computational complexity of SNR_p estimation.

Method of SNR_pestimation	Computational complexity
$\widehat{\text{SNR}_p}^{\text{LS}}$	$\text{CC}_{\text{SNR}_p}^{\text{LS}} = 2PL^2 + 2 + PL + (L-1)L^2L!$
$\widehat{\text{SNR}_p}^{\text{LMS}}$	$\text{CC}_{\text{SNR}_p}^{\text{LMS}} = 2LP + 2 + P$
$\widehat{\text{SNR}_p}^{\text{CMA}}$	$\text{CC}_{\text{SNR}_p}^{\text{CMA}} = N(2L^2 + L + 4) + 2$
$\widehat{\text{SNR}_p}^{\text{inter}}$	$\text{CC}_{\text{SNR}_p}^{\text{inter}} = P + 3$

Furthermore this method has a computational complexity relatively high compared to the other methods.

Concerning the estimation technique by intercorrelations, although it is the least complex one, it presents an increased MSE especially in Figures 1(a), 1(b) and 2(b). Finally, the LS technique has a good performance of estimation, but it is the most complex one. In conclusion, in this comparison, the LMS technique seems to be the most compatible in the compromise (estimation performance/complexity). In addition, it is an adaptive algorithm that is very adapted to the change of the radio channel. So in our simulations, we choose to use the LMS algorithm for the estimation of SNR_p and ISI.

3.3. Statistical Modeling of the Environment and Decision Making. In this section, we will describe the method that we have developed in order to determine the decision rule to make decision for using or not using the equalizer from the statistical modeling of the environment [15].

We consider $\widehat{x}_{\text{SNR}_p}^{i}$ (resp., $\widehat{x}_{\text{ISI}}^{i}$) one estimation of the SNR_p (resp., ISI) in the instant i, given by the sensor algorithms selected in Section 3.2. Then the vectors $(\widehat{x}_{\text{SNR}_p}^{1}, \ldots, \widehat{x}_{\text{SNR}_p}^{n})$ (resp., $(\widehat{x}_{\text{ISI}}^{1}, \ldots, \widehat{x}_{\text{ISI}}^{n})$) represent n estimations of SNR_p (resp., ISI), with $(\widehat{x}_{\text{SNR}_p}^{i})_{i=1,\ldots,n}$ (resp., $(\widehat{x}_{\text{ISI}}^{i})_{i=1,\ldots,n}$) being identically distributed and independent following the same distribution as a random variable $\widehat{X}_{\text{SNR}_p}$ (resp., $\widehat{X}_{\text{ISI}}$).

In order to characterize statistically our channel quality indicators SNR_p and ISI, we have to determine the densities of probability of the variables $\widehat{X}_{\text{SNR}_p}$ and $\widehat{X}_{\text{ISI}}$. From the statistical inference theory, two classes of techniques are proposed in order to estimate the density of probability of a random variable given its realizations; for this, we mention the parametric estimation techniques and the nonparametric estimation techniques [9, 10]. The nonparametric method estimates both the distribution shape and the statistical parameters of a set of observations whereas the parametric method estimates only the statistical parameters when the distribution of the observations is known. In our case the analysis that we perform on the estimation of our radio metrics ISI and SNR_p shows that a Gaussian distribution represents well the distributions of the observations provided by the sensors of the metrics, and therefore the parametric method is enough to determine the statistical parameters of these observations. So, since the information about the distributions of the observed metrics is known, we do not need the nonparametric method especially if it adds complexity to estimate these distributions.

We denote by $N(\widehat{\mu}_{\text{SNR}_p}^{\text{ML}}, \widehat{\sigma}_{\text{SNR}_p}^{\text{ML}})$ and $N(\widehat{\mu}_{\text{ISI}}^{\text{ML}}, \widehat{\sigma}_{\text{ISI}}^{\text{ML}})$ the statistical parameters (mean and variance) of $\widehat{X}_{\text{SNR}_p}$ and $\widehat{X}_{\text{ISI}}$. By applying the parametric technique, maximum likelihood estimator, we find the expressions following:

$$\begin{gathered}
\widehat{\mu}_{\text{SNR}_p}^{\text{ML}} = \frac{1}{n}\sum_{i=0}^{n}\widehat{x}_{\text{SNR}_p}^{i}, \\
\widehat{\sigma}_{\text{SNR}_p}^{\text{ML}} = \sqrt{\frac{1}{n-1}\sum_{i=0}^{n}\left(\widehat{x}_{\text{SNR}_p}^{i} - \widehat{\mu}_{\text{SNR}_p}^{\text{ML}}\right)^2}, \\
\widehat{\mu}_{\text{ISI}}^{\text{ML}} = \frac{1}{n}\sum_{i=0}^{n}\widehat{x}_{\text{ISI}}^{i}, \\
\widehat{\sigma}_{\text{ISI}}^{\text{ML}} = \sqrt{\frac{1}{n-1}\sum_{i=0}^{n}\left(\widehat{x}_{\text{ISI}}^{i} - \widehat{\mu}_{\text{ISI}}^{\text{ML}}\right)^2}.
\end{gathered} \tag{15}$$

So, after n realizations of $\widehat{x}_{\text{SNR}_p}^{i}$ and $\widehat{x}_{\text{ISI}}^{i}$, we obtain the estimated values of our metrics, as described in

$$\begin{gathered}
\widehat{\text{SNR}}_p = \widehat{\mu}_{\text{SNR}_p}^{\text{ML}}, \\
\widehat{\text{ISI}} = \widehat{\mu}_{\text{ISI}}^{\text{ML}}.
\end{gathered} \tag{16}$$

By using these estimated values and their statistical parameters, we try to evaluate the channel quality and to determine the decision rule for deciding whether the equalizer is necessary or unnecessary. For this. The evaluation of the channel quality indicators is performed by asserting or refuting some general hypothesis. The objective is to decide between the two actions:

$$\begin{gathered}
A_1 : \text{turn-off-the-equalizer}, \\
A_0 : \text{turn-on-the-equalizer}.
\end{gathered} \tag{17}$$

It is necessary to determine the situations of the environment where A_1 is possible and those where A_0 is possible. In fact, according to the state of ISI and SNR_p, we confirm if the received signal is deteriorated by the noise only, by the inter-symbol interferences only, or by both of them. The evaluation of this state is done according to the thresholds of performance λ_{SNR} and λ_{ISI} which we defined as the values of SNR_p and ISI for a minimum bit error rate BER $= 10^{-3}$. While the value of λ_{SNR} is normalized by the standard used in the communication, there is no indication about the value of λ_{ISI}. In our work, we determined this threshold by taking the best measured value of ISI that provides BER = 10^{-3}.

According to the values of ISI and SNR_p, we can define three states of the environment. Table 3 presents these states with the corresponding actions for each one.

When the power of the inter-symbol interference is greater than λ_{ISI}, the signal is affected and the equalizer is necessary to reduce the inter-symbol interference. In the case where $\mathrm{SNR}_p < \lambda_{\mathrm{SNR}}$, we choose to keep the equalizer because, despite the fact that the BER still above 10^{-3}, the equalizer will reduce it anyway. The case where we can decide to turn off the equalizer is when the signal is not deteriorated neither by the noise nor by the inter-symbol interferences.

To translate this evaluation, statistically, and deduce the decision rule of our decision system, we construct the binary hypothesis test described in (18), where H_1 corresponds to the decision to turn off the equalizer and the hypothesis H_0 for the decision to turn it on.

$$\begin{aligned} H_1 &: \mathrm{SNR}_p \geq \lambda_{\mathrm{SNR}}, \quad \mathrm{ISI} \leq \lambda_{\mathrm{ISI}}, \\ H_0 &: \mathrm{SNR}_p < \lambda_{\mathrm{SNR}} \quad \text{or} \quad \mathrm{ISI} > \lambda_{\mathrm{ISI}}. \end{aligned} \tag{18}$$

The resolution of this hypothesis test leads to the decision rule to decide to disable the equalizer, and this occours by identifying new thresholds K_{ISI} and K_{SNR} since the quantities ISI and SNR_p are the estimated values and not the exact values. For this, we use the technique of the Neyman Pearson test and we are relying on the statistical characteristics of the metrics estimated and modeled previously. The technique of the Neyman Pearson consists of choosing the hypothesis which has the highest probability of correct decision under the constraint of a false alarm less than α which is the maximum probability of the false alarm tolerated for the decision system and fixed in advance. According to this method, there is no indication about how to set the value of α. The result is presented in (19), where δ_1 is the decision rule to decide to turn off the equalizer:

$$\delta_1 : \widehat{\mathrm{SNR}}_p \geq K_{\mathrm{SNR}}, \quad \widehat{\mathrm{ISI}} \leq K_{\mathrm{ISI}}, \tag{19}$$

$$\begin{aligned} K_{\mathrm{SNR}} &= \lambda_{\mathrm{SNR}} + \frac{\widehat{\sigma}^{\mathrm{ML}}_{\mathrm{SNR}_p}}{\sqrt{n}} F^{-1}(\alpha), \\ K_{\mathrm{ISI}} &= \lambda_{\mathrm{ISI}} + \frac{\widehat{\sigma}^{\mathrm{ML}}_{\mathrm{ISI}}}{\sqrt{n}} F^{-1}(\alpha), \end{aligned} \tag{20}$$

where

(i) F^{-1} is the inverse function of F and $F(x) = \int_{-\infty}^{x} (1/\sqrt{2\Pi}) e^{-t^2/2} dt$ is the distribution function of the standard normal.

(ii) n is the number of the realizations of the metrics' estimation.

(iii) $\widehat{\mathrm{SNR}}_p$ and $\widehat{\mathrm{ISI}}$ are Gaussian random variables with means near to the real values of SNR_p and ISI and with variances, respectively, $(\widehat{\sigma}^{\mathrm{ML}}_{\mathrm{SNR}_p})^2/n$ and $(\widehat{\sigma}^{\mathrm{ML}}_{\mathrm{ISI}})^2/n$ (Figure 3),

(iv) α is a fixed probability of false alarm. In our simulations, we fix this value to 1%.

This decision rule is defined in such a way that takes into account the impact of the uncertain measures of SNR_p and ISI on the decision thresholds.

Now in order to determine the theoretical performance of this decision system, we express the probabilities of false alarm P_{FA} and correct decision P_D defined as follows:

$$\begin{aligned} P_{\mathrm{FA}} &= P\left\{\text{Accept } \frac{H_1}{H_0} \text{ is true}\right\} \\ &= P\left\{\widehat{\mathrm{SNR}}_p \geq K_{\mathrm{SNR}}, \widehat{\mathrm{ISI}} \leq \frac{K_{\mathrm{ISI}}}{\mathrm{SNR}_p}\right. \\ &\qquad \left. < \lambda_{\mathrm{SNR}} \quad \text{or} \quad \mathrm{ISI} > \lambda_{\mathrm{ISI}}\right\}, \\ P_D &= P\left\{\text{Accept } \frac{H_1}{H_1} \text{ is true}\right\} \\ &= P\left\{\widehat{\mathrm{SNR}}_p \geq K_{\mathrm{SNR}}, \widehat{\mathrm{ISI}} \leq \frac{K_{\mathrm{ISI}}}{\mathrm{SNR}_p}\right. \\ &\qquad \left. \geq \lambda_{\mathrm{SNR}} \ , \mathrm{ISI} \leq \lambda_{\mathrm{ISI}}\right\}. \end{aligned} \tag{21}$$

By introducing the terms d_1, d_1', d_2 and d_2' as described in Figure 3, we developed these probabilities and we found the following relations:

$$\begin{aligned} P_{\mathrm{FA}} &= \left(1 - F\left(\frac{d_1}{\widehat{\sigma}^{\mathrm{ML}}_{\mathrm{SNR}_p}/\sqrt{n}} + F^{-1}(\alpha)\right)\right) \\ &\quad \times F\left(F^{-1}(\alpha) - \frac{d_2'}{\widehat{\sigma}^{\mathrm{ML}}_{\mathrm{ISI}}/\sqrt{n}}\right), \\ P_D &= \left(1 - F\left(F^{-1}(\alpha) - \frac{d_1'}{\widehat{\sigma}^{\mathrm{ML}}_{\mathrm{SNR}_p}/\sqrt{n}}\right)\right)^2 \\ &\quad \times \left(F\left(F^{-1}(\alpha) + \frac{d_2}{\widehat{\sigma}^{\mathrm{ML}}_{\mathrm{ISI}}/\sqrt{n}}\right)\right)^2. \end{aligned} \tag{22}$$

From these expressions we can determine the conditions on the value of n (number of the metrics' estimations used to one decision) in order to get situations where we start to have a probability of correct decision equal to 1 (no errors of decision) (23), and to get situations where we start to have a probability of false alarm equal to 0 (24).

$$\frac{\widehat{\sigma}^{\mathrm{ML}}_{\mathrm{SNR}_p}}{\sqrt{n}} \leq \frac{d_1^{'}}{2}, \qquad \frac{\widehat{\sigma}^{\mathrm{ML}}_{\mathrm{ISI}}}{\sqrt{n}} \leq \frac{d_2}{2}, \tag{23}$$

$$\frac{\widehat{\sigma}^{\mathrm{ML}}_{\mathrm{SNR}_p}}{\sqrt{n}} \leq \frac{d_1}{2}, \qquad \frac{\widehat{\sigma}^{\mathrm{ML}}_{\mathrm{ISI}}}{\sqrt{n}} \leq \frac{d_2^{'}}{2}. \tag{24}$$

4. Simulation Results

4.1. Proposed Solutions for Driving the Equalizer. In the implementation of the decision-making method to drive the equalizer, we have thought of two solutions for this. The first solution is described in Figure 4(b); this solution starts with a decision making block which contains our treatments, of metrics' estimations and decision making rule, which we developed previously in Section 3. According to the output of this block, the equalizer is turned off or on. In the second solution, described in Figure 4(c), we keep constantly the calculation of the vector of coefficients of the equalizer's FIR $\overline{W}$. After this, according to the output of the decision block, we turn off or on the equalizer' FIR. With this solution, we aim to avoid a suspected delay of the equalizer when it is re-launched especially in a channel that varies in time.

4.2. Decision-Making Result in a Time-Variable Channel. Since we are in a time-varying channel, the decision of the receiver to turn off or on the equalizer must follow in real time the variation of the channel's state. So we have to determine the constraint on the number of metrics' estimated values that allows to make decision without any delay. For this, we consider a mobile equipment with a speed equal to v, in a multipath channel with a Doppler frequency f_d described in (25):

$$f_d = \frac{v}{c} * f_p, \tag{25}$$

with c: the speed of light, f_p: the carrier frequency.

The time of coherence of the channel is defined as the period of time where the channel does not change, and it is expressed in

$$T_c = \frac{1}{f_d}. \tag{26}$$

In order to follow the variation of the channel, the decision time must be before the time of coherence. Assuming T_{fr} to be the transmission time of one frame, then the decision making must be done after a number of frames, F_r, which verifies the following condition:

$$F_r < \frac{T_c}{T_{\mathrm{fr}}}. \tag{27}$$

Table 3: Decided actions of reconfiguration.

State of the metrics	Decided actions
$\mathrm{SNR}_p < \lambda_{\mathrm{SNR}}$	Keep the equalizer
$\mathrm{SNR}_p \geq \lambda_{\mathrm{SNR}}$ $\mathrm{ISI} \leq \lambda_{\mathrm{ISI}}$	Disable the equalizer
$\mathrm{SNR}_p \geq \lambda_{\mathrm{SNR}}$ $\mathrm{ISI} > \lambda_{\mathrm{ISI}}$	Keep the equalizer

Table 4: Parameters of the channels.

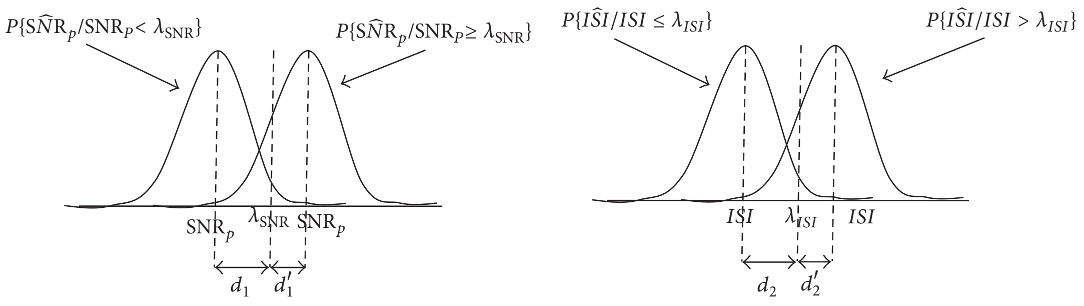
$T_{\mathrm{fr}} = 1$ ms, $f_p = 1800$ MHz			
Channel (Rice or Rayleigh)	f_d(Hz)	v(Km/h)	Condition on n
Slow channel	10	6	$n < 2000$
Fast channel	100	60	$n < 200$

If P is the number of the training symbols in one frame necessary for the metrics' estimations, then the number n of the estimated values considered for one decision is defined as in

$$n = \mathrm{PF}_r < P\frac{T_c}{T_{\mathrm{fr}}}. \tag{28}$$

For simulations, we consider a multipath channel variable in time with length equal to 5. We will consider first the case of a Rician channel and then the Rayleigh channel. For each case, we simulate the scenario of decision making to drive the equalizer by using first solution 1 and second solution 2 described in Figures 4(b) and 4(c). At the same time, we will take two examples of each channel: a slow varying channel and a fast varying channel. The conditions of simulation are described in Table 4.

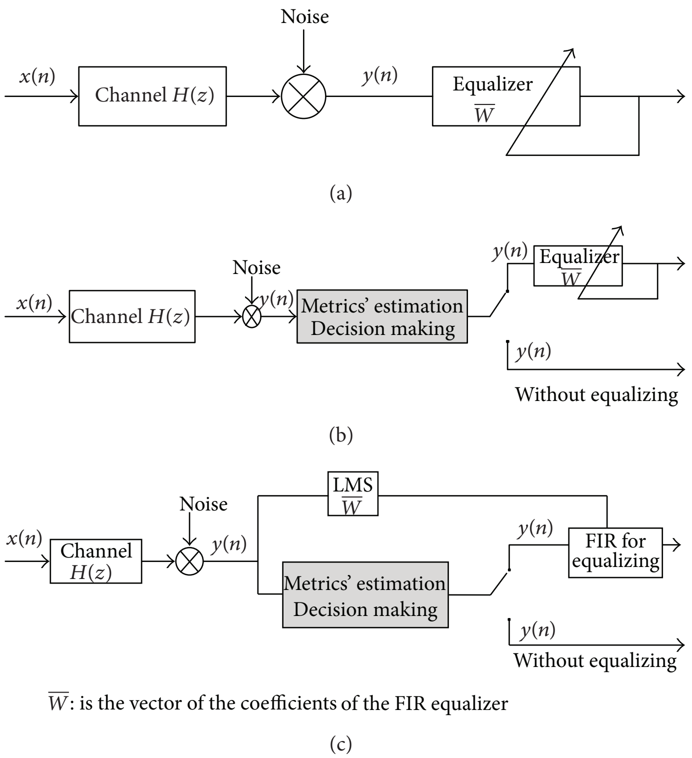
4.2.1. A Case of Rician Channel. The two examples of the Rician channel that we are using are represented in Figure 5 which describes the behavior of the channel's paths in time when it varies slowly (Figure 5(a)) and rapidly (Figure 5(b)). We present also in Figures 6(a) and 6(b) the bit error rate at the receiver, without the equalizer, of the two examples of the channel. The results given by the decision-making block are presented in Figures 7(a) and 7(b); the output of this block is "1" when the decision is to turn off the equalizer and "0" when the decision is to turn it on. The decision block starts to deliver the results after the reception of 3 frames, the time required for booting the system of decision. As we can see for the slow channel (Figure 7(a)), the decision is following the state of the channel described by the BER; the equalizer is turned off when the BER is less than or equal to 10^{-3} from the 4th received frame to the 41th received frame. We note also that the decision makes some false alarms especially for the frame numbers 42 and 43, in fact, although these frames are received by a BER equal to 10^{-3}, the decision is to turn on the equalizer. But these false alarms are not affecting the performance of the system, because the BER will be under 10^{-3}. Our decision method can also follow the change of


$P\{\widehat{\text{SNR}}_p/\text{SNR}_p < \lambda_{\text{SNR}}\}$
$P\{\widehat{\text{SNR}}_p/\text{SNR}_p \geq \lambda_{\text{SNR}}\}$
SNR_p
λ_{SNR}
SNR_p
d_1
d_1'
$P\{\widehat{ISI}/ISI \leq \lambda_{ISI}\}$
$P\{\widehat{ISI}/ISI > \lambda_{ISI}\}$
ISI
λ_{ISI}
ISI
d_2
d_2'

(a) Distribution of $\widehat{\text{SNR}}_p$

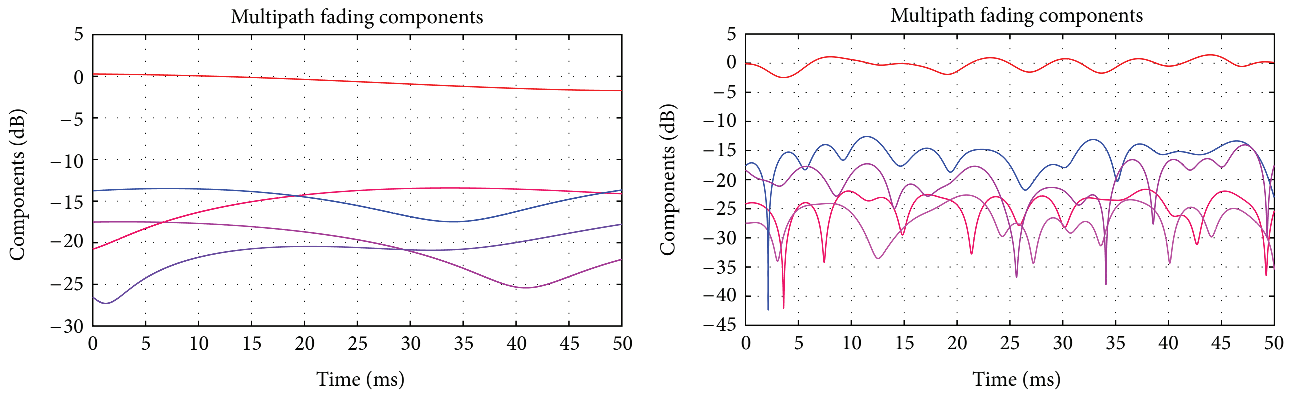
(b) Distribution of $\widehat{\text{ISI}}$

Figure 3: Distributions of the estimated metrics $\widehat{\text{SNR}}_p$ and $\widehat{\text{ISI}}$.



$\overline{W}$: is the vector of the coefficients of the FIR equalizer

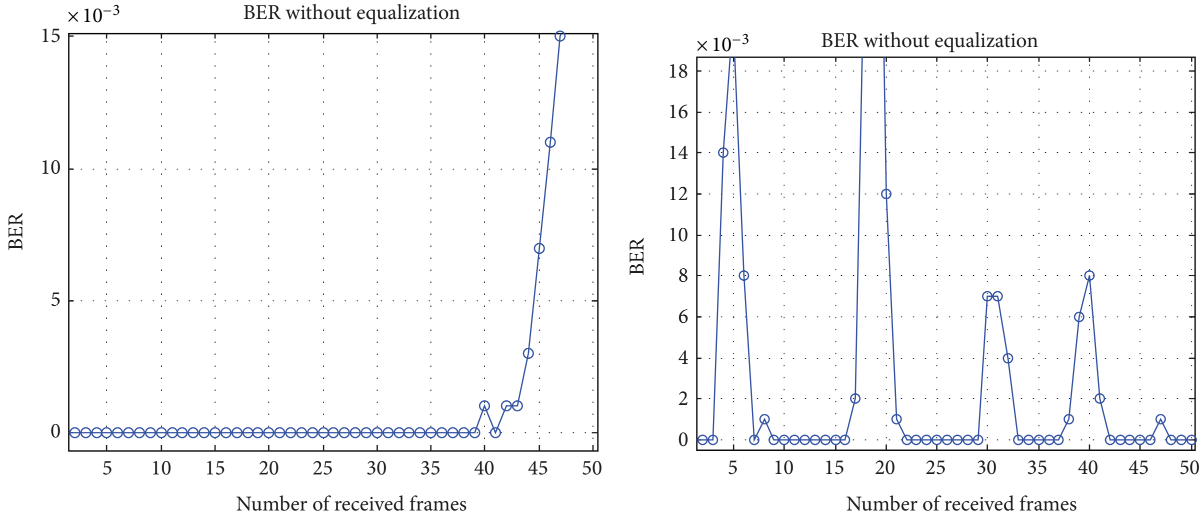
Figure 4: (a) Conventional equalizer, (b) solution 1, and (c) solution 2.



(a) Slow Rician channel

(b) Fast Rician channel

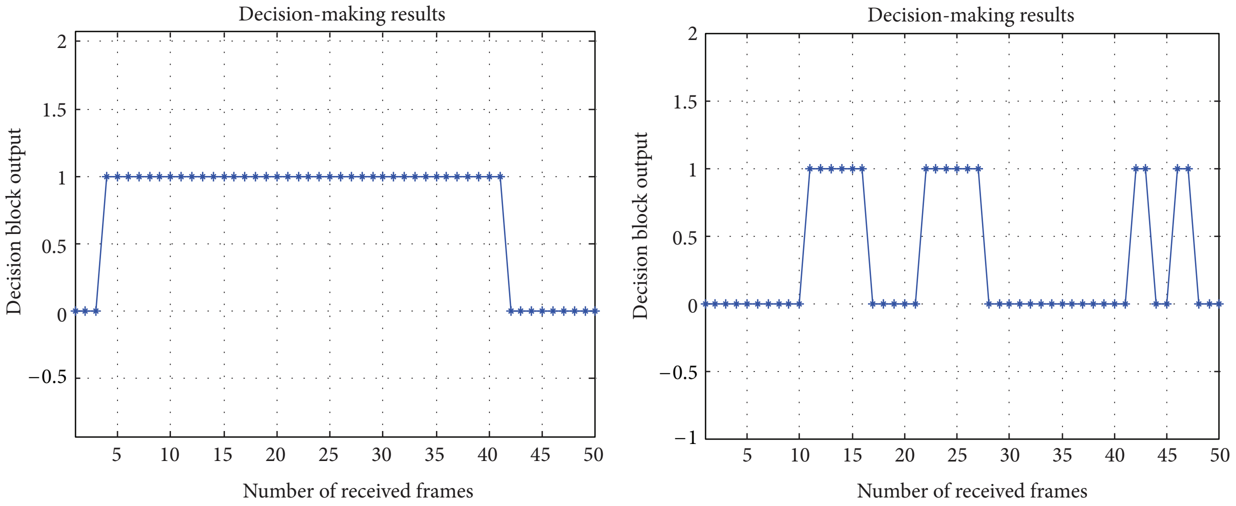
Figure 5: Variations of the Rician channel: change of the 5 paths in time.



(a) Slow Rician channel

(b) Fast Rician channel

Figure 6: BER without equalizer in the Rician channel.



(a) Slow Rician channel

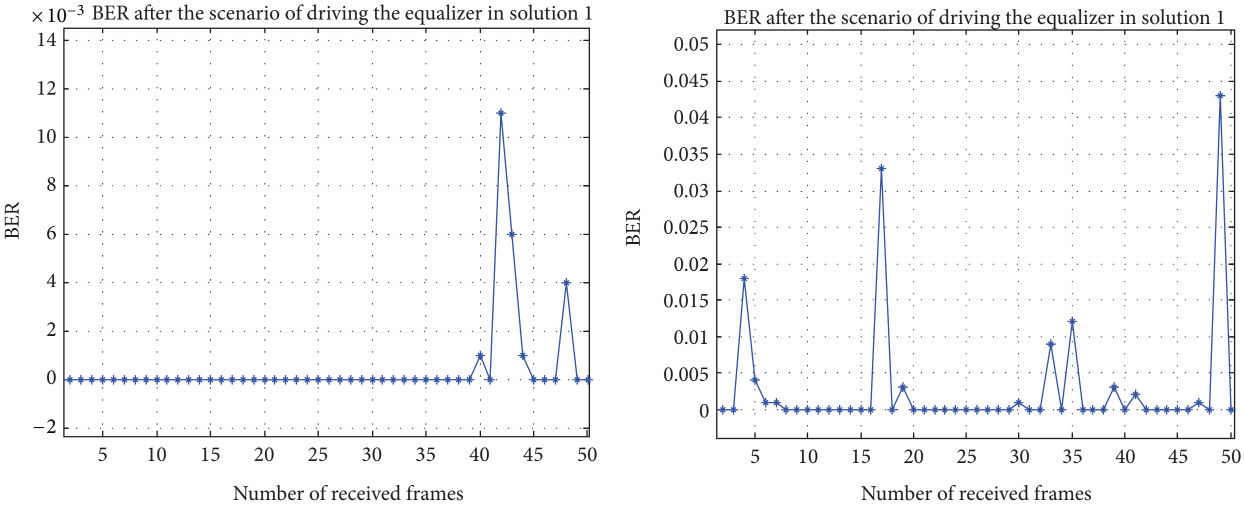
(b) Fast Rician channel

Figure 7: Decision-making results for the Rician channel; "1" → equalizer off, "0" → equalizer on.

the channel even if it is fast (Figure 7(b)), in fact this is because we are taking $n = 60$ which verifies the constraint (28).

The two solutions of configuring the decision scenario, discussed above, are providing the same decision results, since the same decision-making method is used. However, the equalization performance after the decision-making differs from one solution to another. We present in Figures 8 and 9 the BER after the decision making scenario with or without the equalizer for both solutions and for both examples of channels. We notice that the performance of the equalization in solution 2 is better than that in solution 1. Indeed, this is explained by the fact that in solution 1 when the equalizer is turned on, there is a period of time during which the calculation of the equalizer's coefficients $\overline{W}$ is launched. During this time, the channel changes and the equalization will not be efficient. This is why in Figures 8(a) and 8(b) the BER after equalization still degraded in solution 1, whereas for solution 2 (Figures 9(a) and 9(b)), the BER after equalization is always less than 10^{-3}.

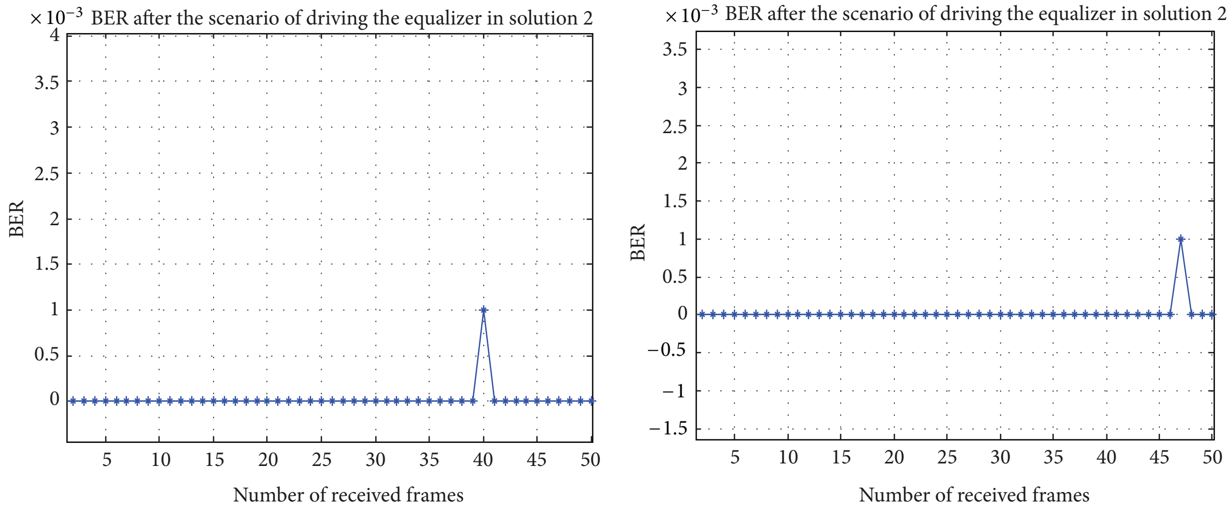
Although solution 1 of driving the equalizer is less efficient in a variable channel, it allows to reduce the computational complexity more than solution 2. In fact we present the results of the computational complexity reduction in Table 5 which describes for each example of channel and for each solution the percentage of the complexity reduction compared to the permanent use of the equalizer. We specify that, in determining the computational complexity of the scenario of driving the equalizer, we include the treatments



(a) Slow Rician channel

(b) Fast Rician channel

Figure 8: Performance of the system after decision making for the Rician channel with solution 1: errors of the equalizer due to its interruptions and reinitializations.



(a) Slow Rician channel

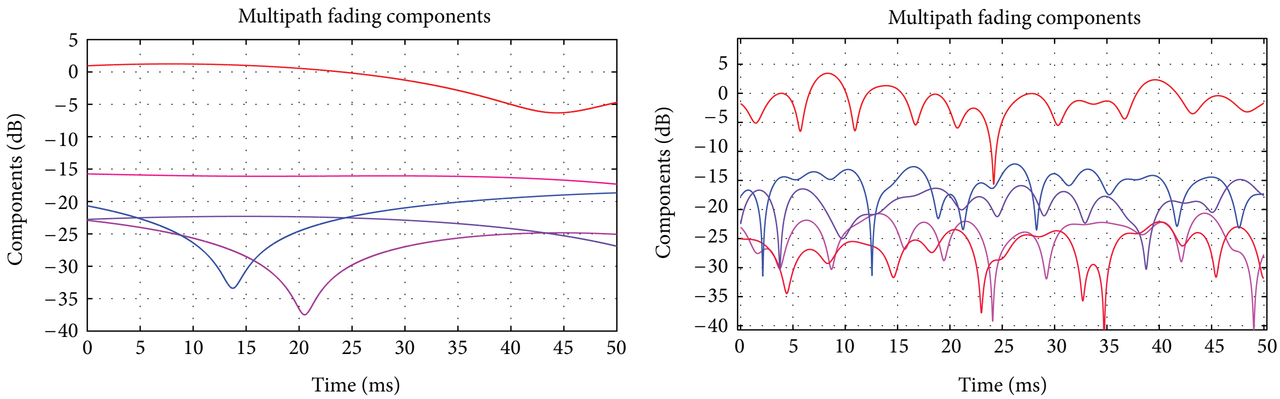
(b) Fast Rician channel

Figure 9: Performance of the system after decision making for the Rician channel with solution 2: no errors of the equalizer; the BER is always < 10^{-3}.

that we perform to make the decisions (estimation of the metrics, modeling, and decision rule).

4.2.2. A Case of a Rayleigh Channel. For the simulation of this scenario in the case of a Rayleigh channel, we assume the same conditions of mobility as described in Table 4 for the Rician channel. The two examples of the Rayleigh channel that we used are presented in Figure 10. As in the case of the Rician channel, the results of decision making are presented in Figures 11, 12, 13 and 14. From these figures, we can notice that for the same channel's mobility conditions, the Rayleigh channel is harder than the Rician channel; there is a less chance of turning off the equalizer in the Rayleigh case than in the Rice case. We can see this in Table 6 where the rate of time for turning off the equalizer is less than that of the Rician channel described in Table 5.

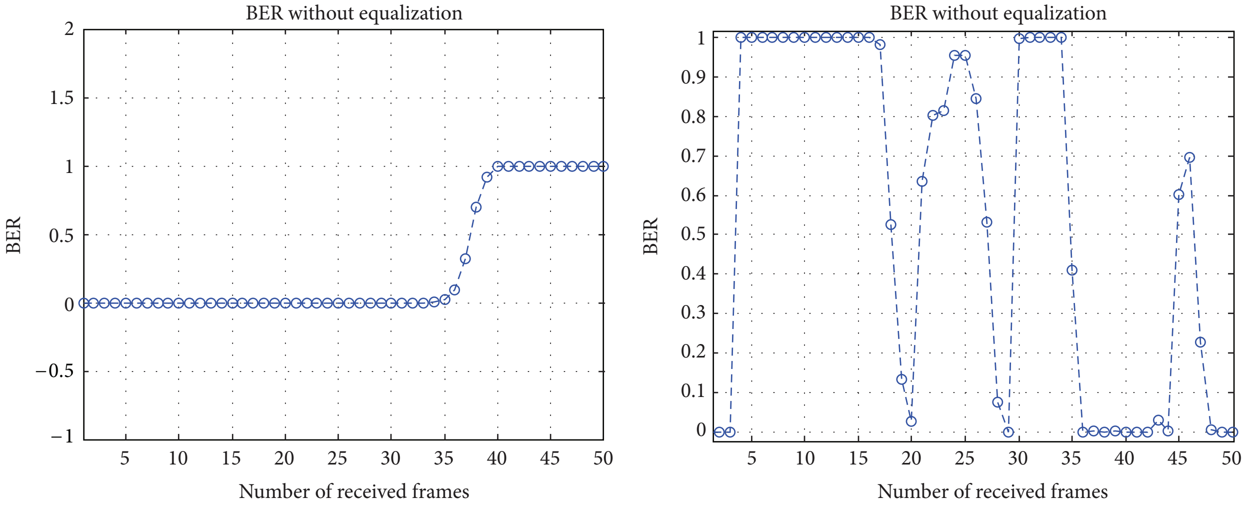
We notice in Figure 13(b) that the final BER is sometimes greater that 10^{-3}, but this is not due to an error of decision. In fact, if we observe clearly the curves, the BER is degraded while the equalizer functions (decision output = 0), this means that this bit error rate is not due to the inter-symbol interferences but rather to the strong attenuation of the path (SNR_p degraded). Despite this, the equalizer has contributed to the reduction of the BER even if it remains higher than 10^{-3}. This statement justifies the channel evaluation that we made in developing our decision making method (Table 3)



(a) Slow Rayleigh channel

(b) Fast Rayleigh channel

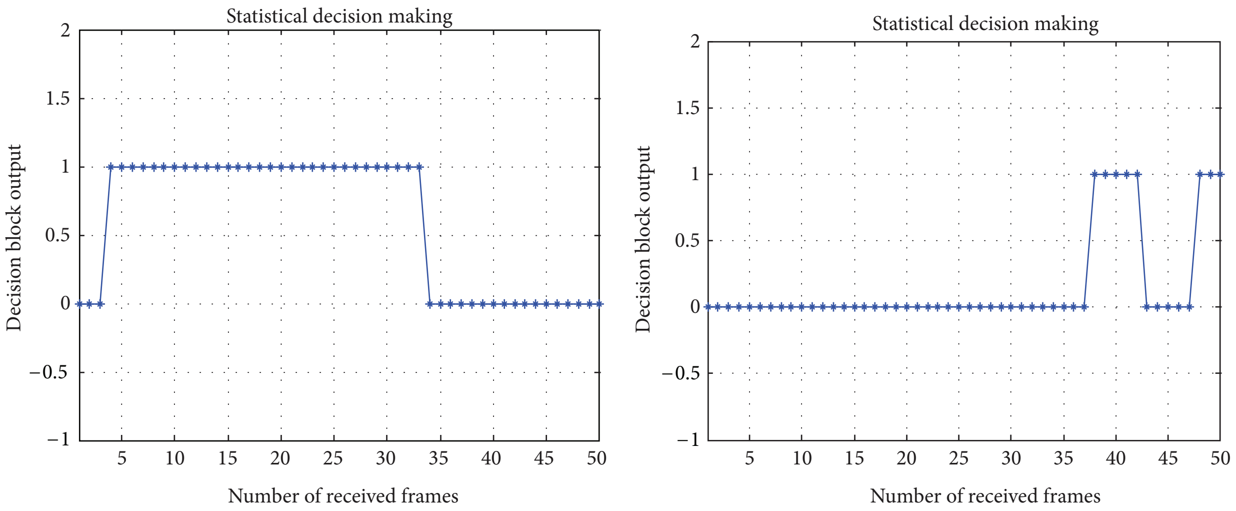
Figure 10: Variations of the Rayleigh channel: change of the 5 paths in time.



(a) Slow Rayleigh channel

(b) Fast Rayleigh channel

Figure 11: Variation of the BER at the receiver for the Rayleigh channel.



(a) Slow Rayleigh channel

(b) Fast Rayleigh channel

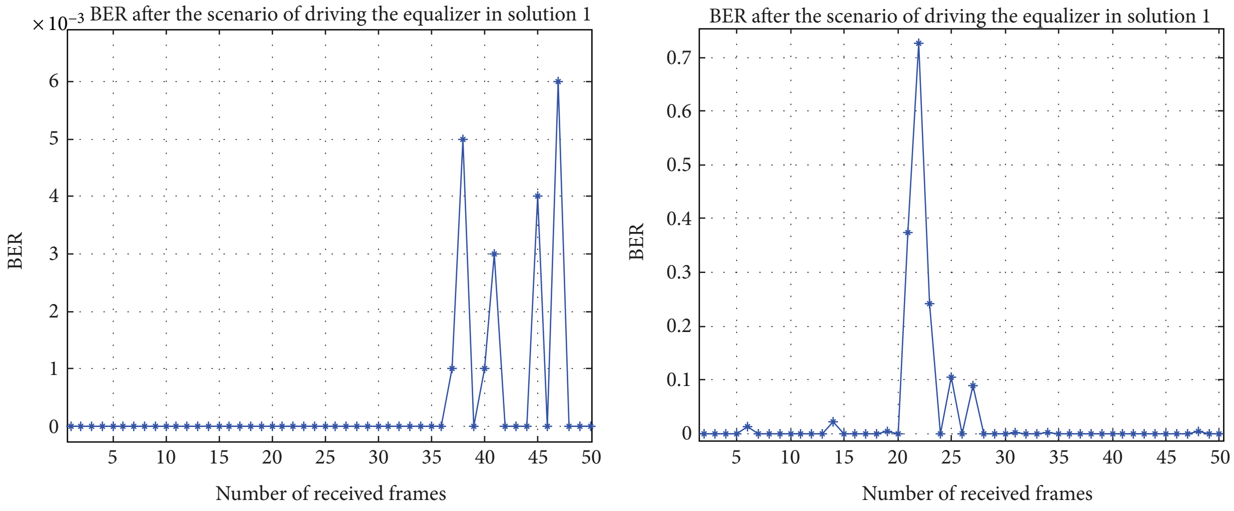
Figure 12: Decision-making results for the rayleigh channel; “1” → equalizer off, “0” → equalizer on.

Table 5: Computational complexity reduction for the rician channel.

	Solution 1		Solution 2	
	Slow channel	Fast channel	Slow channel	Fast channel
Rate of time of turning off the equalizer	80.8%	34.04%	80.8%	34.04%
Percentage of the complexity reduction	71.99%	25.31%	68.15%	23.53%

Table 6: Computational complexity reduction for the rayleigh channel.

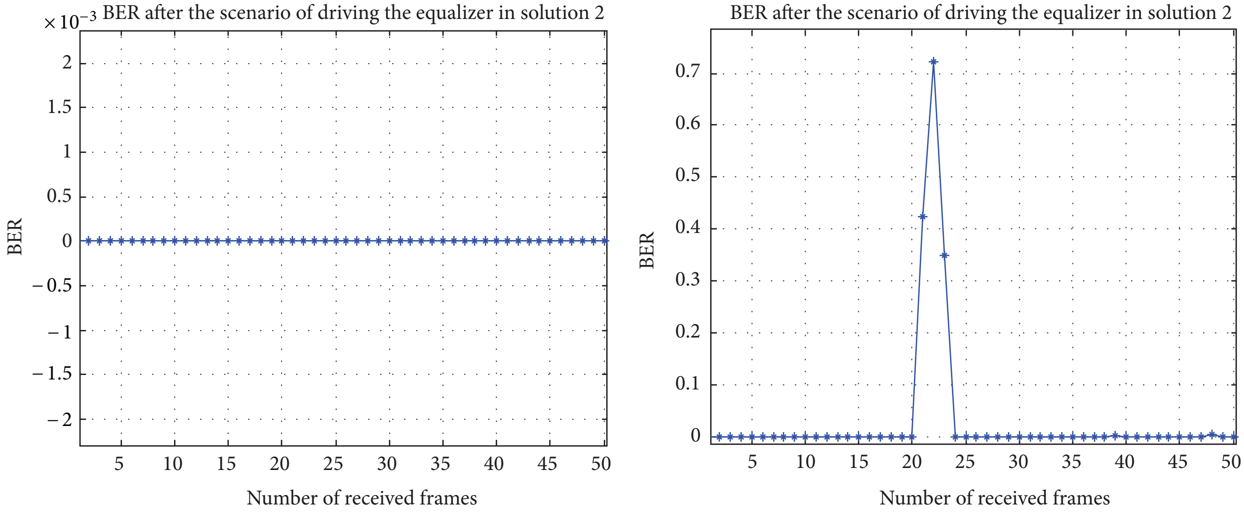
	Solution 1		Solution 2	
	Slow channel	Fast channel	Slow channel	Fast channel
Rate of time of turning off the equalizer	63.82%	17.02%	63.82%	17.02%
Percentage of the complexity reduction	55.01%	8.34%	51.91%	7.31%



(a) Slow Rayleigh channel

(b) Fast Rayleigh channel

Figure 13: Performance of the system after decision making for the rayleigh channel with solution 1: errors of the equalizer due to its interruptions and reinitializations.



(a) Slow Rayleigh channel

(b) Fast Rayleigh channel

Figure 14: Performance of the system after decision making for the rayleigh channel with solution 2.

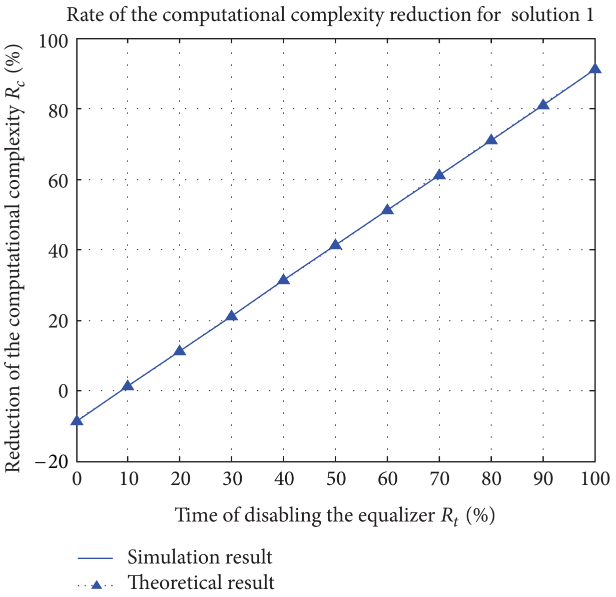


FIGURE 15: Computational complexity reduction with solution 1.

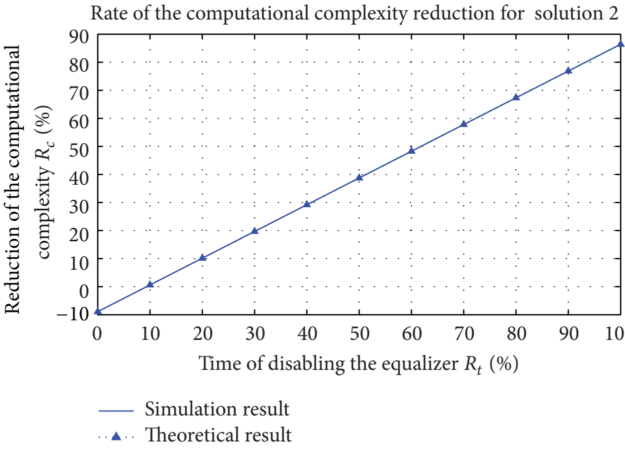


FIGURE 16: Computational complexity reduction with solution 2.

concerning the act of keeping the equalizer when the signal to noise ratio is degraded.

4.3. Computational Complexity Reduction. We are interested in this section to the study of the computational complexity reduction through the scenario of driving the equalizer. The goal is to determine, for a period of time of a communication, from what percentage of time without the equalizer we begin to reduce the complexity, and also what is the maximum gain in complexity. For this, we will first determine the theoretical equation of the computational complexity reduction, and then we will compare this with the result of simulation.

We denote

(i) C_{eq}: the computational complexity of the permanent use of the equalizer,

(ii) C_{tr}: the computational complexity of the added treatments necessary for the decision making method (metrics estimation, modeling, decision rule); this value depends on the width of the window of observations, n, considered for the decision,

(iii) C_d: the total computational complexity of the scenario of driving the equalizer,

(iv) R_t: the rate of time during which the equalizer is turned off

(v) R_c: the rate of the computational complexity reduction.

In a period of time of a communication, during which the equalizer is turned off with a percentage of time of R_t, the total computational complexity C_d is reduced by R_c compared with the complexity of the permanent use of the equalizer. So we can express C_d as follows:

$$C_d = (1 - R_c) C_{\text{eq}}. \tag{29}$$

On the other hand, the total complexity of the scenario contains the complexity of the added treatments of the decision-making method and the complexity of the equalizer when it is turned on. So we can also express C_d differently as in

$$C_d = C_{\text{tr}} + (1 - R_t) C_{\text{eq}}. \tag{30}$$

From (29) and (30), we conclude that

$$(1 - R_c) C_{\text{eq}} = C_{\text{tr}} + (1 - R_t) C_{\text{eq}}. \tag{31}$$

We deduce then the expression of the complexity reduction as follows:

$$R_c = R_t - \frac{C_{\text{tr}}}{C_{\text{eq}}}. \tag{32}$$

From this equation, we can conclude that we begin to reduce the complexity ($R_c > 0$) when

$$R_t > \frac{C_{\text{tr}}}{C_{\text{eq}}}. \tag{33}$$

We notice also that the maximum reduction of the complexity for the considered communication is

$$R_{c\max} = 1 - \frac{C_{\text{tr}}}{C_{\text{eq}}}. \tag{34}$$

We plot in Figures 15 and 16 the curves of $R_c = f(R_t)$ theoretically as in (32) and by simulations, first in the case of solution 1 and then in the case of solution 2.

For solution 1, we begin to reduce the complexity of the receiver when we turn off the equalizer during at least 8.66% of the communication period, whereas for solution 2 this value is 9.36%. When the receiver decides to turn off the equalizer during the whole communication period, it can reduce the complexity by 91.1% in solution 1 and 86.4% in solution 2. As a result, solution 1 is better in terms of computational complexity reduction; however, the equalization's performance in this solution is less efficient than that in the case of solution 2 mainly when the channel varies quickly.

5. Conclusion

The work presented in this paper is a part of the green communication, it consists of reducing energy consumption within the receiver by making it able to adapt dynamically to the changes of its environment. For this, we are particularly interested in adapting the use of the equalizer in the receiver chain according to the state of the channel. In fact, it is to make the receiver able to choose to turn on or off the equalizer as it is necessary or not. The purpose of dispensing with this component, for a period of time, was to reduce the energy consumed in the receiver chain. To achieve this objective, it was necessary that we develop a method of decision making for the receiver so it can be aware of its environment and able to make the right decision concerning the presence of the equalizer. So we defined a technique of decision making based on statistical modeling of the environment. Within this method, we defined two metrics as channel quality indicators (SNR_p and ISI) used to evaluate the state of the channel with respect to the intersymbol interferences and the channel fading. These metrics are then statistically modeled by determining their respective densities of probability from the sets of their estimated values. This statistical model is used in order to construct a statistical decision rule to decide to turn off the equalizer. This decision rule is defined to take into account the impact of the uncertain measures of the metrics on the decision thresholds. Once the decision method developed, we applied it to adapt the use of the equalizer by proposing two solutions to drive this component. In the first solution, the receiver decides to turn off the equalizer, according to the result of the decision rule, and decides to turn it on by restarting it. While in the second solution, we kept constantly the calculation of the equalizer's coefficients. By adopting the adaptive algorithm of channel estimation LMS, we have simulated these two solutions in the case of a time-varying multipath channel (Rice and Rayleigh). We have concluded, first, that the rate of reduction of the computational complexity within the limitation of the equalizer use is a linear function of the percentage of time during which this component is off. Second, we have noticed that the first solution to adapt the use of the equalizer reduces the complexity more than the second solution; in fact, for a period of time according to the reception of 50 frames, we begin to reduce the computational complexity when the equalizer is off for at least 8.66% of the total period, whereas this percentage is 9.36% for the second solution. However, the equalizer in the second solution is more efficient, in fact, unlike the first solution, when this component is turned on it is not restarted because of the permanent presence of the equalizer's coefficients, this allows to avoid the delay in this operation when it begins to operate, especially when the channel varies rapidly. We notice also that the smaller the size of the equalizer, the less the delay caused by the calculation of the coefficients, and so the gap between the two solutions will be reduced. Within the same goal of reducing energy consumption in the radio receiver, we seek, in a future work, to treat other situations, of the adaptation of the receiver to the environment, that allow to reduce the computational complexity of the receiver chain.

References

[1] J. Palicot, "Cognitive radio: an enabling technology for the green radio communications concept," in *Proceedings of the ACM International Wireless Communications and Mobile Computing Conference (IWCMC '09)*, pp. 489–494, June 2009.

[2] H. L. Kazecki, S. H. Goode, D. W. Denis, J. C. Baker, K. L. Baum, and B. D. Mueller, *Method for Channel Adaptive Detecting/Equalize*, 1993.

[3] Su et al., "Radio receiver with selectively disabled equalizer," 2009, US007627030B2.

[4] Furya et al., "Low power consumption receiver with adaptive equalizer," 1994, US005363411A.

[5] L. Husson and J. C. Dany, "A new method for reducing the power consumption of portable handsets in TDMA Mobile systems: conditional equalization," *IEEE Transactions on Vehicular Technology*, vol. 48, no. 6, pp. 1936–1945, 1999.

[6] L. Husson, *Evaluation par le recepteur de la qualite du signal reu dans les systemes de radiocommunication avec les mobiles et ameliorations des performances par l egalisation conditionnelle [Ph.D. thesis]*, University of Paris XI, France, 1998.

[7] Black, M. A. Howard et al., "Communication receiver with hybrid equalizer," 2010, US007646802B2.

[8] Porter et al., *Receiver having Equalizing Demodulator and Non-Equalizing Demodulator and Method for Controlling the Same*, 2006.

[9] I. J. Myung, "Tutorial on maximum likelihood estimation," *Journal of Mathematical Psychology*, vol. 47, no. 1, pp. 90–100, 2003.

[10] M. Lejeune, *Statistique: La Thorie et Ses Applications*, Springer, 2004.

[11] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, Prentice-Hall, 1993.

[12] G. Montalbano, "SIR Estimation techniques," 2011, US, 7941099B2.

[13] S. Borching, *Blind channel esatimation using redundant precoding: new algorithms, analysis and theory [Ph.D. thesis]*, California Institute of Technology, 2008.

[14] C. R. Johnson Jr., P. Schniter, T. J. Endres, J. D. Behm, D. R. Brown, and R. A. Casas, "Blind equalization using the constant modulus criterion: a review," *Proceedings of the IEEE*, vol. 86, no. 10, pp. 1927–1949, 1998.

[15] S. Bourbia, D. Le Guennec, K. Grati, and A. Ghazel, "Statistical decision making method for cognitive radio," in *Proceedings of the International Conference on Telecommunications (ICT '12)*, 2012.

Hindawi Publishing Corporation
Journal of Computer Networks and Communications
Volume 2012, Article ID 427246, 10 pages
doi:10.1155/2012/427246

Research Article

PCA-Guided Routing Algorithm for Wireless Sensor Networks

Gong Chen, Liansheng Tan, Yanlin Gong, and Wei Zhang

Department of Computer Science, Central China Normal University, Wuhan 430079, China

Correspondence should be addressed to Liansheng Tan, lianshengtan688@gmail.com

Received 13 August 2012; Revised 24 October 2012; Accepted 11 November 2012

Academic Editor: Hongxiang Li

An important performance concern for wireless sensor networks (WSNs) is the total energy dissipated by all the nodes in the network over the course of network lifetime. In this paper, we propose a routing algorithm termed as PCA-guided routing algorithm (PCA-RA) by exploring the principal component analysis (PCA) approach. Our algorithm remarkably reduces energy consumption and prolongs network lifetime by realizing the objective of minimizing the sum of distances between the nodes and the cluster centers in a WSN network. It is demonstrated that the PCA-RA can be efficiently implemented in WSNs by forming a nearly optimal K-means-like clustering structure. In addition, it can decrease the network load while maintaining the accuracy of the sensor measurements during data aggregating process. We validate the efficacy and efficiency of the proposed algorithm by simulations. Both theoretical analyses and simulation results demonstrate that this algorithm can perform significantly with less energy consumption and thus prolong the system lifetime for the networks.

1. Introduction

Wireless sensor networks (WSNs) [1] consist of battery-powered nodes which inherit sensing, computation, and wireless communication capabilities. Although there have been significant improvements in processor design and computing issues, limitations in battery provision still exist, bringing energy resource considerations as the fundamental challenge in WSNs. Consequently, there have been active research efforts devoted to lifting the performance limitations of WSNs. These performance limitations include network throughput, energy consumption and, network lifetime. Network throughput typically refers to the maximum amount of packets that can be successfully collected by the cluster heads (CHs) in the network, energy consumption refers to the minimize energy dissipation that nodes in the network consume, and network lifetime refers to the maximum time limit that nodes in the network remain alive until one or more nodes drain up their energy.

The routing algorithms have been specifically designed for WSNs because the energy optimization is an essential design issue. A good routing scheme is helpful in improving these performance limits such as reducing the energy consumption, prolonging the network lifetime, and increasing the network throughput. Network researchers have studied a great variety of routing protocols in WSNs differing based on the application and network architecture. As demonstrated in [2, 3], it can be classified into four categories: flit, hierarchical clustering, location-based routing, and QoS-based routing. The current routing protocols have their own design trade-offs between energy and communication overhead savings, as well as the advantages and disadvantages of each routing technique.

As per the representative hierarchical clustering protocol, low energy adaptive clustering hierarchy (LEACH) [4] has simplicity, flexibility, and scalability because its manipulations rely on randomized rotation of the cluster heads (CHs), but its features of unregulated distribution, unbalanced clustering structure, uniform initial energy, and so on, hinder its performance. Based on LEACH, there are many variants, such as [5–8]. LEACH-E [5] more likely selects the nodes with higher energy as the CHs. LEACH-C [5] analytically determines the optimum number of CHs by taking into account the energy spent by all clusters. PEGASIS [6], TEEN [7], and ATEEN [8] improve the energy consumption by optimizing the data transmission pattern. HEED [9] is a complete distributed routing protocol which has different clustering formations and cluster-heads selecting measures. These protocols have many restrictive assumptions and applicable limitations, so it has great improvement space and extensibility. The rapid development of wireless communications technology, and the miniaturization and low cost of sensing devices, have accelerated the development of wireless sensor networks (WSNs) [10, 11]. As in [12], Zytoune et al. proposed a uniform balancing energy routing protocol (UBERP). The BP K-means [13] and BS K-means [14] can improve the structure of clusters and perform better load-balance and less energy consumptions. HMP-RA [15] proposes a solution to address this issue through a hybrid approach that combines two routing strategies, flat multihop routing and hierarchical multihop routing. ESCFR and DCFR can map small changes in nodal remaining energy to large changes in the function value and consider the end-to-end energy consumption and nodal remaining energy [16]. Biologically inspired intelligent algorithms build a hierarchical structure on the network for different kinds of traffic, thus maximizing network utilization, while improving its performance [17]. In the case where sensor nodes are mobile, as in [18, 19], the nodes can adjust their position to help balance energy consumption in areas that have high transmission load and/or mitigate network partition.

In this paper, we consider an overarching algorithm that encompasses both performance metrics. It desires to minimize the sum of distances in the clusters. We show that the principal component analysis (PCA) [20], a useful statistical technique that has found application in fields such as face recognition and image compression, and a common technique for finding patterns in data of high dimension, can form a near-optimal K-means-like clustering structure, in which the distance between the non-CH nodes and CHs is near minimized.

Moreover, the data aggregating issue associated with the measurements accuracy calls for a careful consideration in scheme about data collecting and fusing. In this paper, we investigate the PCA technology in a high relative measurements context for WSNs. Our objective is to obtain a good approximation to sensor measurements by relying on a few principal components while decreasing the network load.

The remainder of this paper is organized as follows. In Section 2, we describe the network and energy model. Section 3 presents the PCA-guided routing algorithm (PCA-RA) model and gives numerical results to demonstrate the working mechanism of PCA-RA. Section 4 discusses the PCA-RA algorithm solution strategies. In Section 5, we simulate the PCA-RA and compare it with LEACH and LEACH-E. Finally, Section 6 concludes this paper.

2. Syestem Model

2.1. The Network Model. Let us consider a two-tier architecture for WSNs. Figure 1 shows the physical network topology for such a network. There are three types of nodes in the networks, namely, a base station (BS), the cluster-head nodes (CHNs), and the ordinary sensor nodes (OSNs).

For each cluster of sensor nodes, there is one a CHN, which is different from an OSN in terms of functions. The primary functions of a CHN are (1) data aggregation for data measurements from the local clusters of OSNs and (2) relaying the aggregated information to the BS. For data fusion, a CHN analyzes the content of each measurement it receives and exploits the correlation among the data measurements. An CHN has a limited lifetime, so we need consider rotating the CHNs to balance to the energy consumption.

The third component is the BS. We assume that there is sufficient energy resource available at the BS and thus there is no energy constraint at the BS.

2.2. The Energy Consumption Model. We compute the energy consumption using the first-order radio model [5]. The equations, which are used to calculate transmission costs and receiving costs for an L-bit message to cross a distance d, are shown below,

$$E_{\mathrm{TX}}(L,d) = \begin{cases} L\left(E_{\mathrm{elec}} + \varepsilon_{\mathrm{fs}} d^2\right), & d < d_0 \\ L\left(E_{\mathrm{elec}} + \varepsilon_{\mathrm{mp}} d^4\right), & d \geq d_0, \end{cases} \tag{1}$$

$$E_{\mathrm{RX}}(L) = LE_{\mathrm{elec}}.$$

In (1), the electronics energy, E_{elec}, depends on factors such as the digital coding, modulation, filtering, and spreading of the signal, whereas the amplifier energy, $\varepsilon_{\mathrm{fs}} d^2$ or $\varepsilon_{\mathrm{mp}} d^4$, depends on the distance to the receiver and the acceptable bit-error rate. d_0 is the distance threshold.

3. PCA-Guided Routing Algorithm Model

PCA is a classic technique in statistical data analysis, data compression, and image processing. PCA transforms a number of correlated variables into a number of uncorrelated variables called principal components. The objective of PCA is to reduce the dimensionality of the dataset, but not only retain most of the original variability in the data. The first principal component accounts for as much of the variability in the data as possible. Mathematically, how to pick up the dimensions with the largest variances is equivalent to finding the best low-rank approximation of the data via the singular value decomposition (SVD) [21].

The design of routing algorithm is important in wireless sensor networks. Although plenty of interests are drawn on it, there is still a challenge to face on the aspect of efficiency and energy consumption. In this section, we will describe the notations about PCA-guided routing algorithm model firstly and then propose the PCA-guided clustering model, and finally, we will present the PCA-guided data aggregating model.

3.1. Notations. Let $X = \{x_1, x_2, \ldots, x_n\}$ represents the location coordinate matrix of a set of n sensors, $Y = \{y_1, y_2, \ldots, y_n\}$ represents the centered data matrix, where

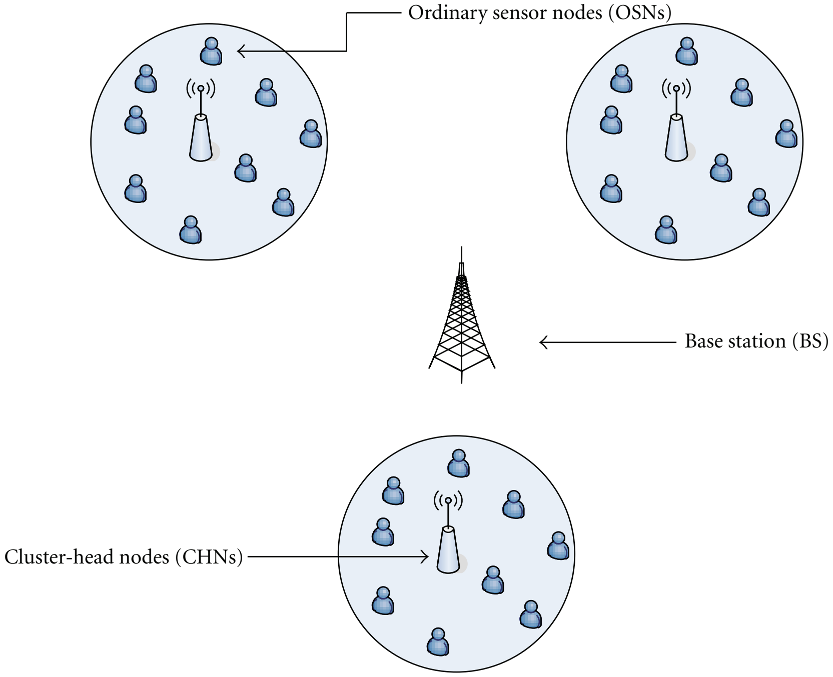


Figure 1: Physical topology for two-tier wireless sensor networks.

$y_i = x_i - \overline{x}$, which defines the centered distance vector column wise, and $\overline{x} = \sum_i x_i/n$ is the mean vector column wise of X matrix.

Let $M = \{m_1, m_2 \cdots m_p\}$ be a group of measurements collecting from the sampling period. Each sensor generates a stream of data. Let $D_{N\times P}$ be a matrix with elements D_{ij}, $1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant p$, being the measurement taken by sensor i at point j. Let $Q_{N\times P}$ be a centered matrix with elements $q_{ij} = d_{ij} - d_{i.}/n$, $d_{i.} = \sum_j d_{ij}$.

3.2. PCA-Guided Clustering Model. We define the equation for the SVD of matrix Y [22] as follows:

$$Y = \sum_k \lambda_k u_k v_k. \tag{2}$$

The covariance matrix (ignoring the factor $1/n$) is

$$\sum_i (x_i - \overline{x})^T (x_i - \overline{x}) = Y^T Y. \tag{3}$$

The principal components v_k are eigenvectors satisfying

$$Y^T Y v_k = \lambda_k^2 v_k, \qquad v_k = \frac{Y^T u_k}{\lambda_k^2}. \tag{4}$$

3.2.1. K-Means Clustering Model. According to [23, 24], we find the PCA dimension reduction automatically by performing data clustering according to the K-means objective function [25, 26]. Using K-means algorithm, it can form a better cluster structure by minimizing the sum of squared errors. We define the squared distance between sensor nodes and cluster centers as

$$J_K = \sum_{k=1}^{K} \sum_{i\in C_k} (x_i - m_k)^2, \tag{5}$$

where $m_k = \sum_{x_i \in C_k} x_i/n_k$ is the center of cluster C_k and n_k is the number of sensor nodes in C_k. Given the fact that by minimizing the distance between sensor nodes and cluster centers, the energy consumption can be effectively reduced. Our clustering algorithm is thus designed to be capable of minimizing the above metric J_K.

For the sake of convenience, let us start with the case of $K = 2$. To obtain the explicit expression for J_k, letting

$$d\left(C_p, C_l\right) = \sum_{i\in C_p} \sum_{j\in C_l} \left(x_i - x_j\right)^2 \tag{6}$$

be the sum of squared distances between two clusters C_p and C_l, after some algebra one obtains the following:

$$J_2 = \frac{d(C_1, C_1)}{2n_1} + \frac{d(C_2, C_2)}{2n_2}, \tag{7}$$

where n_1 and n_2 are the numbers of sensor nodes in C_1 and C_2, n is the total number of sensor nodes; therefore, we get $n = n_1 + n_2$.

If denoting

$$\overline{y^2} = \frac{\sum_i y_i^T y_i}{n} = \frac{d(C_1, C_1)}{2n^2} + \frac{d(C_2, C_2)}{2n^2} + \frac{d(C_1, C_2)}{n^2}, \tag{8}$$

$$J_D = \frac{n_1 n_2}{n} \left[2\frac{d(C_1, C_2)}{n_1 n_2} - \frac{d(C_1, C_1)}{n_1^2} - \frac{d(C_2, C_2)}{n_2^2} \right], \tag{9}$$

we thus have

$$\begin{aligned}
n\overline{y^2} - \frac{1}{2}J_D &= n\left(\frac{d(C_1,C_1)}{2n^2} + \frac{d(C_2,C_2)}{2n^2} + \frac{d(C_1,C_2)}{n^2}\right) \\
&\quad - \frac{1}{2}\left(\frac{n_1 n_2}{n}\left[2\frac{d(C_1,C_2)}{n_1 n_2} - \frac{d(C_1,C_1)}{n_1^2} \right.\right. \\
&\qquad \left.\left. - \frac{d(C_2,C_2)}{n_2^2}\right]\right) \\
&= \frac{d(C_1,C_1)}{2n} + \frac{d(C_2,C_2)}{2n} + \frac{d(C_1,C_2)}{n} \\
&\quad - \frac{2n_1 n_2 d(C_1,C_2)}{2nn_1 n_2} + \frac{n_1 n_2 d(C_1,C_1)}{2nn_1^2} \\
&\quad + \frac{n_1 n_2 d(C_2,C_2)}{2nn_2^2} \\
&= \left(\frac{d(C_1,C_1)}{2n} + \frac{n_1 n_2 d(C_1,C_1)}{2nn_1^2}\right) \\
&\quad + \left(\frac{d(C_2,C_2)}{2n} + \frac{n_1 n_2 d(C_2,C_2)}{2nn_2^2}\right) \\
&\quad + \left(\frac{d(C_1,C_2)}{n} - \frac{2n_1 n_2 d(C_1,C_2)}{2nn_1 n_2}\right) \\
&= \left(\frac{n_1 d(C_1,C_1)}{2nn_1} + \frac{n_2 d(C_1,C_1)}{2nn_1}\right) \\
&\quad + \left(\frac{n_2 d(C_2,C_2)}{2nn_2} + \frac{n_1 d(C_2,C_2)}{2nn_2}\right) \\
&= \frac{(n_1+n_2)d(C_1,C_1)}{2nn_1} + \frac{(n_2+n_1)d(C_2,C_2)}{2nn_2} \\
&= \frac{d(C_1,C_1)}{2n_1} + \frac{d(C_2,C_2)}{2n_2} = J_2.
\end{aligned} \tag{10}$$

That is

$$J_2 = n\overline{y^2} - \frac{1}{2}J_D, \tag{11}$$

where $\overline{y^2}$ is a constant and it denotes the distance between the sensor nodes and the center for all nodes; thus min (J_K) is equivalent to max (J_D), and because the two resulting clusters are as separated and compact as possible. Because the averaged intracluster distance is greater than the sum of the averaged intercluster distances, that is

$$\frac{d(C_1,C_2)}{n_1 n_2} - \frac{d(C_1,C_1)}{n_1^2} - \frac{d(C_2,C_2)}{n_2^2} > 0. \tag{12}$$

From (9), it is seen that J_D is always positive. This is to say, evidenced from (11), for $K = 2$ minimization of cluster objective function J_K is equivalent to maximization of the distance objective J_D, which is always positive.

When $K > 2$, we can do a hierarchical divisive clustering, where each step using the $K = 2$ is a clustering procedure. This procedure can get an approximated K-means clustering structure.

3.2.2. PCA-Guided Relaxation Model. In [24], it proves that the relaxation solution of J_D can get via the principal component. It sets the cluster indicator vector to be

$$q(i) = \begin{cases} \sqrt{\dfrac{n_2}{nn_1}}, & \text{if } i \in C_1 \\ -\sqrt{\dfrac{n_1}{nn_2}}, & \text{if } i \in C_2. \end{cases} \tag{13}$$

The indicator vector satisfies the sum-to-zero and normalization conditions. Consider the squared distance matrix $H = (h_{ij})$, where $h_{ij} = \|x_i - x_j\|^2$. $q^T H q = -J_D$ is easily observed.

(1) *The First Relaxation Solution.* Let q take any value in $[-1, 1]$; the solution of minimization of $J(q) = q^T Hq/q^T q$ is given by the eigenvector corresponding to the lowest eigenvalue of the equation $Hz = \lambda z$.

(2) *The Second Relaxation Solution.* Let $\widehat{H} = (\widehat{h}_{ij})$, where the element is given by

$$\widehat{h}_{ij} = h_{ij} - \frac{h_{i.}}{n} - \frac{h_{.j}}{n} + \frac{h_{..}}{n^2}, \tag{14}$$

in which, $h_{i.} = \sum_j h_{ij}$, $h_{.j} = \sum_i h_{ij}$, $h_{..} = \sum_{ij} h_{ij}$.

After computing, we have $q^T \widehat{H} q = q^T H q = -J_D$, then relaxing the restriction q, the desired cluster indicator vector is the eigenvector corresponding to the lowest eigenvalue of $\widehat{H}z = \lambda z$.

(3) *The Third Relaxation Solution.* With some algebra, we can obtain $\widehat{H} = -2Y^T Y$. Therefore, the continuous solution for cluster indicator vector is the eigenvector corresponding to the largest eigenvalue of the covariance matrix $Y^T Y$ which by definition, is precisely the principal component v_1.

3.2.3. PCA-Guided Clustering Model. According to the mentioned above, for K-means clustering where $K = 2$, the continuous solution of the cluster indicator vector is the principal component v_1, that is, clusters C_1 and C_2 are given by

$$C_1 = \{i \mid v_1(i) \le 0\}, \qquad C_2 = \{i \mid v_1(i) > 0\}. \tag{15}$$

We can consider using PCA technology to clustering sensor nodes for WSNs. It near minimizes the sum of the distances between the sensor nodes and cluster centers.

Example 1. Let us assume in a WSN there are 20 sensors distributed in the network. X represents the 2D coordinate matrix.

$$\begin{aligned} X = \Big[\ & 29.471,\ 4.9162,\ 69.318,\ 65.011,\ 98.299,\\ & 55.267,\ 40.007,\ 19.879,\ 62.52,\ 73.336,\\ & 37.589,\ 0.98765,\ 41.986,\ 75.367,\ 79.387,\\ & 91.996,\ 84.472,\ 36.775,\ 62.08,\ 73.128;\\ & 19.389,\ 90.481,\ 56.921,\ 63.179,\ 23.441,\\ & 54.878,\ 93.158,\ 33.52,\ 65.553,\ 39.19,\\ & 62.731,\ 69.908,\ 39.718,\ 41.363,\ 65.521,\\ & 83.759,\ 37.161,\ 42.525,\ 59.466,\ 56.574\ \Big]. \end{aligned} \tag{16}$$

Compute the eigenvector of the matrix, Y^TY, that is, the principal component v_1:

$$\begin{aligned} v_1^T = \Big[\ & -0.11623,\ -0.47282,\ 0.10624,\ 0.058301,\\ & 0.40896,\ 0.0013808,\ -0.20539,\ -0.22552,\\ & 0.033439,\ 0.17823,\ -0.15446,\ -0.45626,\\ & -0.067391,\ 0.18908,\ 0.16501,\ 0.22147,\\ & 0.2697,\ -0.11445,\ 0.04397,\ 0.13673\ \Big]. \end{aligned} \tag{17}$$

If $v_1(i) \leqslant 0$, sensor i belongs to C_1, otherwise it belongs to C_2. We depicted the above clustering results into Figure 2.

When $K > 2$, we do a hierarchical divisive clustering, where each step uses the $K = 2$ clustering procedure.

In summary, a PCA-guided clustering model can be used to form a nearly optimal K-means-like clustering structure.

3.3. PCA-Guided Data Aggregating Model. As mentioned above, $D_{N\times P}$ represents the data measurement matrix collected from the sampling period by the cluster heads. Q is a centered matrix about D.

The vector $\{w_k\}_{1\leq k\leq N}$ is the principal components satisfying

$$QQ^T w_k = \lambda_k w_k. \tag{18}$$

Because in most cases there exist high correlations between sensor measurements, good approximations to sensor measurements can be obtained by relying on few principal components. The first principal component accounts for as much of the variability in the data as possible, so we can find the first principal component vector w_1, such that their projection data can effectively express the original measurements. Approximations $\widehat{Q}$ to Q are obtained by

$$\widehat{Q} = w_1 w_1^T Q = w_1 Z, \tag{19}$$

where

$$Z = w_1^T Q. \tag{20}$$

The CHs only send three packets about Z, w_k, and the mean vector column-wise G of D matrix to the base station:

$$G = \left(g_i = \frac{d_{i.}}{p}\right), \qquad d_{i.} = \sum_j d_{ij}. \tag{21}$$

Because the mean vector column-wise G is subtracted by the CHs prior to the aggregation of its value, the base station can add back after the computation of the approximation.

Example 2. If the CH collects the matrix D as follow:

$$D = \begin{bmatrix} 20.3 & 20.2 & 20.8 & 20.3 & 20.3 & 20.4 & 20.5 & 20.4\\ 20.4 & 20.3 & 20.6 & 20.4 & 20.4 & 20.5 & 20.6 & 20.5\\ 20.2 & 20.1 & 20.4 & 20.2 & 20.2 & 20.3 & 20.4 & 20.3\\ 20.3 & 20.2 & 20.8 & 20.3 & 20.3 & 20.4 & 20.5 & 20.4\\ 20.3 & 20.2 & 20.8 & 20.3 & 20.3 & 20.4 & 20.5 & 20.4\\ 20.4 & 20.3 & 20.6 & 20.4 & 20.4 & 20.5 & 20.6 & 20.5\\ 20.2 & 20.1 & 20.4 & 20.2 & 20.2 & 20.3 & 20.4 & 20.3\\ 20.3 & 20.2 & 20.8 & 20.3 & 20.3 & 20.4 & 20.5 & 20.4\\ 20.3 & 20.2 & 20.8 & 20.3 & 20.3 & 20.4 & 20.5 & 20.4\\ 20.2 & 20.1 & 20.4 & 20.2 & 20.2 & 20.3 & 20.4 & 20.3 \end{bmatrix}, \tag{22}$$

Then the CH can compute the matrix Q and the principal components w_1,

$$\begin{aligned} Z = w_1^T Q = \Big[\ & -0.26306,\ -0.56555,\ 0.93394,\\ & -0.26306\ -0.26306,\ 0.039433,\\ & 0.34193,\ 0.039433\ \Big],\\ w_1 = \Big[\ & 0.39468,\ 0.21031,\ 0.21031,\ 0.39468,\\ & 0.39468,\ 0.21031,\ 0.21031,\ 0.39468,\\ & 0.39468,\ 0.21031\ \Big],\\ G = \Big[\ & 20.4,\ 20.462,\ 20.262,\ 20.4,\ 20.4\\ & 20.462,\ 20.262,\ 20.4,\ 20.4,\ 20.262\ \Big], \end{aligned} \tag{23}$$

then the packet about Z, w_1, and G are delivered to the base station. The base station will compute the approximation $\widehat{Q}$.

After adding back the subtracted mean value G, we can obtain:

$$\hat{D} = \begin{bmatrix} 20.296 & 20.177 & 20.769 & 20.296 & 20.296 & 20.414 & 20.535 & 20.416 \\ 20.407 & 20.344 & 20.659 & 20.407 & 20.407 & 20.471 & 20.534 & 20.471 \\ 20.207 & 20.144 & 20.459 & 20.207 & 20.207 & 20.271 & 20.334 & 20.271 \\ 20.296 & 20.177 & 20.769 & 20.296 & 20.296 & 20.414 & 20.535 & 20.416 \\ 20.296 & 20.177 & 20.769 & 20.296 & 20.296 & 20.414 & 20.535 & 20.416 \\ 20.407 & 20.344 & 20.659 & 20.407 & 20.407 & 20.471 & 20.534 & 20.471 \\ 20.207 & 20.144 & 20.459 & 20.207 & 20.207 & 20.271 & 20.334 & 20.271 \\ 20.296 & 20.177 & 20.769 & 20.296 & 20.296 & 20.416 & 20.535 & 20.416 \\ 20.296 & 20.177 & 20.769 & 20.296 & 20.296 & 20.416 & 20.535 & 20.416 \\ 20.207 & 20.144 & 20.459 & 20.207 & 20.207 & 20.271 & 20.334 & 20.271 \end{bmatrix}. \tag{24}$$

4. PCA-Guided Routing Algorithm Solution Strategies

In Section 3, we have actually proposed a PCA-guided clustering and data aggregating model for routing optimization problem in WSNs by theoretical analyses and numerical examples. The following steps provide an overview of the solution strategy.

4.1. Initialization Stage. In the first stage, we assume that a set of N location coordinates are gathered at the base station. The base station computes the first principal component v_1. The two clusters C_1 and C_2 are determined via v_1 according to (15) by the BS.

4.2. Clusters Splitting Stage. When the number of sensor nodes is huge, two clusters are not enough and can induce the energy consume rapidly, considering splitting these clusters whose memberships are more than the CH can support.

If there are K clusters, there are on average N/K nodes per cluster (one CH node and non-CH nodes). We define that

$$\text{Ave} = N/K. \tag{25}$$

We can estimate the average energy dissipated per round to get the most energy efficient number of the clusters as follows:

$$\begin{aligned} E_{\text{round}} &= k(E_{\text{CH}} + E_{n\text{on-CH}}) \\ &= k\left[lE_{\text{elec}}\left(\frac{N}{k} - 1\right) + alE_{\text{elec}} + al\varepsilon_{\text{mp}}d^4_{\text{toBS}} \right. \\ &\quad \left. + \left(\frac{N}{k} - 1\right)\left(lE_{\text{elec}} + l\varepsilon_{\text{fs}}d^2_{\text{toCH}}\right)\right]. \end{aligned} \tag{26}$$

The notation and definition of the parameters in (26) are described as Table 1.

We can get

$$E_{\text{round}} = l\left[(2N + k)E_{\text{elec}} + 3k\varepsilon_{\text{mp}}d^4_{\text{toBS}} + \varepsilon_{\text{fs}}\frac{M^2}{2\pi}\frac{N}{k} - \varepsilon_{\text{fs}}\frac{M^2}{2\pi}\right]. \tag{27}$$

According to the average energy dissipated per round, the scope of the clusters' number can be estimated when it is the most energy efficient. In [4, 13, 14], the authors use the average energy dissipated per round to obtain the optimal cluster number. Based on this methodology, here in this paper, we study the appropriate upper limit of the cluster nodes to perform the clusters splitting and thus extend this methodology.

Example 3. If we assume the number of sensor nodes is 100, the average energy dissipated per round as the number of clusters is varied between 1 and 20. Figure 3 shows that it is most energy efficient when there are between 3 and 5 clusters in the 100-node network. We define that the most appropriate number of clusters is varied from 3 to 5 because

$$\begin{aligned} &\frac{(E_{\text{round}}(3) - E_{\text{round}}(\min))}{E_{\text{round}}(\min)} < 0.03, \\ &\frac{(E_{\text{round}}(5) - E_{\text{round}}(\min))}{E_{\text{round}}(\min)} < 0.03. \end{aligned} \tag{28}$$

We obtain that the appropriate upper limit of the cluster nodes is 100/3 = 34.

If the number of cluster nodes is more than 34, the PCA-guided clustering algorithm will implement to split it.

4.3. Cluster Balancing Stage. We use the BS running the PCA-guided clustering algorithm to divide sensor nodes based on the geographical information. We get K clusters from N nodes in the field rapidly and form better clusters by dispersing the CHs throughout the network.

Now let us introduce the basic idea of the cluster balancing stage. Above, we get the number of clusters K if there are N nodes. We define the average node number per cluster as Ave. The cluster-balanced step is added in each iteration process. If $|C_j| > \text{Ave}$, $1 \leq j \leq K$, the BS computes $\text{dist}(s_i, \overline{u_q})$, where $s_i \in C_j, j \neq q$ and $|C_q| < \text{Ave}$. This means to compute the distances between the nodes and each cluster center whose cluster's node number is less than Ave. The BS gets s_i if its value is minimum and adjusts the node into the corresponding cluster computed. After implementing the cluster-balanced step, the BS limit the node number in each

Table 1: Parameter description.

Parameters	Head description
k	The number of the clusters
N	The total number of the sensor nodes
E_{elec}	The energy consumption per bit when sending and receiving
l	The sending data bit
$\varepsilon_{\text{mp}}/\varepsilon_{\text{fs}}$	The energy consumption about the amplifier
a	The data aggregating rate. It is application specific, where we assume $a = 3$ as mentioned in Section 2.
d_{toBS}	The average distance between the CHs and the BS. We assume it = 75 m
d_{toCH}	The average distance between the CHs and the non-CH nodes. From [5], we obtain $d_{\text{toCH}} = \sqrt{(1/2\pi)(M^2/k)}$, in which, M denotes the area of this field.

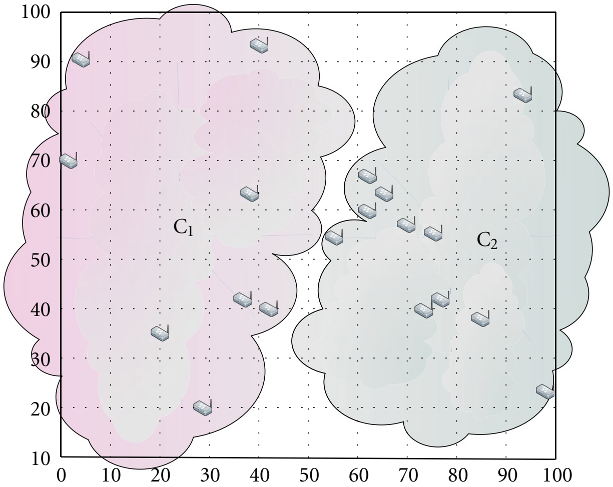


Figure 2: The clustering structure for 20 sensor nodes.

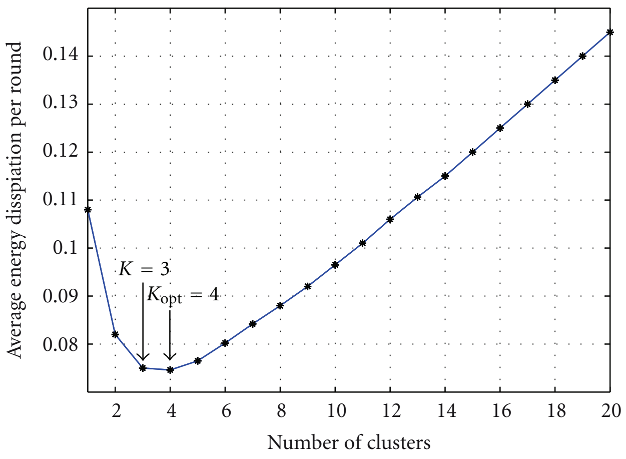


Figure 3: Average energy dissipation per round with varying the number of clusters.

cluster and change the clusters unequal distribution in the space of nodes originally.

Example 4. Figure 4 gives an example to illustrate how the cluster-balanced step works. In this example, if we only consider the geographical information of sensor nodes while using PCA-guided splitting algorithm, the sensor nodes s_1 to s_1 should belong to the C_1. However, the nodes in C_1 are more than Ave(=5). Thus, this scenario motivates the cluster-balanced step. To have a balance among all the clusters, in our method, we suggest that the sensor s_6 should be grouped into C_2 because its distance is nearest to C_2. We obtain the final balanced cluster structure by a serial standard PCA-guided splitting stage and the cluster balancing stage.

4.4. Cluster Heads Selecting Stage. After the base station divides the appropriate clusters using PCA technique, it needs to select the optimal cluster heads in these clusters. Assume that the initial energy is same, the base station can rank the matrix Y for each cluster and section the sensor nodes which is the nearest to the cluster centers.

4.5. Data Aggregating Stage. The sensor nodes begin to transfer the data to the cluster heads after finishing the cluster formation. The cluster heads collect the measurements from the sensor nodes and then compute the first principal component w_1 (as (18)), the mean vector G (as (21)), and Z (as (20)). They can be delivered to the base station with a constant packet size for each cluster head. At last, the base station will compute the approximate measurements by these packets.

4.6. The Description for PCA-Guided Routing Algorithm. The procedure taken by the base station is as follows.

Step 1: compute the first principal component v_1,

Step 2: according to (15), the two clusters C_1 and C_2 are determined via v_1.

Step 3: compute E_{round} and get the appropriate upper limit of the cluster nodes.

Step 4: while the cluster node number are more than the appropriate upper limit.

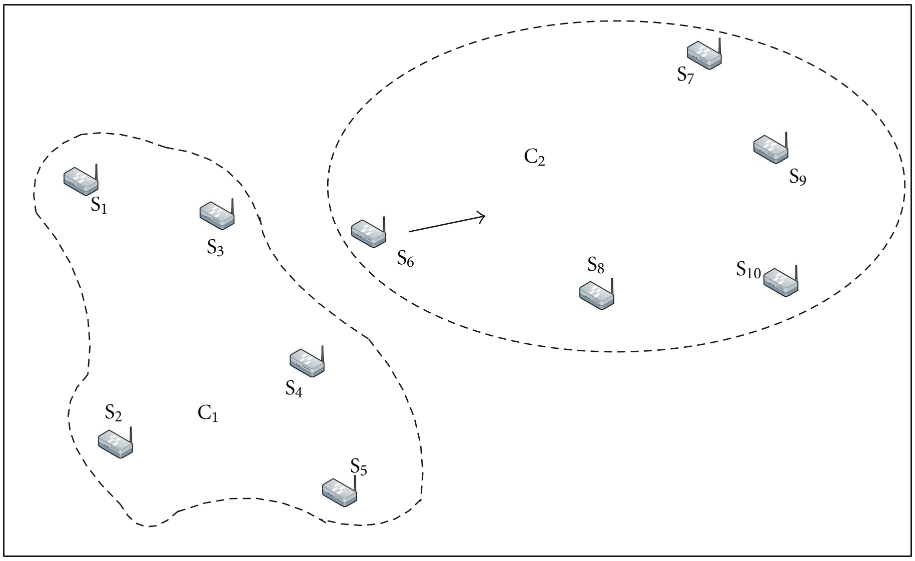


Figure 4: An example of the cluster balancing stage.

Step 5: compute a new v_1 for the cluster.

Step 6: repeat Step 2 and Step 3.

Step 7: end

Step 8: if needed, implement the cluster balancing stage.

Step 9: select the cluster heads.

Step 10: compute w_1 (according to (18)), G (according to (21)), and Z (according to (20)) for the approximate measurements.

5. Simulation Results

To evaluate the performance of PCA-RA, we simulate it, LEACH and LEACH-E, using a random 100-node network. The BS is located at (50, 150) in a $100 \times 100\,\mathrm{m}^2$ field.

Figures 5 and 6 show the clustering structure for using LEACH and PCA-RA. Comparing Figures 5 and 6, one finds that each cluster is as compact as possible, and the cluster heads locate more closely to the cluster centers by using PCA-RA. This gives us an intuition that it is more efficient to balance the load of network and to even distribute the nodes among clusters by using PCA-RA.

The benefits of using PCA-RA are further demonstrated in Figures 7 and 8, where we compare the network performance of network lifetime and throughput under the PCA-RA with that under LEACH and LEACH-E. Figure 7 shows the total number of nodes that remain alive over the simulation time, while the first dead node remains alive for a more long time in PCA-RA, this is because PCA-RA takes into account the structure of clusters and the location of the cluster heads. Figure 8 shows that PCA-RA sends much more data in the simulation time than LEACH and LEACH-E.

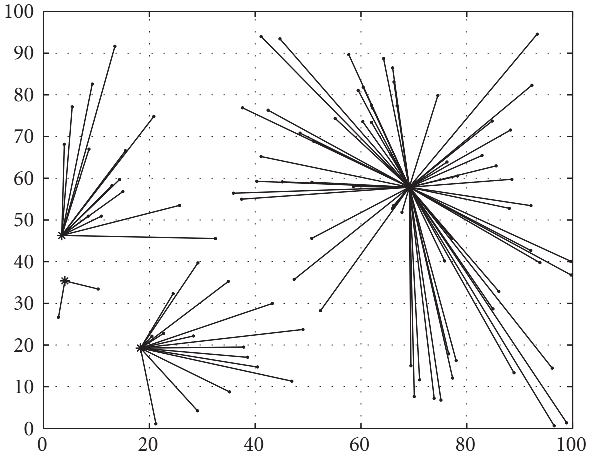


Figure 5: The clustering structure for using LEACH.

Note that the survey nodes in each round are NS_i; each node can send D_i data. Then, the maximum throughput can be expressed as follows;

$$\text{Throughput}_{\text{PCA-RA}} = \sum_{i=1}^{T_{\max}} NS_i \cdot D_i. \tag{29}$$

From Figure 7, the $T_{\max}$ for PCA-RA are more than the $T_{\max}$ for LEACH and LEACH-E. Then, $\text{Throughput}_{\text{PCA-RA}} > \text{Throughput}_{\text{LEACH}}$ and $\text{Throughput}_{\text{PCA-RA}} > \text{Throughput}_{\text{LEACH-E}}$.

For example, under the given test data, there are $60.456 * 10^3$ bits data sent in whole network lifetime with PCA-RA. And there are $33.394 * 10^3$ bits and $28.235 * 10^3$ bits by using LEACH-E and LEACH, respectively. The mathematics demonstrates that PCA-RA has 80.01% increase about

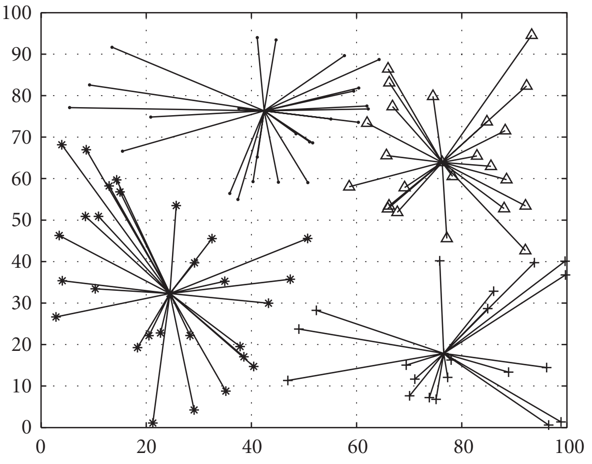


Figure 6: The clustering structure for using PCA-RA.

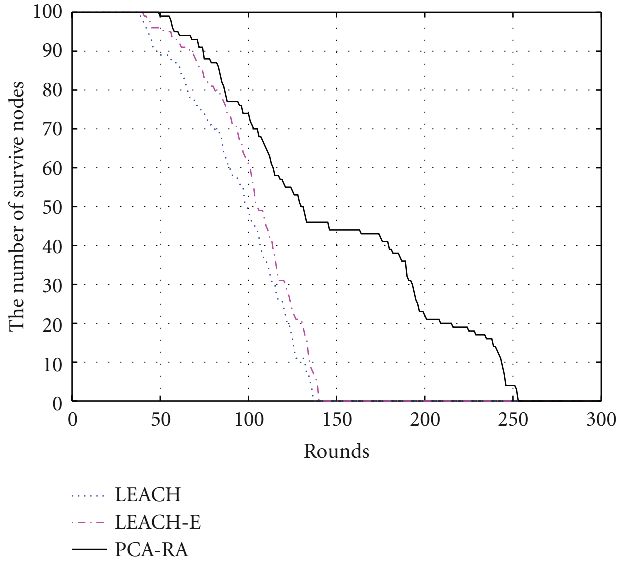


Figure 7: System lifetime using LEACH, LEACH-E and PCA-RA.

throughput compared with LEACH-E and 114.12% increase about throughput compared with LEACH.

PCA-RA is a centralized algorithm, and the complexity and communication cost of PCA-RA mostly happen in BS. About the balance structure stage, we think the effect on time complexity is small and consider that the time complexity is comparable to LEACH-C.

Table 2 displays the network lifetime (in terms of the time that the first node becoming dead) and the resulted square error function of the senor node structure under K-means algorithms and PCA-RA.

From Table 2, we can find that K-means algorithm can get a minimum square error. Because of the cluster-balanced step, PCA-RA can get a bigger square error, but the sensor nodes can survive a longer time. This implies that one can reach a certain tradeoff between the total spatial distance of sensor structure and the network lifetime. The suboptimal solution in PCA-RA can achieve such tradeoff.

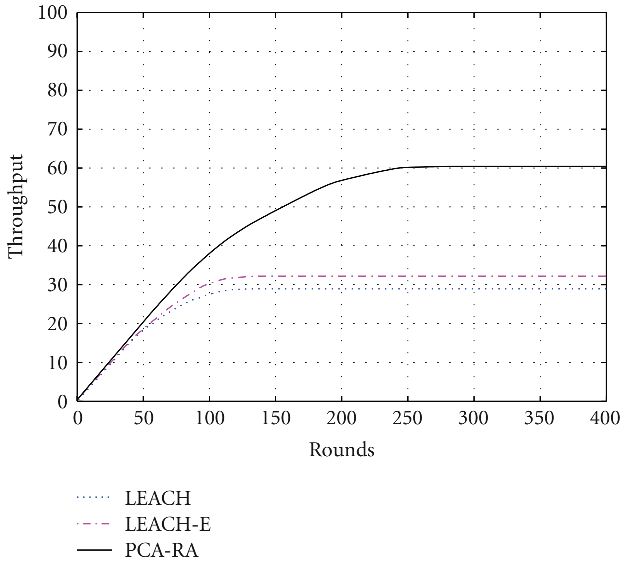


Figure 8: Throughput using LEACH, LEACH-E, and PCA-RA.

Table 2: A numerical example.

Parameters	K-means	PCA-RA
Square error	34392.980	38405.656
The first node dead time	88 rounds	99 rounds

In Figure 9, we simulate the sensor nodes collecting the measurements about temperature in some regions. Assume that the base station receive the packets from the cluster head in two minutes interval. Figure 9 demonstrates the approximations obtained by the base station about a certain sensor node in some intervals.

6. Conclusions

In this paper, we propose the PCA-guided routing algorithm for WSNs. By disclosing the connection between PCA and K-means, we design a clustering algorithm by utilizing PCA technique which efficiently develops a clustering structure in WSNs. Moreover, as a compression method, we demonstrate that the PCA technique can be used in data aggregation for WSNs as well. We establish the explicit procedure of PCA-guided routing algorithm for WSNs by incorporating PCA technique into both the data aggregating and routing process. The advantages of the proposed algorithm are demonstrated through both theoretical analyses and simulation results. The simulation results show that the PCA-guided routing algorithm significantly reduces the energy consumption, prolongs the lifetime of network, and improves network throughput when compared with LEACH and LEACH-E. Further, it keeps the accuracy about the measurements while reducing the network load.

Future research will focus on the distributed strategies of PCA-guided data aggregation and will investigate the performance of PCA-RA with different values of parameter K.

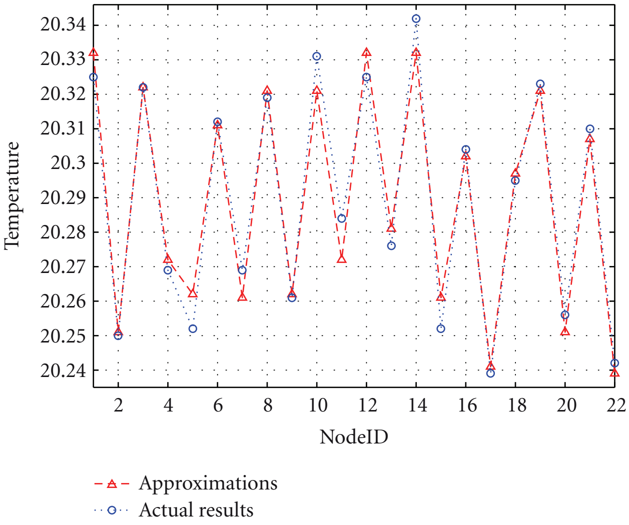


Figure 9: Approximation obtained for the sensor nodes.

Acknowledgments

The work described in this paper was supported by the National Natural Science Foundation of China under Grant no. 61070197 and by self-determined research funds of CCNU from the colleges' basic research and operation of MOE under Grant CCNU11A01017.

References

[1] I. F. Akyildiz, W. L. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[2] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: a survey," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6–28, 2004.

[3] Y. Tang, M. T. Zhou, and X. Zhang, "Overview of routing protocols in wireless sensor networks," *Journal of Software*, vol. 17, no. 3, pp. 410–421, 2006.

[4] W. Heinzelman, A. Chanrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Hawaii International Conference on System Sciences*, vol. 2, pp. 1–10, January 2000.

[5] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.

[6] S. Lindsey and C. Raghavendra, "PEGASIS: power-efficient gathering in sensor information systems," in *Proceedings of the IEEE Aerospace Conference*, vol. 3, pp. 1125–1130, 2002.

[7] A. Manjeshwar and D. P. Agarwal, "TEEN: a routing protocol for enhanced efficiency in wireless sensor networks," in *Proceedings of the 15th International Conference on Parallel and Distributed Processing Symposium*, pp. 2009–2015, April 2002.

[8] A. Manjeshwar and D. P. Agarwal, "APTEEN: a hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks," in *Proceedings of the 15th International Conference on Parallel and Distributed Processing Symposium*, pp. 195–202, 2002.

[9] O. Younis and S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004.

[10] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[11] M. Tubaishat and S. Madria, "Sensor networks: an overview," *IEEE Potentials*, vol. 22, no. 2, pp. 20–23, 2003.

[12] O. Zytoune, M. El Aroussi, and D. Aboutajdine, "A uniform balancing energy routing protocol for wireless sensor networks," *Wireless Personal Communications*, vol. 55, no. 2, pp. 147–161, 2010.

[13] L. S. Tan, Y. L. Gong, and G. Chen, "A balanced parallel clustering protocol for wireless sensor networks using K-means techniques," in *Proceedings of the IEEE 2nd International Conference on Sensor Technologies and Applications*, pp. 300–305, Cap Esterel, France, August 2008.

[14] Y. L. Gong, G. Chen, and L. S. Tan, "A balanced serial K-means based clustering protocol for wireless sensor networks," in *Proceedings of the 4th IEEE International Conference on Wireless Communications, Networking and Mobile Computing*, Dalian, China, October 2008.

[15] A. Abdulla, H. Nishiyama, and N. Kato, "Extending the lifetime of wireless sensor networks: a hybrid routing algorithm," *Computer Communications*, vol. 35, no. 9, pp. 1056–1063, 2012.

[16] A. F. Liu, J. Ren, X. Li, and Z. C. Xuemin, "Design principles and improvement of cost function based energy aware routing algorithms for wireless sensor networks," *Computer Networks*, vol. 56, no. 7, pp. 1951–1967, 2012.

[17] L. Cobo, A. Quintero, and S. Pierre, "Ant-based routing for wireless multimedia sensor networks using multiple QoS metrics," *Computer Networks*, vol. 54, no. 17, pp. 2991–3010, 2010.

[18] S. He, J. Chen, Y. Sun, D. K. Y. Yau, and N. K. Yip, "On optimal information capture by energy-constrained mobile sensors," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 5, pp. 2472–2484, 2010.

[19] X. Cao, J. Chen, C. Gao, and Y. Sun, "An optimal control method for applications using wireless sensor/actuator networks," *Computers and Electrical Engineering*, vol. 35, no. 5, pp. 748–756, 2009.

[20] J. Shlens, "A tutorial on principal component analysis," December 2005, http://www.snl.salk.edu/~shlens/pca.pdf.

[21] D. P. Berrar, W. Dubitzky, and M. Granzow, in *A Practical Approach to Microarray Data Analysis*, pp. 91–109, Norwell, Mass, USA, 2003.

[22] G. Golub and C. Van Loan, *Matrix Computation*, Baltimore, Md, USA, 3rd edition, 1996.

[23] H. Zha, C. Ding, M. Gu, X. F. He, and H. Simon, "Spectral relaxation for K-means clustering," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS '01)*, vol. 14, pp. 1057–1064, 2002.

[24] C. Ding and X. F. He, "K-means clustering via principal component analysis," in *Proceedings of the International Conference on Machine Learning (ICML'04)*, p. 29, Alberta, Canada, July 2004.

[25] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[26] J. Ham and M. Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufman Publishers, 2nd edition, 2006.

Hindawi Publishing Corporation
Journal of Computer Networks and Communications
Volume 2012, Article ID 438654, 14 pages
doi:10.1155/2012/438654

Research Article

Performance Analysis of the IEEE 802.11s PSM

Mirza Nazrul Alam,[1] Riku Jäntti,[1] Jarkko Kneckt,[2] and Johanna Nieminen[3]

[1] *Department of Communications and Networking, Aalto University School of Electrical Engineering, Otakaari 5A, 02150 Espoo, Finland*
[2] *Nokia Corporation, Nokia Research Center, Otaniementie 19b, 02150 Espoo, Finland*
[3] *TeliaSonera, Sturenkatu 16, 00510 Helsinki, Finland*

Correspondence should be addressed to Mirza Nazrul Alam, mirza.alam@aalto.fi

Received 15 June 2012; Accepted 18 October 2012

Academic Editor: Krishna Sayana

With the introduction of IEEE 802.11 power save mode (PSM), a lot of work has been done to enhance the energy saving ability of the wireless nodes. The ultimate goal of the research is to make the networking equipment carbon neutral and prolong the lifetime of the energy limited device for various applications; in some cases it is a trade-off between energy efficiency and delay. However, few studies have been made until now in the area of IEEE 802.11s based link specific power mode. The essence of this method is the ability of a node to maintain different power modes with its different peer nodes at the same time. A new peer service period (PSP) mechanism is also proposed in IEEE 802.11s amendment for transmitting to a receiver operating in PSM. In this paper the performance of the link specific power mode is studied for a single- and a multilink network in terms of energy, delay throughput, and sleep duration. It is found that at small load the energy saving could be as high as eighty percent when compared with the active mode operation. A stochastic model, based on discrete time discrete state Markov chain, is developed for one peer link operation to study the system behavior closely during PSM operation.

1. Introduction

From the last decade, the deployment of mobile wireless devices has been rising by leaps and bounds in varieties of applications, such as in data access purpose, in wireless automation and control, making Voice over Internet Protocol (VoIP) calls, and in other QoS sensitive applications. However, these mobile devices are battery powered and have limited amount of operational time. As the batteries are limited resource of energy, devising an efficient energy saving protocol is now a critical issue for battery-constrained wireless devices. Generally, a significant amount of energy of a wireless node is dissipated in wireless radios. During normal operation a radio typically undergoes the following three states: transmitting, receiving, and idle state. The transmitting state consumes the highest amount of energy since the major portion of the energy is dissipated in power amplifier. In the idle state the radio obtains transmission opportunities (TXOPs) and is capable to start receiving the transmissions. The idle state consumes a little bit less energy than receiving state but in the long run this state contributes a significant portion of the total energy consumption. Therefore in PSM, the node avoids unnecessary operation in idle state and when a node neither receives nor transmits it switches to the sleep state. The sleep state consumes the lowest amount of energy. As an example, the current drawn by the Lucent IEEE 802.11 WaveLan card, for transmit, receive, idle, and sleep modes is 284, 190, 156, and 10 mA, respectively [1]. In basic PSM operation the destined packets are temporarily buffered by the transmitter station (STA) and are later delivered to the destination. The packets are transmitted in burst at some agreed-upon intervals. When the queue of the transmitter STA becomes empty, it switches to doze sate.

With the introduction of IEEE 802.11 PSM operation [2], a lot of work has been done to reduce the energy consumption of the nodes while maintaining the QoS for various applications [3–5]. However, few studies have been made until now in the area of IEEE 802.11s based link specific power mode. In this scheme, a mesh STA creates links with peer mesh STAs and maintains a link specific power mode towards each peer. A mesh STA also tracks the power mode of each peer mesh STA and exchanges data frames with only peer mesh STAs. Three link specific power modes are defined and they are active, light sleep, and deep sleep. The power modes of peerings are independent and a mesh STA may operate in different power modes for each peering [6]. A peer service period (PSP), an agreed contiguous time period, is used to exchange buffered frames in a link if the receiver STA operates in PSM; that is, it operates in light sleep or in deep sleep mode for the link. The IEEE 802.11s has proposed its own guideline to initiate, maintain, and terminate a PSP.

In this paper the performance of the IEEE 802.11s PSM for one peer link operation in a single 20 MHz channel is first studied both numerically and analytically. Several mathematical models are developed to validate the simulation output such as the batch size, packet delay, sleep duration, and energy for one peer link. Later the simulation study is extended for a multihop network consisting of 8 STAs. It is found that at low traffic rate the percentage of average energy saving could be as high as 79% when compared to active mode operation. The results indicate that the total energy consumption depends on the independent power modes of the STAs for their link as well as the application traffic rates.

The rest of the paper is organized as follows. Previous works done on the IEEE 802.11 PSM are presented in Section 2. In Section 3 the PSM link concept in the IEEE 802.11s and the frame exchange methods in PSM are discussed. The setup for the experiment is addressed in Section 4. The overall operation of the implementation, especially the batch scheduling and sleep/wake-up method, are explored in Section 5. The mathematical model for one peer link operation is introduced in Section 6. The simulation results for one peer link are validated in Section 7. The simulation result of a multihop network is examined in Section 8. A summary is drawn in Section 9.

2. Previous Works

Krashinsky and Balakrishnan [7] proposed an algorithm called bounded slow down (BSD) to save energy for web traffic. It bounds the delay to a user specified value while expending minimum energy. The sleep duration is adapted depending on the previous activity. Another solution for the same kind of problem is done by Yan et al. [8] where energy is saved in two steps: by switching to doze state during inactive periods and by controlling the traffic flow by declaring enough buffer space in the TCP ACK packet when necessary. It creates burst of data flow from the AP towards the STA and shortens the total transmission time. Zhu and Cao [9] proposed a rate based scheduling algorithm for streaming traffic where the AP will decide the service order of the flows. Gan and Lin [10] proposed a power conservation scheme to optimally schedule the awake time of the STAs to reduce the contention. Lei and Nilsson [11] modeled the queue for the infrastructure mode to predict the system performance if the configuration parameters are known. Baek and Bong [12] later extended this work and derived the exact average value of the Percentage of Time a station stays in the Doze state (PTD) and variance of delay. It is assumed in both [11, 12] that the polling message from the STA has negligible impact and the AP always transmits a fixed amount of packets within one beacon interval. Chao et al. [1] and Memarzadeh et al. [13] proposed a quorum based energy conserving protocol for a single hop MANET. A quorum set is a set of time frames wherein a node must wake up. Any two nodes can wake up and meet each other at some time frame defined in the set. During the nonquorum time frames, nodes are allowed to remain in sleep state to save their energy. Lee and Kim [14] used the Kalman filter to predict the best Announcement Traffic Indication Message (ATIM) window size to increase the time spent in doze state.

Almost all the sleep/wake-up based power saving schemes that have been proposed yet mainly address the following aspects of the IEEE 802.11 PSM. These are the effective notification in traffic indication map (TIM) field or the efficient scheduling of data frames in the queue [9, 10], traffic shaping to create burst when necessary [7, 8], ATIM window size adjustment [14], asynchronous wake-up of STAs [1, 13, 15], Unscheduled Power Save Delivery (UPSD) [3, 4], and so forth. However, few papers are found to focus on the IEEE 802.11s based link specific power mode operation. In this paper the performance of the link specific power modes is studied from the energy efficiency point of view.

3. IEEE 802.11s System

In this section the IEEE 802.11s PSM link concept as well as the frame transmission and reception method in a link in power save mode is introduced.

3.1. The PSM Link Concept in IEEE 802.11s. The most significant point of the 802.11s PSM operation is the STA's link specific power mode. A mesh STA creates a link and maintains a link specific power mode towards each peer STA. It also tracks the power mode of each peer and only exchanges data frames with its peer. A link consists of two mesh STAs and both STAs have their own independent power mode for each other. A STA can operate in any of the three power modes for a link. These modes are active, light sleep, and deep sleep. A single mesh STA can serve its various peers in different power mode at the same time. In Figure 1, STA A is in active mode for both link-X and link-Y. STA C is in deep sleep mode for link-Y and link-Z. The same STA B is in light sleep mode for link-X and in deep sleep mode for link-Z. In light sleep mode for a link, a PSM STA wakes up periodically to listen to all the beacons of the peer STA. On the other hand, in deep sleep mode for a link a PSM STA may not wake up to listen to all the beacons. In active mode operation, a STA remains in awake state all the time. If a STA works in either light sleep or deep sleep mode for any of its peer links, the STA will alternate between awake and doze states for that link, as determined by the frame transmission and reception rules [6].

3.2. Frame Transmission and Reception Method. In previous section the concept of link specific power modes and how

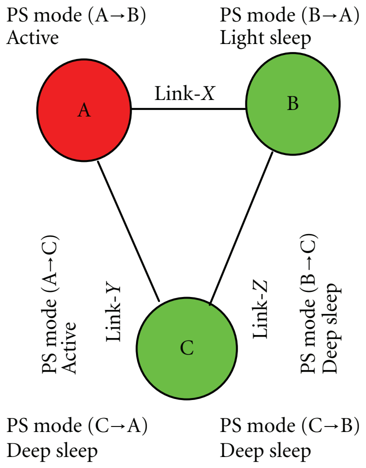


Figure 1: Link specific power modes of three mesh STAs. STA A is in active mode. STAs B and C are in PSM.

these modes affect the internal power states of a mesh STA were introduced. In this section the transmission and reception method of frames in PSM will be examined. Unlike 802.11, beacons of the mesh STAs are not synchronized in 802.11s, rather evenly distributed. Every STA in a network has its own Targeted Beacon Transmission Time (TBTT) as illustrated in Figure 2. The Mesh Beacon Collision Avoidance (MBCA) mechanism detects and mitigates the collisions among beacon frames in the network [6]. The Mesh Coordinated Channel Access (MCCA) enables mesh STAs to reserve transmission time to increase transmission efficiency. The transmission time reservations avoid hidden terminals and in forwarding operation, the time may be reserved for both transmission and reception of the forwarded data. All this coordination improves network efficiency. However, the details of this mechanism are outside the scope of this paper. In IEEE 802.11s, multiple STAs can transmit in different time within one beacon interval duration. It obviously increases the network throughput. Each mesh STA maintains a Time Synchronization Function (TSF) timer with modulus 2^{64} [6]. The beacon frames contain a copy of its TSF timer. In PSM, a mesh STA first enters the awake state prior to its every TBTT and remains in awake state for an awake-window duration as shown in Figure 2. After hearing the beacon, the corresponding peer STA triggers the beaconing STA to initiate a PSP. A PSP is a contiguous period of time during which the buffered frames are transmitted to the PSM mesh STAs. A mesh STA announces the presence of buffered data in the beacon's TIM element in its own TBTT. A trigger might be a mesh data or mesh null packet which should be acknowledged. After receiving the trigger, the beaconing STA starts to transmit the buffered packet one by one. The PSP will be terminated after a successfully acknowledged data frame or a mesh null frame with the End of Service Period (EOSP) bit set to 1 from the transmitter of the PSP [6].

The awake state could be extended if any PSP is initiated within the awake-window duration. In this case, a STA will remain in awake state until the PSP is terminated successfully. If no PSP is initiated, the mesh STA should return to doze state just after the awake window. A PSM STA can transmit frames to its active mode peer at any time without establishing any PSP. However, frame transmission towards a PSM peer will never take place arbitrarily. A PSP should be established before any transmission towards the PSM peer takes place. Figure 2 shows the activity of the mesh STAs A, B, and C during frame exchange. As STA B is in light sleep mode for STA A, it wakes up prior to every TBTT of STA A to listen to A's beacons. It may not hear all the beacons of STA C since it acts in deep sleep mode for peer STA C. STA A will remain in awake state all the time as it serves its links in active mode. In Figure 2, mesh STAs B and C transmit their buffered frame to active mesh STA A directly without initiating any PSP. On the other hand, transmission of buffered packet towards PSM STAs B or C takes place only after the initiation of PSP as described before.

4. Implementation and Experimental Setup

In order to evaluate the performance, the IEEE 802.11s PSM related operations such as beacon management, PSP protocol, PSM efficient queue, and peer specific buffers on top of the queue are implemented in the extended version [16] of Network Simulator NS-2. The details of the implementations are out of the scope of this paper. In the simulations, the UDP packet size is 1 KB and the beacon interval is 102.4 ms or 100 time unit (TU). The traffic is Poisson distributed and the access mechanism is DCF. The propagation model is taken as free space to observe the MAC performance. Thus, packets are lost only for collision. During the PSM operation, a STA wakes up for a fraction of a millisecond before its own TBTT for safety purpose. In this paper this duration before TBTT is called safety margin. If there is no transmission, the transmitter goes to doze state just after the awake window. In the experiment, safety margin is 0.1024 ms and awake-window duration is 5 ms. Therefore, the maximum duration of sleep possible within a beacon interval is around 97.29 ms (102.4 ms − 5 ms − 0.1024 ms = 97.29 ms). The necessary parameters used to measure the performance are given in Table 1.

The experiment is carried out for one peer link and for a multihop network separately. In one peer link scenario there are two STAs; one STA operates as transmitter and another as receiver. For a multihop scenario the network consists of eight STAs. Two linear data flows consisting of 4 STAs are placed across each other as shown in Figure 15. The mathematical modeling and the validation of the simulation results for one peer link operation are explored in Sections 6 and 7, respectively. In Section 8 the simulation results for a multihop network are discussed.

5. The Batch Scheduling Method

In this section, the overall system response of the implementation, that is, the batch scheduling process and the corresponding sleep/wake-up event, are explained. In the new queuing system, there are separate temporary buffer spaces for all PSM peer STAs as shown in Figure 3. The

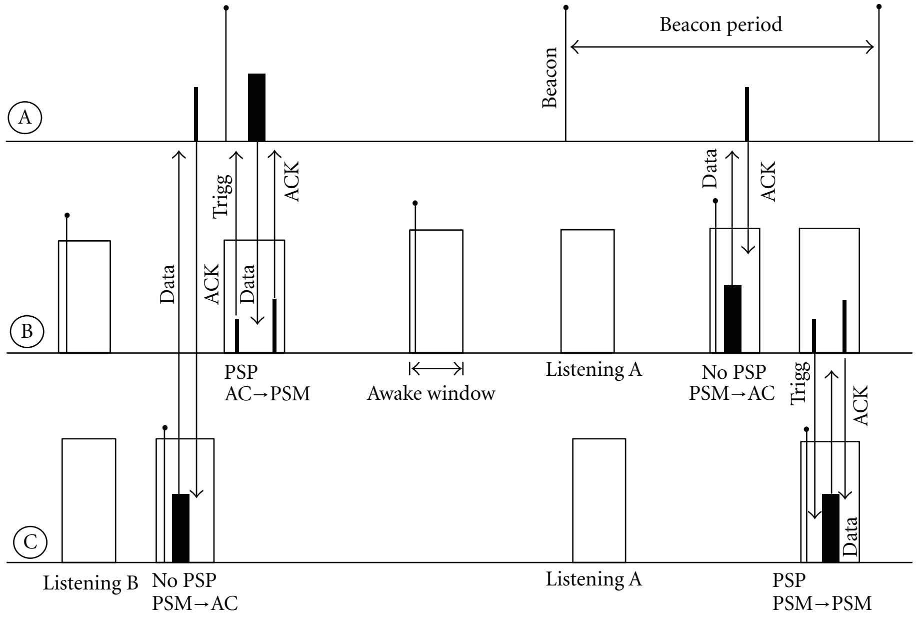


Figure 2: The internal states of mesh STAs during frame transmission and reception. A is in active mode. B and C are in PSM. STA's power modes to links are given in Figure 1.

Table 1: Simulation parameters.

Parameters	Values
TX state power	1.327 W
RX state power	0.967 W
Idle state (neither TX nor RX state) power	0.844 W
Doze state power	0.066 W
Energy per switching	0.422 mJ
Switching time, from doze to idle state	250 μs
Beacon transmission interval	102.4 ms
Data transmission Rate	6 Mbps
Packet size	1000 Byte
Beacon frame size	272 Byte
PSP trigger frame size	28 Byte
Awake-window duration	5 ms
Safety margin or wake-up before TBTT	0.1024 ms
DIFS	34 μs
SIFS	16 μs
CW slot duration	9 μs
CW_{min}	15

queue-logic unit first stores the incoming packet from LL to the exact peer specific buffer in MAC. When MAC wakes up to transmit its own beacon, it first checks all of its peer specific buffers to see whether any stored packet exists. If so, the STA declares those peers' identities in its beacon's TIM element. However, there is no intermediate buffer for the data destined to the active mode peers. This data is directly transferred to the final transmission queue as shown in Figure 3.

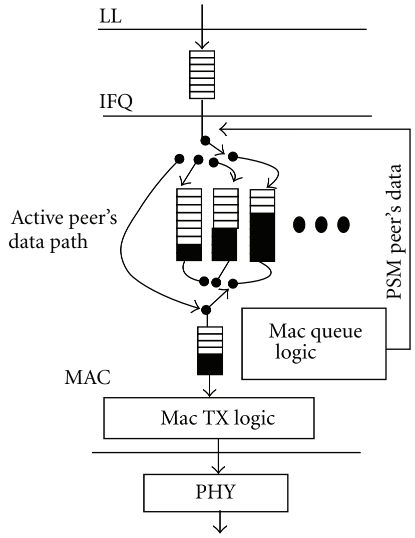


Figure 3: A simplified diagram of new queuing system where MAC maintains individual queue for each of its PSM link or peers.

Prior to every PSP, the buffered packets are moved to the MAC transmission queue as a group or batch before delivering them one by one. After this batch a null packet with EOSP field set to 1 is placed as an indication of the end of an individual batch in this implementation. The transmitter STA remains in active state as long as it has frames in the transmission queue to deliver or it successfully receives the ACK to the last frame. The receiver STA also remains in active state until it receives the end trigger frame. The total batch service time may take several beacon intervals. New data packets destined to the STA, to which a PSP is ongoing, will not be set immediately to the MAC transmission queue; rather these will be stored in the peer specific buffer and will be delivered in another PSP. However, in the implementation, if the arrived frame is destined to some other mesh STA with whom no PSP currently exists, the transmitter can establish a new PSP. A STA can continue several PSP operations in parallel with its peer STAs, but not more than one PSP with a particular peer STA according to the current implementation.

An example of this operation is given in Figure 4 for a single link. Here STA1 is generating traffic and has only one peer link with STA2. STA2 is working in light sleep mode for the link and receiving frames from STA1. The busy period, service period, and sleep duration of STA1 during the operation are depicted in Figure 4. Batches B21, B22, and B23 are transferred to the queue of STA1 from its temporary buffer after successful initiation of PSP at 1st, 3rd, and 4th beacon, respectively. Here batch Bxy means that this batch is destined to STAx and the batch number for this station x is y. Suppose B21, B22, and B23 batches have service times $x1$, $x2$, and $x3$, respectively. The service rate in Figures 4 and 5 is considered 1 sec/sec and therefore the unfinished work $U(t)$ decreases with slope equal to -1. It is shown in Figure 4 that no batches arrive in the transmission queue just after the 2nd, 5th, and 6th beacons. One good reason for this is that the STA1 did not send any indication of buffered packet in these beacons as the PSP with STA2 is still continuing. Let us consider another scenario where STA1 is connected with another station STA3 and is also generating traffic for it, therefore it has two peers, STA2 and STA3. The activities of STA1 queue in this scenario are depicted in Figure 5. At the first beacon a PSP is established with STA2 and the batch B21 is transferred to the transmission queue in STA1. Similarly, after beacon 2, another batch B31 is transferred to the queue from its temporary buffer. The batch B31 will be placed behind the batch B21. The batches are transferred to the queue after being triggered by the corresponding receiver STA. It should be noted that although a PSP is progressing with STA2, a new PSP with STA3 is established after beacon 2 and at this moment STA1 has two parallel PSPs, one with STA2 and another with STA3. As shown in Figure 5 the first PSP ends at time t' after delivering all the packets to STA2. A little bit different situation occurs after announcing the existence of buffered frame for both stations in beacon 4. Here STA3 wins to send the PSP trigger first and the batch B32 is transferred to the queue for immediate transmission. STA1 also receives the trigger from STA2 after few milliseconds due to contention and subsequently the batch B23 is transferred to the transmission queue and placed behind the previous batch B32. Like before the system will start to serve the batch B23 at time t'' after emptying the previous batch B32 as shown in Figure 5. In the current implementation a STA can maintain N links and therefore in case of N parallel PSPs the batch scheduling process in the MAC transmission queue will be the same as explained above. The primary concern of this paper is to evaluate the performance of the proposed 802.11s PSM from the energy efficiency point of view. Flow control and scheduling methods could be utilized to maintain fairness among the flows. However, the fairness considerations are outside the scope of this paper.

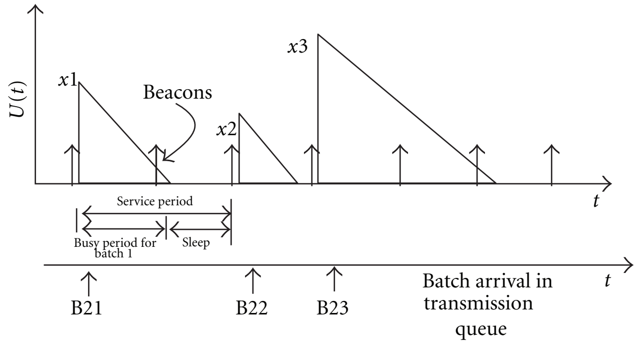


Figure 4: Batch arrival process in STA1 queue for a single peer. B21 denotes the first batch for STA2. Similarly B23 is the third batch for STA2.

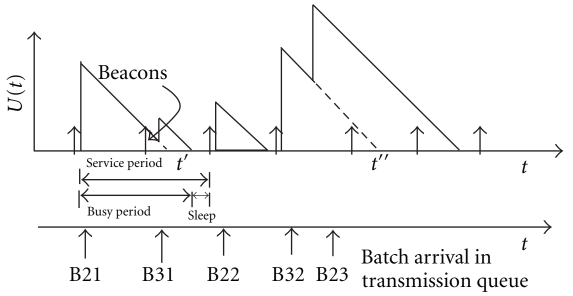


Figure 5: Batch arrival process in STA1 queue for a two peers. B21 denotes the first batch for STA2. Similarly B32 is the second batch for STA3.

6. Modeling One Peer Link Operation

In this section, the dynamic that affects the energy-related issues in a network is observed closely. In order to do this, a mathematical model based on discrete time discrete state Markov chain is developed for one peer link operation. The model is later used in Section 7 to validate the simulation output. There are two STAs in this modeling; one STA operates as a transmitter and another as a receiver.

In this modeling, several assumptions are made. It is assumed that the transmitter STA operates in deep sleep mode for the link. The transmitter only wakes up before its own TBTT to transmit and does not wake up to hear receiver's beacon. However, the receiver STA wakes up periodically to hear the transmitter's beacon. As there is no need to declare the presence of buffered traffic in its own TBTT for this typical scenario, the receiver STA does not wake up to transmit its own beacon. After successful transmission of a batch the transmitter goes to sleep if the remaining time to the next wake-up is greater than zero. The transmission time of beacon and PSP trigger is ignored in this modeling. As propagation model is free space, there is no packet loss in the channel for a single transmitter. The buffers are large enough so that the probability of packet loss due to buffer overflow is negligible in this region of operation.

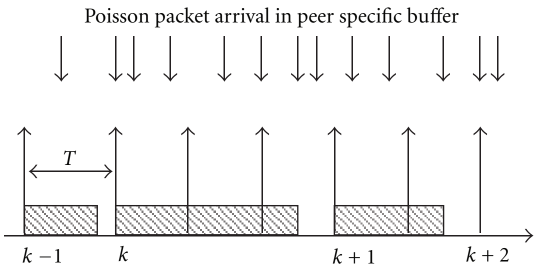


Figure 6: The system observation instants are given by k. The batch transmission time is shown by hatched region.

6.1. Modeling Batch Size Distribution in Steady State. Assume that packet length is fixed. Let X denote the deterministic part of the packet transmission time $X = \text{DIFS} + \text{Data} + \text{SIFS} + \text{ACK}$. Due to the contention based access the total transmission time of a packet p is random variable $\widetilde{X}_p = X + \widetilde{C}_p$ where $\widetilde{C}_p \sim U(0, \text{CW}_{\min})$ is uniform random variable corresponding to the contention window length. The total transmission time of a batch of a packets is $\widetilde{B}(a) = aX + \widetilde{\Delta}(a)$ where $\widetilde{\Delta}(a) = \sum_{p=0}^{a-1} \widetilde{C}_p$ denotes the additional transmission time due to contention process. Let $F_{\Delta,a}(t)$ denote the cumulative probability density (CDF) function of $\widetilde{\Delta}(a)$. Closed form solution of $F_{\Delta,a}(t)$ is known [17], but for large a it has numerical problems. In this paper $F_{\Delta,a}(t)$ is approximated by *truncated Gaussian* CDF. The details of this approximation can be found from the appendix. The approximation is based on two observations. Firstly, the central limit theorem states that sum of i.i.d. random variables approaches to Gaussian distribution when number of variables a grows. Secondly, it is known that $0 \leq \widetilde{\Delta}(a) \leq a \times \text{CW}_{\min}$.

Let T be beacon interval. New batch can only start in the beginning of a beacon interval. Hence, there is idle time $\widetilde{I}(a) = \widetilde{N}(a)T - \widetilde{B}(a)$ between the end of the batch and beginning of the next batch. The random variable $\widetilde{N}(a) = \text{ceil}\{\widetilde{B}(a)/T\}$ denotes the number of beacon intervals needed to serve the batch. Let us define $I(a;n) = nT - aX$. The probability that the transmission of a packet spans over n beacon intervals can now be expressed as

$$\begin{aligned}\Pr\left\{\widetilde{N}(a) = n\right\} &= \Pr\left\{(n-1)T < \widetilde{B}(a) \leq nT\right\} \\ &= F_{\Delta,a}(I(a;n)) - F_{\Delta,a}(I(a;n-1)).\end{aligned} \tag{1}$$

Now the batch size distribution could be characterized. The system is observed at each TBTT after a successful transmission of a batch. The observation instants are shown by k in Figure 6. The accumulated packets or a new batch are transferred to the MAC transmission queue from the temporary buffer before transmission at each observation instant. Assume that packets arrive according to Poisson process with rate λ. Let $\widetilde{a}_k$ be the batch size at kth instant. The batch size $\widetilde{a}_k$ depends only on the size of the previous batch $\widetilde{a}_{k-1}$. Hence the batch size process fulfils Markov property. Assume that the arrival rate is small enough such that the batch size is not growing without bound. Then the probability that the size of batch k is j conditioned that the size of batch $k-1$ was i can be written as

$$p_{ij} = \sum_{n=n_{\min}(i)}^{n_{\max}(i)} \frac{1}{j!}[\lambda T n]^j \exp[-\lambda T n] \Pr\left\{\widetilde{N}(i) = n\right\}, \tag{2}$$

where $n_{\min}(a) = \text{ceil}\{aX/T\}$ and $n_{\max}(a) = \text{ceil}\{a(X + \text{CW}_{\min})/T\}$ denote the minimum and maximum number of data frames needed to serve batch of size a, respectively. The steady state probability distribution of the batch π_j size can be solved from the following set of linear equations:

$$\pi_j = \sum_{j=0}^{\infty} \pi_i p_{ij}, \quad j = 0, 1, 2, \ldots \tag{3}$$

$$1 = \sum_{j=0}^{\infty} \pi_i. \tag{4}$$

In practice, the buffer size is constrained. Let $a_{\max}$ denote the maximum MAC buffer size in terms of packets. Now the state space of the Markov chain can be truncated and standard linear algebraic methods can be used to solve π_j.

6.2. Modeling Sleep Segment Distribution. Let's assume that the size of the random part due to CW, $\widetilde{\Delta}(a)$, is always less than a beacon period. The ground of this assumption is that even for a very big size batch, suppose 300 packets, the maximum backoff time that may come is 300 × 15 × 0.009 = 40.5 ms, less than the beacon interval of 102.4 ms. Here $\text{CW}_{\min} = 15$ and slot size = 0.009 ms as given in Table 1. Therefore, the transmission time of a batch of size a will occupy either $n_{\min}(a)$ or $n_{\max}(a) = n_{\min}(a) + 1$ numbers of beacon periods as shown in Figure 7. Now the size of a sleep segment depends on at which point an ongoing batch transmission ends inside a beacon interval or the time left for the next wake-up. Therefore any one of the following three cases may arise: (i) a batch transmission may end within the safety margin before beacon $n_{\min}(a)$ or $n_{\max}(a)$; (ii) it may end within an awake window after beacon $n_{\min}(a)$ or $n_{\max}(a)$; (iii) it may end neither within awake window nor within safety margin as shown in Figure 7.

The transmitter can be put to sleep for the duration of the idle time $\widetilde{I}(a)$. Let $\widetilde{s}(a)$ be the sleep duration after the successful transmission of batch of size a. The above three cases are analyzed below for $a > 0$.

Case 1. Let the transmission end within the safety margin window x before beacon $n_{\min}(a)$ or beacon $n_{\max}(a)$, that is, in region C or F as shown in Figure 7. The corresponding sleep duration here will be zero because the time left for

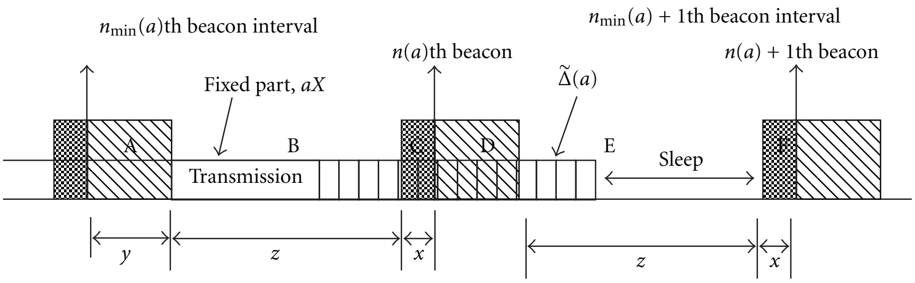


Figure 7: The safety margin and awake-window area are shown by deep hatched and light hatched, respectively. Here $T = 102.4$ ms, $y = 5$ ms, $x = 0.1024$ ms, and $z = 97.29$ ms. Based on awake window and safety margin beacon interval $n(a)$ and $n(a) + 1$ are divided into 6 regions A to F. The deterministic part and indeterministic part of a transmission is also shown here separately.

the next wake-up is zero. Therefore the probability that the sleep duration is zero,

$$\begin{aligned}\Pr\{\tilde{s}(a) = 0 \mid \tilde{a} = a\} &= \Pr\left\{\tilde{I}(a) \le x\right\} \\ &= F_{\Delta,a}(n_{\min}(a)T - aX) \\ &\quad - F_{\Delta,a}(n_{\min}(a)T - aX - x) \\ &\quad + F_{\Delta,a}(n_{\max}(a)T - aX) \\ &\quad - F_{\Delta,a}(n_{\max}(a)T - aX - x).\end{aligned} \tag{5}$$

Case 2. Let the transmission end within the awake window after beacon $n_{\min}(a)$ or beacon $n_{\max}(a)$, that is, in region A or D as shown in Figure 7. These beacons will not contain any indication of buffered packet as PSP is still continuing. Hence STA will go to sleep just after the awake window and the sleep duration will be equal to z or maximum value of 97.29 ms. Therefore the probability that the sleep duration is z,

$$\begin{aligned}\Pr\{\tilde{s}(a) = z \mid \tilde{a} = a\} &= \Pr\left\{\tilde{I}(a) \ge z + x\right\} \\ &= F_{\Delta,a}(n_{\min}(a)T - aX - (z + x)) \\ &\quad - F_{\Delta,a}(n_{\max}(a)T - aX - (z + x)) \\ &\quad + F_{\Delta,a}(n_{\min}(a)T - aX).\end{aligned} \tag{6}$$

Case 3. Let the transmission end in region B or E in Figure 7. In this case the sleep duration will vary between 0 to z. The probability that the sleep duration is s, where $0 < s < z$,

$$\begin{aligned}\Pr\{\tilde{s}(a) \le s \mid \tilde{a} = a\} &= \Pr\left\{x < \tilde{I}(a) < z + x\right\} \\ &= F_{\Delta,a}(n_{\min}(a)T - aX - x) \\ &\quad - F_{\Delta,a}(n_{\min}(a)T - aX - (x + s)) \\ &\quad - F_{\Delta,a}(n_{\max}(a)T - aX - x) \\ &\quad + F_{\Delta,a}(n_{\max}(a)T - aX - (x + s)).\end{aligned} \tag{7}$$

From (5), (6), and (7) and from the total probability theorem for all size of batches,

$$\begin{aligned}&\Pr\{\tilde{s} \le s\} \\ &= \begin{cases}\displaystyle\sum_{i=1}^{a_{\max}} \big(\Pr\{\tilde{s}(i) \le s \mid \tilde{a} = i\} \\ \qquad + \Pr\{\tilde{s}(i) = 0 \mid \tilde{a} = i\}\big)\Pr\{\tilde{a} = i\}, & s < z, \\ \displaystyle\sum_{i=0}^{a_{\max}} \big(\Pr\{\tilde{s}(i) = z \mid \tilde{a} = i\}\big)\Pr\{\tilde{a} = i\}, & s = z.\end{cases}\end{aligned} \tag{8}$$

In above, $\Pr\{\tilde{a} = i\} = \pi_i$ is derived in Section 6.1. $\Pr\{\tilde{s}(0) = z \mid \tilde{a} = 0\} = 1$; that is, if there is no packet to transmit, STA will go to sleep just after the awake window for the maximum period of z or 97.29 ms.

6.3. Modeling Sleep per Packet and Percentage of Energy Saving. Another important metric in this experiment is the sleep time per packet. A sleep segment always follows a successful batch transmission. Thus sleep per packet is the ratio of the sleep segment to the corresponding batch size. The distribution of sleep per packet can be derived from the probability of sleep conditioned on batch size a as follows:

$$\begin{aligned}&\Pr\{\tilde{s}(i) \le s \mid \tilde{a} = i\} \\ &\quad = \Pr\left\{\frac{\tilde{s}(i)}{i} \le \frac{s}{i} \mid \tilde{a} = i\right\} \\ &\quad = \Pr\left\{\tilde{S}_{\mathrm{pp}}(i) \le S_{\mathrm{pp}} \mid \tilde{a} = i\right\}, \quad i \ne 0.\end{aligned} \tag{9}$$

In the above equation $\Pr\{\tilde{S}_{\mathrm{pp}}(i) \le S_{\mathrm{pp}} \mid \tilde{a} = i\}$ is the distribution of sleep per packet conditioned on batch i. For a batch of size i with corresponding sleep segment s, the sleep per packet, $S_{\mathrm{pp}} = s/i$. The $\Pr\{\tilde{s}(i) \le s \mid \tilde{a} = i\}$ can be found in (7). Therefore, from the total probability theorem the distribution of sleep per packet becomes

$$\Pr\left\{\tilde{S}_{\mathrm{pp}} \le S_{\mathrm{pp}}\right\} = \sum_{i=1}^{a_{\max}} \Pr\left\{\tilde{S}_{\mathrm{pp}}(i) \le S_{\mathrm{pp}} \mid \tilde{a} = i\right\}\Pr\{\tilde{a} = i\}. \tag{10}$$

The percentage of energy saving in the network is the ratio of the saved energy in PSM mode to the energy expended in the active mode to transfer an average size batch. Let $\text{Energy}_{\text{active}}$ and $\text{Energy}_{\text{PSM}}$ are the energy spent to transmit an average size batch if both nodes operate in active mode and power save mode, respectively. Let E_{tx} and E_{rx} are the energy expended to transmit and receive a packet, respectively. $\text{Power}_{\text{idle}}$ and $\text{Power}_{\text{sleep}}$ are the power dissipated in idle state and sleep state correspondingly. $\text{Batch}_{\text{avg}}$ is the average batch size and $\text{Sleep}_{\text{avg}}$ is the average sleep time. Hence percentage of energy saving is

$$\frac{\text{Energy}_{\text{active}} - \text{Energy}_{\text{PSM}}}{\text{Energy}_{\text{active}}} = \frac{2 \times \text{Sleep}_{\text{avg}} \times \left(\text{Power}_{\text{idle}} - \text{Power}_{\text{sleep}}\right)}{\left(E_{\text{tx}} + E_{\text{rx}}\right) \times \text{Batch}_{\text{avg}} + 2 \times \text{Power}_{\text{active}} \times \text{Sleep}_{\text{avg}}}, \tag{11}$$

where $\text{Batch}_{\text{avg}} = \sum_{j=0}^{a_{\max}} \pi_j \times j$ can be evaluated from (3) and $\text{Sleep}_{\text{avg}} = \int_0^{S_{\max}} sf(s)ds = S_{\max} - \lim_{\Delta s \to 0} \sum_{\Delta s} F(\Delta s) \times \Delta s$ can be evaluated from (8). The $f(s)$ and $F(s)$ are the PDF and CDF of sleep time, respectively. In this modeling it is assumed that the $P_{\text{tx}} = P_{\text{rx}} = \text{Power}_{\text{idle}} = 750\,\text{mW}$ and $\text{Power}_{\text{sleep}} = 50\,\text{mW}$ [7]. P_{tx} and P_{rx} are transmission and receiving power, respectively.

6.4. Modeling Average Queueing Delay. Let average batch size in the peer specific buffer is $\text{Batch}_{\text{avg}}$ and average service time of a single packet is $E[\widetilde{X}_p]$. From (3), $\text{Batch}_{\text{avg}} = \sum_{j=0}^{a_{\max}} \pi_j \times j$. Suppose a PSP started with average batch of size $\text{Batch}_{\text{avg}}$. Therefore, after the transmission of nth packet the number of packets in the system (both in peer specific buffer and in MAC queue) $\widetilde{P}_n = \text{Batch}_{\text{avg}} - n + \widetilde{k}$. Here $\widetilde{k}$ is the arrived packet from the upper layer within the service time of $x_n = E[\widetilde{X}_p] \times n$. Hence, the average number of packet the nth packet leaves behind is

$$\begin{aligned} E\left[\widetilde{P}_n\right] &= \sum_{k=0}^{\infty} \widetilde{P}_n \times \exp(-\lambda x_n) \times \frac{(\lambda x_n)^k}{k!} \\ &= \text{Batch}_{\text{avg}} - n + \lambda x_n. \end{aligned} \tag{12}$$

Thus the average number of packet in the system is

$$E\left[\widetilde{P}\right] = \frac{\sum_{n=0}^{\text{Batch}_{\text{avg}}} E\left[\widetilde{P}_n\right]}{\text{Batch}_{\text{avg}}}. \tag{13}$$

From the Little's theorem, average queueing delay,

$$D = \frac{E\left[\widetilde{P}\right]}{\lambda}. \tag{14}$$

7. Results of One Peer Link Operation

In this section the energy-related features are explored both numerically and analytically for one peer link topology. Figure 8 represents the batch size distribution for the entire period of operation. It is found that the batch size increases as the packet rate λ increases. For packet rate less than 400 pps (packet per second), the batch size is less than 50. These batches could be delivered completely within one beacon interval as 65 packets on average are possible to transfer in a single beacon period for this setup. For average packet rate of 400 pps (≈40 packet/beacon interval), around 2% of the batches take more than one beacon interval to be delivered completely. The batch size increases rapidly after 400 pps. As shown in Figure 8, for 500 pps (≈50 packet/beacon interval), almost all the batches vary from 75 to 185; that is, they take $\lceil 75/65 \rceil = 2$ to $\lceil 185/65 \rceil = 3$ beacon intervals to be transferred. Figure 9 shows the average batch size in different packet rate. It increases linearly up to packet rate 400 pps and after 400 pps it builds up rapidly.

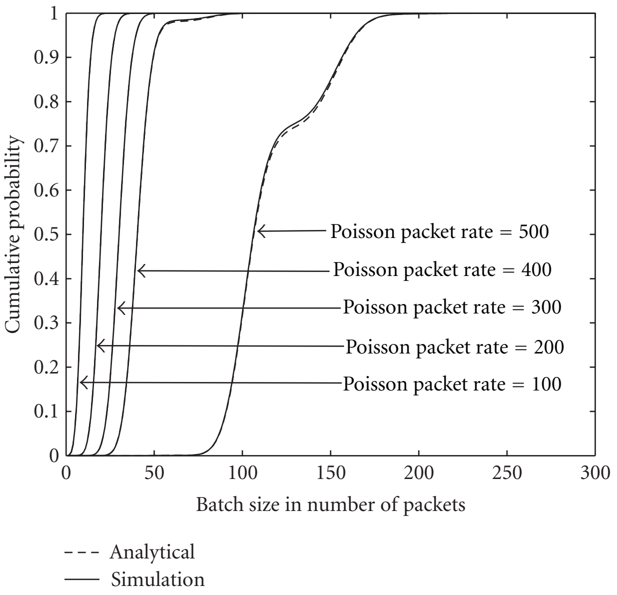


FIGURE 8: Batch size distribution.

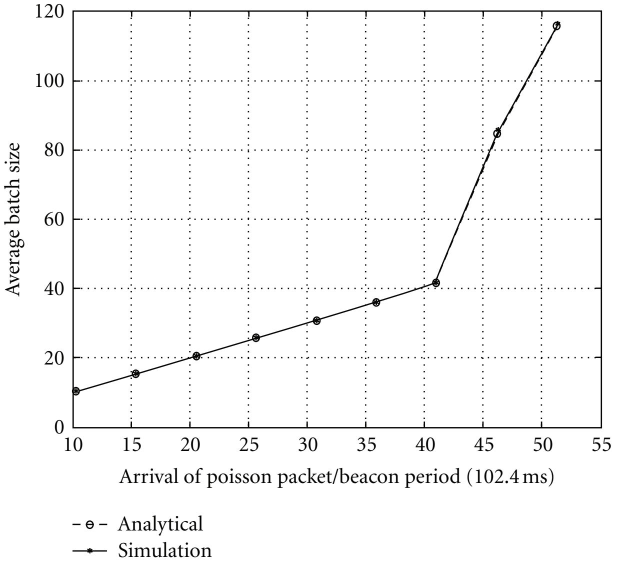


FIGURE 9: Average batch Size.

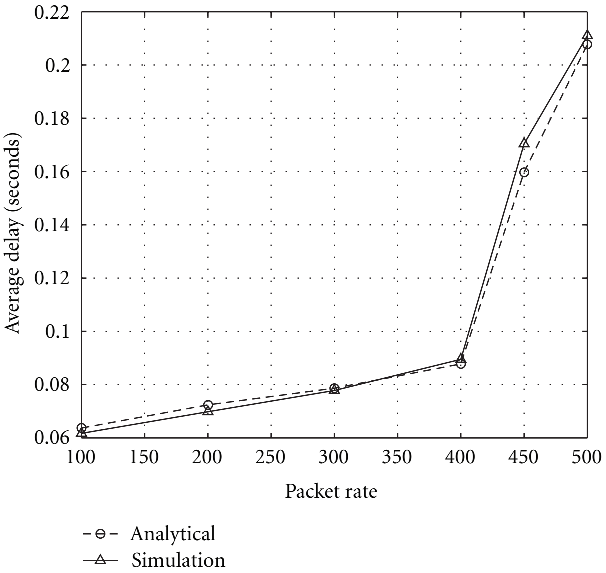


Figure 10: Average packet delay.

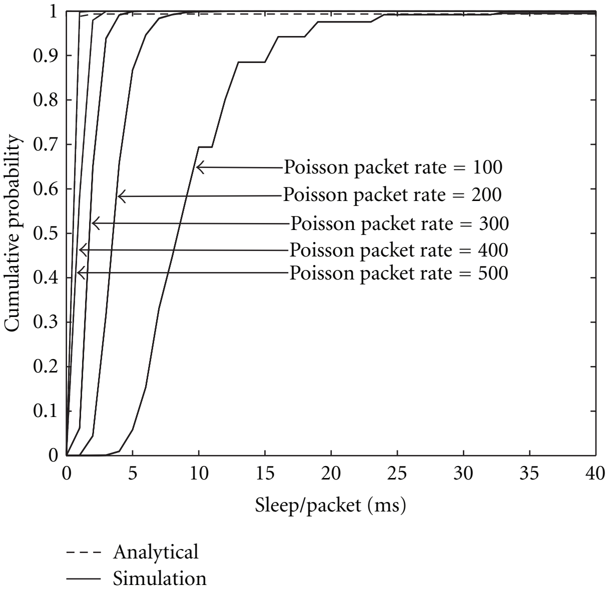


Figure 11: Sleep per packet distribution.

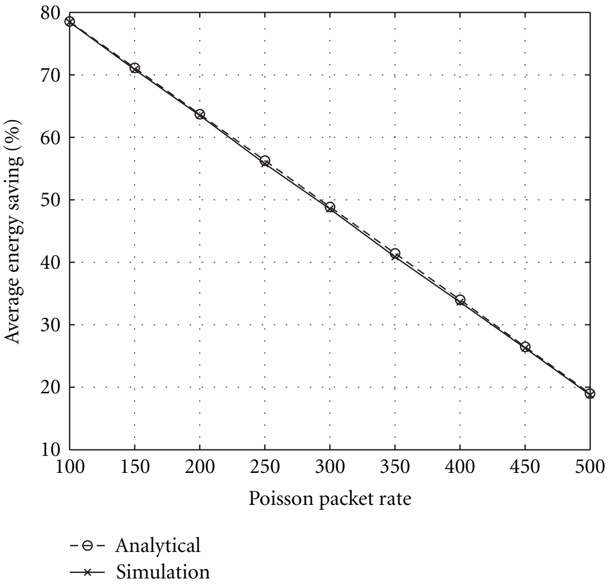


Figure 12: Percentage of energy saving.

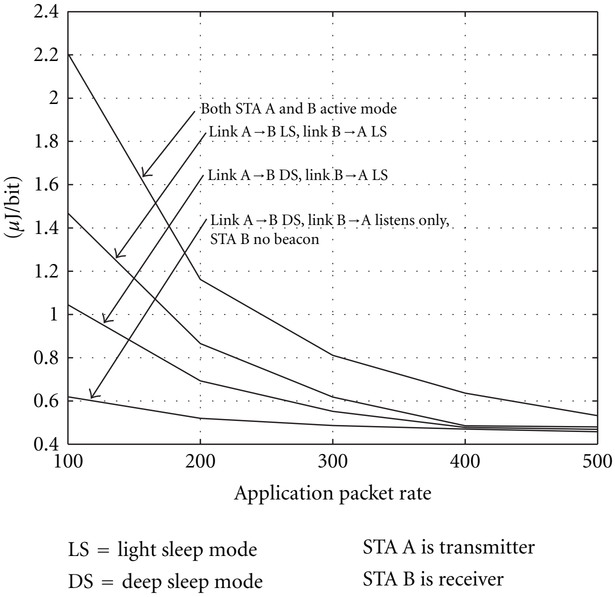


Figure 13: Energy consumption by the network in different power save mode for the link.

The impact of the batch size and its transfer time is clearly visible in average packet delay. The average packet delay in PSM is illustrated in Figure 10. The delay increases linearly up to 400 pps and after that it rises sharply. At 400 pps the average delay is around 88 ms and at 500 pps it rises to 210 ms. However, the average delay remains below 5.5 ms even in 500 pps in active mode operation.

The sleep per packet is illustrated in Figure 11. The sleep per packet is the ratio of sleep segment to the corresponding batch size. It is found that as the packet rate increases the sleep per packet decreases. Only the sleep segment does not describe the actual energy saving picture, because a big sleep segment may also follow a big batch or a long transmission. Therefore the ratio of sleep segment to the corresponding batch size could be a good metric to measure the actual scenario. As shown in Figure 11, in 90% cases, the sleep per packet is less than 15 ms at 100 pps and it is less than 1 ms at 500 pps.

The percentage of energy saving for a simple energy model [7] is plotted in Figure 12. At packet rate 100 pps the percentage of energy saving is around 79% and decreases linearly as the packet rate increases. At 500 pps it is almost 19%.

For a realistic energy model [18], only the simulation results of energy/bit in the network for different link specific power modes are plotted in Figure 13. Energy/bit is the

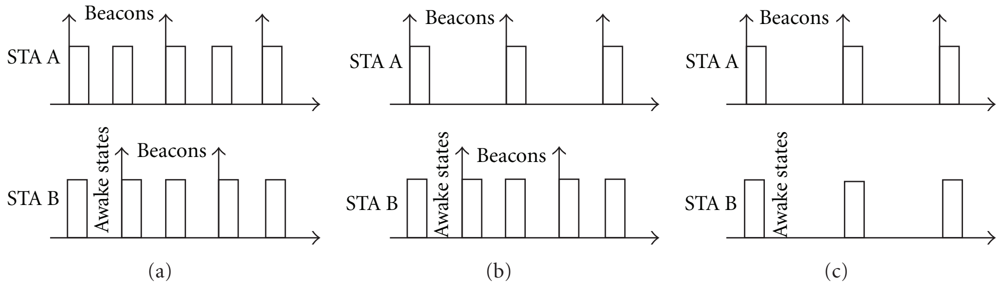


Figure 14: Wake-up patterns of STA A and STA B for different link specific power modes. (a) Link A → B light sleep and Link B → A light sleep. (b) Link A → B deep sleep and Link B → A light sleep. (c) Link A → B deep sleep and Link B → A listens only A's beacon. STA B transmits no beacon.

expended energy in the network to transfer a single bit successfully. The Figure shows that as the rate increases, the energy per bit decreases. In PSM operation, a fixed amount of energy is required to maintain the link regardless of data. Link maintenance requires periodic wake-up for listening peers, transmitting beacon, and switching and alternating between doze and awake states. These activities consume a certain amount of energy.

As illustrated in Figure 13, the highest amount of energy is consumed while both transmitter and receiver STAs operate in active mode. At 100 pps the energy consumption per bit is around 2.2 μJ. The lowest amount of energy is consumed while the transmitter operates in deep sleep mode for the link and the receiver wakes up only to listen to the transmitter's beacon as in Figure 14(c). In this mode the energy consumption per bit is around 0.62 μJ/bit at application packet rate 100 pps. When receiver STAs operate in light sleep mode for the link, given that the mode of the transmitter STA remains unchanged as shown in Figure 14(b), the expenditure of energy per bit increases to 1.044 μJ/bit for the same rate. This is due to the one additional wake-up of the receiver per beacon interval, beacon transmission, and the increased switching number. For the same reason, when both transmitter and receiver operate in light sleep mode for the link as shown in Figure 14(a), the energy consumption increases to 1.46 μJ/bit.

8. Performance Study in Multihop Network

In this section, some experiments are carried out for a multihop network consisting of total eight STAs. As shown in Figure 15, two linear data flows, consisting of 4 STAs, are placed across each other. One data flow is from STA1 to STA4 and another is from STA5 to STA8. The number of data flows remains constant for the entire simulation. The performance is evaluated for two separate setups. It is assumed that the STAs are MCCA enabled as discussed in Section 3.2. In the first setup (Setup 1), all STAs are within the transmission range; that is, a STA can hear all STAs in the network. Thus, there are no hidden nodes. In the second setup (Setup 2), the transmission range is reduced to two hops; that is, a STA can hear only those STAs that are situated within two-hop distance. As a result, the transmitting STA1 and STA5 act as hidden node to each other.

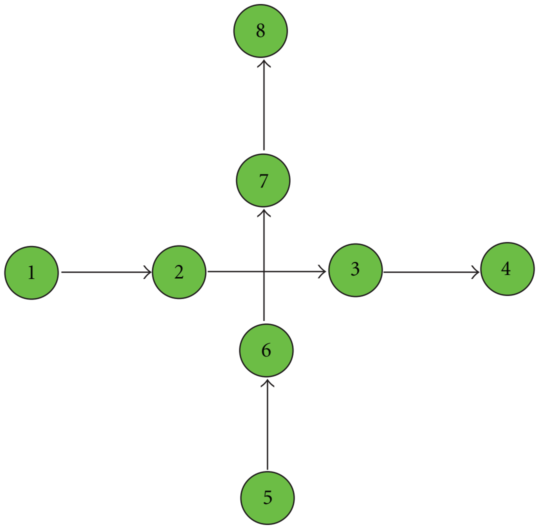


Figure 15: A multihop network consisting of total eight STAs. STA4 and STA8 are sink for source STA1 and STA5, respectively. Data flows are indicated by arrow.

Both of the above setups are studied for three power mode configurations of the network. The power mode configurations are as follows (i) all STAs are in PSM; (ii) only sink STAs are in active mode; (iii) all STAs are in active mode. In the first power mode configuration, all transmitting STAs are in deep sleep (DS) mode for its receiving STA and all receiving STAs are in light sleep (LS) mode for its transmitting STA. For example, STA2 is in DS mode for receiving STA3 but at the same time it is in LS mode for transmitting STA1. In the second configuration, the sink STA only operates in active (AC) mode for the link; that is, STA4 and STA8 are in AC mode for STA3 and STA7, respectively. The power modes of the rest PSM STAs are kept unchanged. In the third configuration, all STAs operate in active mode.

Figures 16(a) and 16(b) shows the throughput/flow in terms of packet per second for Setup 1 and Setup 2 correspondingly. In Setup 1, both active and power saving (PSM) operation give almost the same performance; that is, the system starts to cross the stability limit after 75 pps. After 75 pps, active mode performs better than PSM. However, in Setup 2, where hidden node exists, the PSM performs much better than active mode. In active mode, the system goes to

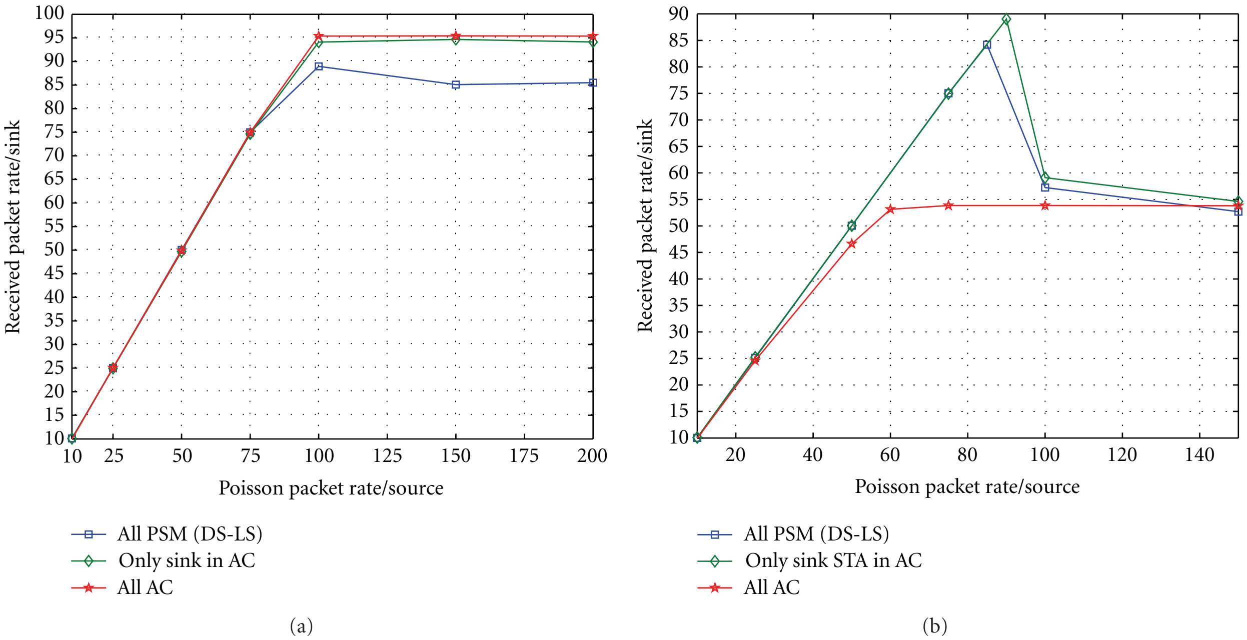


Figure 16: Throughput/sink for different packet rate. (a) For Setup 1 (no hidden node). (b) For Setup 2 (source STA1 and STA5 are hidden nodes).

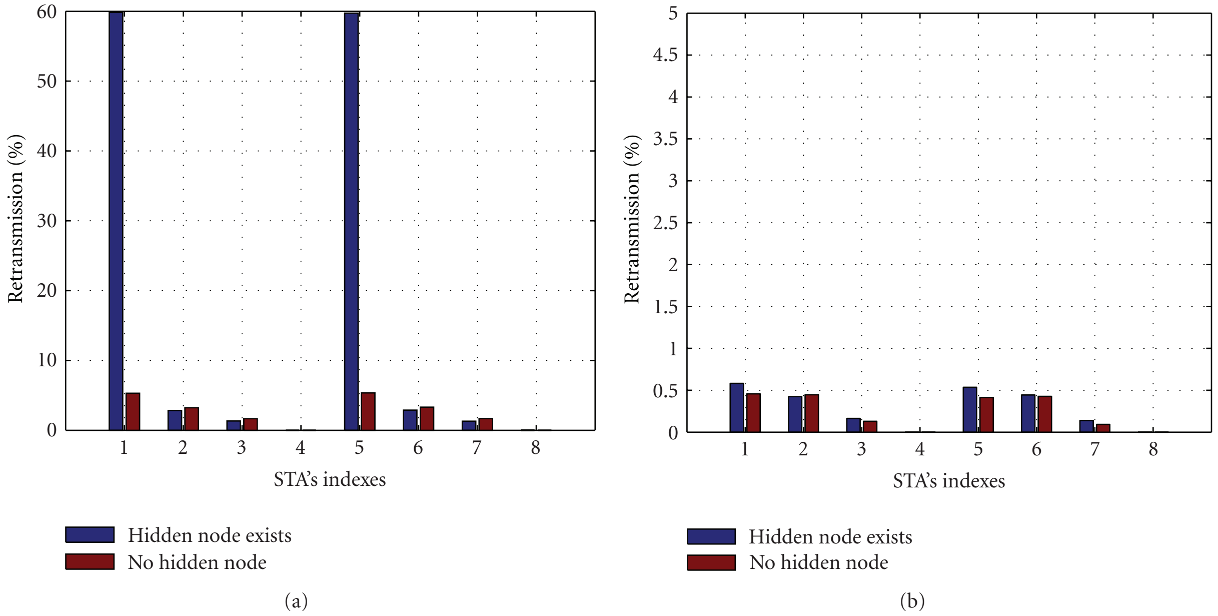
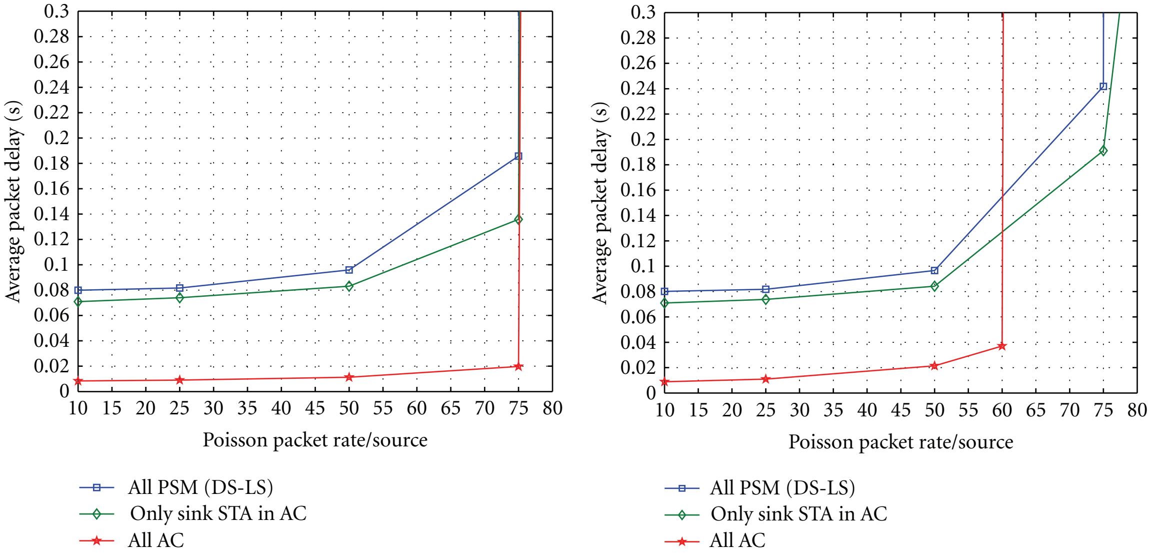


Figure 17: Percentage of retransmission of each STA at 50 pps. (a) All STAs are in active mode. (b) All STAs are in PSM. At 50 pps the impact of hidden node is negligible on PSM operation when compared with active mode.

saturation after 60 pps. On the other hand, in PSM, it still operates linearly up to 75 pps. The reason of this response is explained in Figure 17. After 100 pps, that is, at high packet rate, the network throughput is almost same both in PSM and in active mode.

In Figure 17, the percentage of retransmission of each STA is depicted for both Setup 1 and Setup 2 at 50 pps. The percentage of retransmission is the ratio of the number of retransmitted packets to the total number of transmission. The results for Setup 1 and Setup 2 are shown by blue and red bar, respectively. In active mode, as shown in Figure 17(a), the percentage of retransmission at source STA1 and STA5 is around 60% due to the hidden node problems. Basically, this is the main bottleneck to achieve linear operation at high packet rate in Setup 2 as described in Figure 17(b). However, in PSM as shown in Figure 17(b), the percentage of retransmission is negligible in both Setup 1 and Setup 2. Suppose in PSM, the percentage of retransmission of



(a)

(b)

FIGURE 18: The average packet delay over 3 hop. (a) For Setup 1 (no hidden node) (b) For Setup 2 (hidden node exists). Hidden node has significant impact on active mode operation.

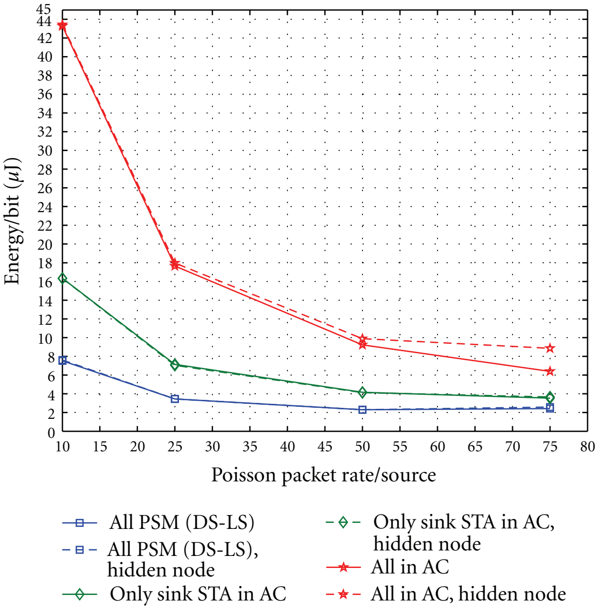


FIGURE 19: Energy/bit without or with hidden nodes. Solid lines are for Setup 1 (no hidden node) and dotted lines are for Setup 2 (hidden node exists).

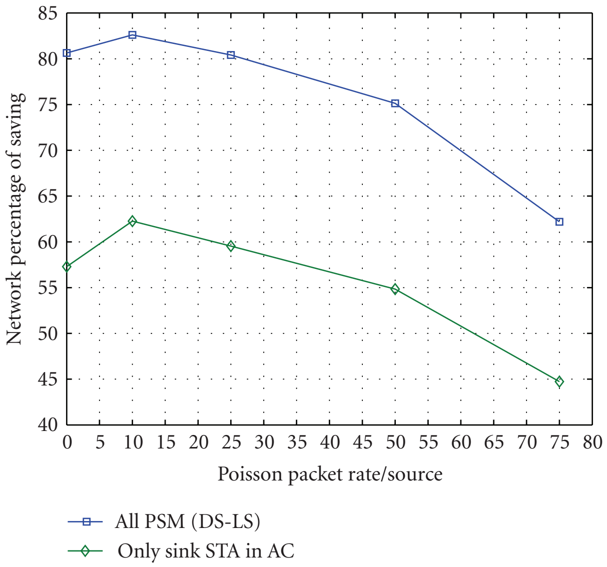


FIGURE 20: Percentage of saving for Setup 1 where no hidden node exists.

source STA1 and STA5 is only 0.582% in the existence of hidden node and it is 0.457% without hidden node. Basically, in IEEE 802.11s, the beacon transmission times of the STAs are evenly distributed in a beacon interval and so the transmission times of the PSM STAs are also distributed evenly as discussed in Section 3.2. As a result, the probability of collision between two transmitting STAs is low as long as a STA on average can complete its transmission before another STA wakes up to transmit. Thus, at high packet rate the number of collision increases. For example, for hidden node, the percentage of retransmission at source STAs rises from 0.582% to 8.30% as packet rate increases from 50 pps to 75 pps.

The average packet delay over three-hop data flow is explored in Figure 18. The delay here is the time difference between packet generation and its successful reception by the MAC of the sink STA. As shown in Figure 18, the average delay increases with the packet rate. For Setup 1, the average delay is less than 20 ms in active mode and is less than 185 ms in PSM for packet rate below 75 pps. For Setup 2, the average delay in active mode rises sharply after 60 pps and at 75 pps it is around 23 second. However, at the same packet rate of

75 pps the PSM experiences delay below 242 ms. The figure suggests that the impact of hidden node on delay in PSM operation is not as significant as it is in active mode. Still in PSM, a large amount of delay comes from the waiting time of the packet in the peer specific buffer. In IEEE 802.11s, a PSM STA can transmit to its AC mode peer as soon as it receives packet from the upper layer. In order to do so, it can immediately wake up from doze state if necessary. This reduces the queueing delay of that particular hop. The effect of this operation is also checked in this experiment. It is found that at 50 pps, the average delay in the last link reduces from 19 to 5.5 ms when sink STA switches to AC mode for this link. Figure 18 illustrates that this change in power modes also helps to improve the overall delay. However, this costs some extra energy.

The energy/bit in various operating mode is illustrated in Figure 19. The energy/bit is the expended energy in the whole network to transfer a single bit from source to sink successfully. In PSM, the lowest amount of energy is consumed. It increase when sink STAs switche to active mode to reduce the delay of its link. The highest amount of energy is consumed if all STAs operate in active mode. This consumption can further increase in the presence of hidden node. As shown in Figure 19, at high packet rate, the impact of hidden node in energy consumption is prominent in active mode operation. However, in case of PSM, this impact is negligible as hidden nodes don not start to transmit their buffered packet at the same time as discussed before.

The percentage of saving for Setup 1 is depicted in Figure 20. The percentage of saving is calculated for the whole network. The percentage of saving is the percentage of saved energy to successfully transfer a single bit in PSM when compared to active mode operation. One has

$$\text{Saving} = \frac{\text{Energy/bit in AC} - \text{Energy/bit in PSM}}{\text{Energy/bit in AC}} \times 100. \tag{15}$$

When there is no packet at all, the percentage of saving is around 80% in PSM and it is around 57% when only sink operates in AC. In PSM operation, a fixed amount of energy is required to maintain the link regardless of data. Link maintenance requires periodic wake-up for listening peers, transmitting beacon, and switching and alternating between doze and awake state. These activities consume a certain amount of energy. At 10 pps the saving is the highest and starts to fall with the increase of packet rate. If all STAs are in PSM, the saving is around 62% near the stability limit at 75 pps.

9. Conclusions

In this paper, the energy consumption for the IEEE 802.11s link specific PSM for one peer link operation has been analyzed both numerically and analytically for the first time. Later the simulation study is extended for a multihop network consisting of eight STAs. The study suggests that at the cost of increased packet delay, the IEEE 802.11s PSM operation not only saves a large amount of energy but also provides almost the same throughput as the active mode operation offers. For a large network the energy saving could be as high as eighty percent when compared with active mode operation. Especially in the presence of hidden node, the PSM can perform much better than active mode, if the nodes avoid simultaneous operation. However, the average packet delay in PSM is large when compared with the active mode operation. At very small load, the average packet delay over three-hop data flow is around 10 ms in active mode and it increases to 80 ms in PSM. Still, this delay is adjustable. By switching to AC mode, receiving STAs can reduce the link delay considerably. The study shows that the link delay reduces from 19 to 5.5 ms as receiving STA switches to active mode for the link. This clearly indicates that there is a trade-off between delay and energy and by choosing suitable power modes for the link an IEEE 802.11s network can adjusts its delay, throughput, and energy consumption. In future, the critical parameters in PSM operation could be investigated thoroughly to set a network in optimum point of operation.

Appendix

In this paper $F_{\Delta,a}(t)$ is approximated by truncated Gaussian CDF for large value of a. The value of $F_{\Delta,a}(t)$ can be calculated from the Q function as follows:

$$F_{\Delta,a}(t) = \Pr\left\{\widetilde{\Delta}(a) \le t\right\} = 1 - Q\left(\frac{t - E\left\{\widetilde{\Delta}(a)\right\}}{\sqrt{\operatorname{var}\left\{\widetilde{\Delta}(a)\right\}}}\right). \tag{A.1}$$

Here $E\{\widetilde{\Delta}(a)\}$ is the expected value of $\widetilde{\Delta}(a)$ and $\operatorname{var}\{\widetilde{\Delta}(a)\}$ is the variance. For a batch of size a, the maximum value of $\widetilde{\Delta}(a)$ is $a \times \mathrm{CW}_{\min}$. As it is assumed there is no contention or packet loss in the network, the contention window size will remain fixed to $\mathrm{CW}_{\min}$ all the time.

References

[1] C. M. Chao, J. P. Sheu, and I. C. Chou, "An adaptive quorum-based energy conserving protocol for IEEE 802.11 ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 5, pp. 560–570, 2006.

[2] "Wireless LAN medium access control (MAC) and physical layer (PHY) specification," ANSI/IEEE Std 802.11, 1999.

[3] X. Pérez-Costa and D. Camps-Mur, "AU-APSD: adaptive IEEE 802.11e unscheduled automatic power save delivery," in *Proceedings of the IEEE International Conference on Communications, (ICC '06)*, pp. 2020–2027, Istanbul, Turkey, July 2006.

[4] V. Namboodiri and L. Gao, "Energy-efficient voip over wireless LANs," *IEEE Transactions on Mobile Computing*, vol. 9, no. 4, pp. 566–581, 2010.

[5] Y. H. Zhu and V. C. M. Leung, "Efficient power management for infrastructure IEEE 802.11 WLANs," *IEEE Transactions on Wireless Communications*, vol. 9, no. 7, pp. 2196–2205, 2010.

[6] "Wireless LAN ,medium access control (MAC) and physical layer (PHY) specification," Amendment 10:Mesh Networking IEEE P802.11s/D3.0, 2009.

[7] R. Krashinsky and H. Balakrishnan, "Minimizing energy for wireless web access with bounded slowdown," in *Proceedings of the 8th Annual International Conference on Mobile Computing*

and Networking, pp. 119–130, Atlanta, GA, USA, September 2002.

[8] H. Yan, S. A. Watterson, D. K. Lowenthal, K. Li, R. Krishnan, and L. L. Peterson, "Client-centered, energy-efficient wireless communication on IEEE 802.11b networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 11, pp. 1575–1590, 2006.

[9] H. Zhu and G. Cao, "On supporting power-efficient streaming applications in wireless environments," *IEEE Transactions on Mobile Computing*, vol. 4, no. 4, pp. 391–403, 2005.

[10] C. H. Gan and Y. B. Lin, "An effective power conservation scheme for IEEE 802.11 wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 4, pp. 1920–1929, 2009.

[11] H. Lei and A. A. Nilsson, "Queuing analysis of power management in the IEEE 802.11 based wireless LANs," *IEEE Transactions on Wireless Communications*, vol. 6, no. 4, pp. 1286–1294, 2007.

[12] S. Baek and D. C. Bong, "Performance analysis of power save mode in IEEE 802.11 infrastructure WLAN," in *Proceedings of the International Conference on Telecommunications, (ICT '08)*, pp. 1–4, Petersburg, Russia, June 2008.

[13] H. Memarzadeh, M. Dehghan, and S. Jabbehdari, "A new quorum-based power saving protocol to maintain QoS for MANETs with burst traffics," in *Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications Workshops, (WAINA '10)*, pp. 674–679, Perth, Australia, April 2010.

[14] J. Lee and S. Kim, "Mathematical system modeling and dynamic resource allocation through Kalman filter based prediction in IEEE 802.11 PSM," in *Proceedings of the IEEE International Conference on Industrial Technology, (ICIT '09)*, Gippsland, VIC, Australia, February 2009.

[15] A. Bhardwaj, Divya, and S. Sofat, "An efficient energy conserving scheme for IEEE 802.11 ADHOC networks," in *Proceedings of the 4th IEEE and IFIP International Conference on Wireless and Optical Communications Networks, (WOCN '07)*, pp. 1–5, Singapore, July 2007.

[16] Q. Chen, F. Schmidt-Eisenlohr, D. Jiang, M. Torrent-Moreno, L. Delgrossi, and H. Hartenstein, "Overhaul of IEEE 802.11 modeling and simulation in NS-2," in *Proceedings of the 10th ACM Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems*, pp. 159–168, New York, NY, USA, October 2007.

[17] Wolfram Mathworld, http://mathworld.wolfram.com/UniformSumDistribution.html.

[18] N. Li, Y. Xu, and S. L. Xie, "A power-saving protocol for Ad hoc networks," in *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing, (WCNM '05)*, pp. 808–811, China, September 2005.

Hindawi Publishing Corporation
Journal of Computer Networks and Communications
Volume 2012, Article ID 870281, 20 pages
doi:10.1155/2012/870281

Research Article

Effective Stochastic Modeling of Energy-Constrained Wireless Sensor Networks

Ali Shareef and Yifeng Zhu

Department of Electrical and Computer Engineering, University of Maine, Orono, ME 04469, USA

Correspondence should be addressed to Yifeng Zhu, yifeng.zhu@maine.edu

Received 16 June 2012; Accepted 11 September 2012

Academic Editor: Runhua Chen

Energy consumption of energy-constrained nodes in wireless sensor networks (WSNs) is a fatal weakness of these networks. Since these nodes usually operate on batteries, the maximum utility of the network is dependent upon the optimal energy usage of these nodes. However, new emerging optimal energy consumption algorithms, protocols, and system designs require an evaluation platform. This necessitates modeling techniques that can quickly and accurately evaluate their behavior and identify strengths and weakness. We propose Petri nets as this ideal platform. We demonstrate Petri net models of wireless sensor nodes that incorporate the complex interactions between the processing and communication components of an WSN. These models include the use of both an open and closed workload generators. Experimental results and analysis show that the use of Petri nets is more accurate than the use of Markov models and programmed simulations. Furthermore, Petri net models are extremely easier to construct and test than either. This paper demonstrates that Petri net models provide an effective platform for studying emerging energy-saving strategies in WSNs.

1. Introduction and Motivations

Application for wireless sensor networks (WSNs) has abounded since their introduction in early 2000. WSNs are being used from surveillance, environmental monitoring, inventory tracking, and localization. A sensor network typically comprises of individual nodes operating with some limited computation and communication capabilities, and powered by batteries with limited energy supply. Furthermore, these networks are situated at a location where they may not be easily accessible. Their distributed nature, small footprint, cheap, and wireless characteristics make them very attractive for these outdoor, unattended, and hostile environment applications.

One of the motivating visions of WSNs was large-scale remote sensing such as large areas of a rainforest for environmental parameters such as humidity and temperature. However, given the remoteness of such a site, this can be a challenging problem. Modern WSNs were proposed for solving such problems, and it was envisioned that these WSN nodes could be sprinkled over an area from the back of an airplane as it flew over such an area. The nodes wherever they fell would automatically set up an ad hoc network, collect the necessary sensory information, and route the information to a base-station. Although great strides have been made in WSN designs and implementation, we are nowhere near meeting this original motivating problem.

One reason why this original problem has been difficult to solve is that WSNs are still relatively expensive in large quantities. However, the much larger problem is the limited energy available on these devices. The utility of WSNs is limited to the life of the battery under the energy consumption rates. While energy harvesting in WSNs is an active research area [1], generally this is not feasible yet for entirely sourcing the energy needs of an WSN. WSNs are still very much bound to batteries.

An avenue for mitigating this energy dilemma is through the design of energy-efficient communication and active/sleep scheduling algorithms. In this way, a vital resource can be rationed to last a much longer time. This minimizes the overall maintenance and replacement costs of a WSN network. However, proposing energy-efficient designs requires a detailed understanding of the energy consumption behavior of the nodes which comprise WSNs. This detailed understanding arises from accurate implementation of models for these nodes and analyzing these models under a variety of different states.

For example, many processors are available today that are capable of moving to a sleep mode where they consume minimal energy. However, should a processor be put to sleep immediately after computation, or after some time has elapsed? Or even, perhaps it should never be put to sleep? If it is best to move the processor to sleep after a time delay, what should this delay or *Power Down Threshold* be for a given system? Keep in mind, there is a high energy cost associated with waking up a processor from a sleep mode due to the internal capacitances. If the threshold is too short, then the CPU goes to sleep more often, and there is a stiff price to be paid each time the CPU needs to be woken up. If the threshold is too long, the CPU idles consuming energy wastefully. Nevertheless, there is an optimum threshold that results in the least amount of energy consumption that strikes an optimum balance between putting the CPU to sleep and maintaining it in an active mode. This threshold can also be referred to as break-even time [2].

Another example of emerging technology that can be exploited in wireless sensor networks is the use of processors that have dynamic voltage scaling capabilities. In these processors, the voltage and clock frequency can be dynamically adjusted to obtain a minimum clock frequency to complete the task while using minimal energy [3, 4]. Currently the two types of DVS systems available are those that stop execution while changing voltage and frequency and those that are capable of changing its operating parameters at run time [5, 6]. However, in order to begin investigating energy optimization techniques, such as answering the questions given earlier, we need to devise models that can be used to accurately compute the energy consumption of a wireless sensor node. This need motivates the research presented in this paper.

Currently, there are two classes of modeling and simulating techniques: stochastic and simulation-based methods. Each has its strengths and weaknesses. We propose another method of modeling that has not been used in the past to model WSNs: Petri nets. This paper studies two methods of energy modeling: Markov chains, and Petri nets [7, 8]. These modeling techniques will be compared against a programmed simulation model. This paper makes the following contributions.

(i) We successfully model a processor using a Markov model based on supplementary variables; we also model a processor using colored Petri nets. These models are capable of estimating the average energy consumption of a processor that can power down to a sleep mode. While Markov models have long been used for modeling systems, we show that for estimating CPU energy consumption, the Petri net is more flexible and accurate than the Markov model.

(ii) Using Petri nets, we develop a model of a wireless sensor node that can accurately estimate the energy consumption. We successfully apply this model to identify the optimum powering down threshold for a given wireless sensor network application.

(iii) Our model of a sensor node based on Petri net can be utilized to construct a wireless sensor network. This provides a platform that can be used to study energy consumption at the network layer for application such as cross-layer energy-aware routing.

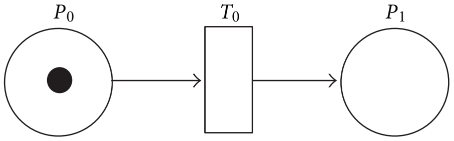


Figure 1: Example of Petri net.

The paper is organized as follows. Section 2 gives a short introduction to Markov models and Petri nets and discusses related work. Section 3 presents the CPU energy models and Section 4 validates the CPU models. Section 6 presents the model for a wireless sensor node and Section 4 uses this model to study the energy optimal *Power Down Threshold*. Section 8 introduces the use of Petri nets in modeling wireless sensor networks. Section 9 concludes this paper.

2. Background and Related Work

2.1. Introduction to Markov Models and Petri Nets. Traditionally, Markov models have been used; however, they are very restrictive in the type of behaviors that can be modeled. A Markov model is a discrete-time stochastic process composed of event chains with Markov property, meaning that the next state in a Markov model is only dependent upon the current state instead of previous ones.

The advantage of using Markov chains for modeling systems is that once the appropriate equations are derived, the average behavior can be easily obtained by evaluating the appropriate equations. However, the task of obtaining the equations relevant to the system can be time-consuming, if not impossible.

Petri nets, on the other hand, are very powerful tools that can be utilized to build statistical models of complicated systems that would otherwise be very difficult. Petri nets is a simulation-based approach for modeling. A Petri net is a directed graph of nodes with arcs. Nodes are referred to as places and are connected to transitions with arcs. In this paper, arcs will be drawn as directed arrows.

An example of a simple Petri net is shown in Figure 1 that contains two places P_1 and P_0, and a transition T_0. The input place of T_0 is P_0, and the output place of T_0 is P_1. T_0 is enabled only if P_0 contains as many tokens as specified by the arc. In this example, T_0 is enabled because it requires only one token in P_0. Once a transition is enabled, it will fire according to a specified timing parameter. During the process of firing, a transition will remove a number of tokens, as specified by the arc, from the input place and deposit these tokens in the output place. If an immediate transition is used, the transition will fire as soon as it is enabled. Deterministic transitions fire if some predetermined time has passed after it has been enabled, and exponential transitions fire if some random time in some time interval has passed. A Petri net can be used to model the behavior of a system utilizing this flow of tokens to represent movement of control through the different processes of the system. Statistical analysis of the number of tokens in a specific place or the number of particular transitions can provide insights into the behavior of the modeled system. Many software packages are available that can be used to design and perform analysis upon Petri nets. The software that will be used in this study is TimeNET 4.0 [9].

TimeNET 4.0 [9], written in JAVA, is a discrete event simulator of Petri nets. A standalone GUI program called Platform Independent Editor for Net Graphs (PENG) is used to build the Petri net model. The model is translated to a XML schema file which describes all the places, transitions, tokens, and arcs. Thereafter, an executable is generated to simulate the model. Depending on the model implemented in TimeNET, different solution techniques are applied. For example, TimeNET can compute the steady-state solutions for Petri nets with mutually exclusive nonexponential firing parameters. However, for models that result in more than one deterministic transition becoming enabled at a time, appropriate techniques are applied to maintain statistical steady-state accuracy as the model is simulated over an extended time.

2.2. Related Work. Many techniques have been proposed for modeling embedded systems and minimizing the energy consumption. As discussed earlier, existing method includes stochastic Markov models and programmed simulation methods. There are many proposed methods based on Markov models [1, 10]. In [11], Markov models are used to model active and sleep capabilities of a node and the resulting energy and performance characteristics. Steady-state probability equations are derived that describe the number of packets that must be serviced in different modes of operation.

Another proposed method [12] employs a stochastic queueing model to develop a cross-layer framework. This framework is utilized to find the distribution of energy consumption for nodes for a given time period. When the time period is long, it is demonstrated that the distribution approaches a normal distribution. This information is then used to predict node and network lifetime. This framework is also used to identify relationships between energy consumption and network characteristics such as network density, duty cycle, and traffic throughput.

Jung et al. in [13] proposed the use of Markov models to model nodes in a wireless sensor network. However, Jung's focus was on identifying the power consumption rates between trigger-driven and schedule-driven modes of operation and, as a result, the lifetimes of node using these methods. We feel that the use of Markov models is cumbersome and results in limitation of the model due to the inability of Markov models to account for fixed constant arrival or service rates.

Coleri et al. [14] have demonstrated the use of a Hybrid Automata to model TinyOS and hence the resulting power consumption of the nodes. Coleri was able to analyze the nodes in the network on a much wider scale than what is presented in this paper. By utilizing the TinyOS framework, an Automata model was constructed that resulted in the ability to analyze power dissipation of a node based on its location in the sensor network. Finite Automata have also been used in [15].

Liu and Chou [2] present a model based on tasks, constraints, and schedules. Energy minimization is proposed through the use of scheduling for DVS processors capable of executing at different modes. Other works include [16, 17].

One of most common methods of modeling is through the use of programmed simulation using tools such as NS2, OMNet++, OPNET, and TOSSIM. Each of these tools have their strengths and weakness as described in [18].

In [19], the author propose the use of Petri nets for modeling the behavior and characteristics of WSN nodes in what they define as intelligent wireless sensor networks (IWSNs). Their Petri net models can be used to simulate the actual applications, and they present results of a target tracking system prototype that they implement using a Petri net tool called integrated net analyzer (INA).

We have not found any literature that discusses the use of Petri nets for modeling energy consumption of nodes in a wireless sensor network [7, 8]. We attempt to view the minimization of energy from a systems standpoint rather than just focus on the CPU. Because the CPU is intricately associated with the system, all the other parameters of the system affect the energy consumption associated with the CPU.

3. Evaluation of a CPU with a Markov Model and Petri Net

Intrinsically, embedded systems operating in a wireless sensor network offer great potential for power minimization. Generally, the level of computation required is low and usually interspersed with communication between other nodes in the network. The power consumption of the CPU can be minimized by moving to a low power mode and conserving energy when it is not directly involved in any computation.

3.1. Open versus Close Workload Model. There are two types of workload generators that are widely used for generating jobs for a simulation. Both are used quite frequently depending upon the application. One is called the closed workload generator and the other is called the open workload generator. In a closed workload generator, a new job cannot be generated until the system has completed servicing the current job. This can be used to model schedule-driven systems that poll at given intervals for task requests. Since jobs are not generated until the current job is processed, this workload model is easy to implement and analyze. In open workload generators, on the other hand, jobs arrive independent of the state of the system. These can be used to model trigger-driven systems that service requests when an interrupt occurs. However, since jobs can arrive at any time, a buffer needs to be implemented to store those requests that arrive while the system is busy with another request. This workload model can

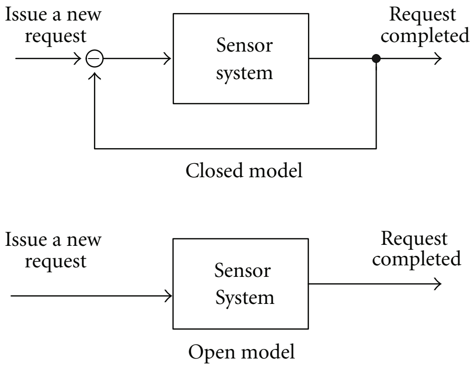


Figure 2: Closed workload generator model and open workload generator model.

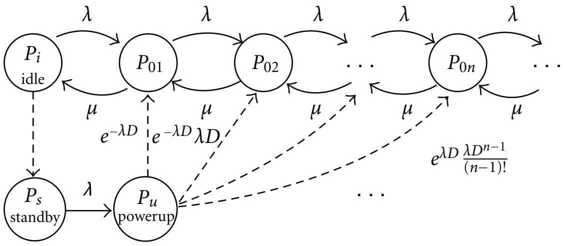


Figure 3: Birth-death process of CPU jobs.

be more difficult to implement and analyze. Figure 2 depicts both closed and open workload generators.

3.2. Modeling Energy Consumption of a CPU Using a Markov Model. An example of a Markov model of a CPU is given in Figure 3. The CPU "power-ups" (p_u) from some low power "standby" mode (p_s) when jobs begin arriving. The Markov model depicts the various increasing states (p_{01}, p_{02}, p_{03}, etc.) the CPU enters as the number of jobs increase under a given job arrival rate λ. The CPU services the jobs at rate μ and strives to move the CPU to lower states and eventually to the "idle state" (p_i). If the processor remains in the idle state for some time interval greater than some threshold, the CPU moves back to the "standby" (p_s) state.

In this example, the following assumptions are made.

(1) The request arrivals follow a Poisson process with mean rate λ.

(2) The service time is exponentially distributed with mean $1/\mu$.

(3) The CPU enters the standby mode (state p_s) if there are no more jobs to be serviced for a time interval longer than T.

(4) The power up process (state p_u) takes a constant time D.

The CPU model consists of a mix of deterministic and exponential transitions. While all transitions shown as solid lines in Figure 3 follow exponential time distributions, the transitions shown as dashed lines are deterministic. This includes the transition from the idle state to the standby state. The CPU enters the standby state after idling for a constant time threshold T. This power down transition depends on its history and is not memoryless. Accordingly, the CPU transitions cannot be modeled directly as a Markov process. Using the method of supplementary variables proposed in [20], we can derive an alterative set of state equations to approximate the transitions for stationary analysis. Let $X = [x_1, x_2]$ denote two age variables representing how long a deterministic transition has become enabled [21]. And let $\mathbf{P}_i(x_1)$ and $\mathbf{P}_u(x_2)$ be the age density functions with respect to x_1 in the idle state and with respect to x_2 in the power up state, respectively.

The state equations for this mixed transition process can be derived by the inclusion of two supplementary variables X. The deterministic transition from the idle state to the standby state can be modeled as below:

$$\begin{aligned} p_i &= \int_0^T \mathbf{P}_i(x_1)dx_1, \\ \frac{d}{dx_1}\mathbf{P}_i(x_1) &= -\lambda \mathbf{P}_i(x_1), \\ \mathbf{P}_i(0) &= \mu p_{01}, \\ \mathbf{P}_i(T) &= \lambda p_s, \end{aligned} \tag{1}$$

where $\mathbf{P}_i$ is an exponential function with coefficient λ (p_i is the steady-state probability of being in the idle state).

The deterministic power up process can be modeled as below:

$$\begin{aligned} p_u &= \int_0^D \mathbf{P}_u\ (x_2)dx_2, \\ \frac{d}{dx_2}\mathbf{P}_u(x_2) &= -\lambda \mathbf{P}_u(x_2), \end{aligned} \tag{2}$$

$$\mathbf{P}_u(0) = \lambda p_s. \tag{3}$$

In addition, when the system is stable, we have

$$(\lambda + \mu)\,p_{01} = \lambda p_i + \mu p_{02} + e^{-\lambda D}\mathbf{P}_u(0), \tag{4}$$

$$\begin{aligned} (\lambda + \mu)\,p_{0n} = {} & \lambda p_{0,n-1} + \mu p_{0,n+1} \\ & + e^{-\lambda D}\frac{(\lambda D)^{n-1}}{(n-1)!}\mathbf{P}_u(0) \quad \text{for } n \geq 2, \end{aligned} \tag{5}$$

$$1 = \sum_{n=1}^{\infty} p_{0,n} + p_i + p_s + p_u. \tag{6}$$

From (1), we can get

$$\begin{aligned} p_{01} &= \frac{\lambda}{\mu}e^{\lambda T}p_s, \\ p_i &= \left(e^{\lambda T} - 1\right)p_s. \end{aligned} \tag{7}$$

From (2), and (3), we have

$$p_u = \left(1 - e^{-\lambda D}\right)p_s. \tag{8}$$

We define the generating function of $p_{0,n}$ $(n = 1, 2, \ldots)$ as

$$G_0(z) = \sum_{n=1}^{\infty} p_{0,n} z^n. \tag{9}$$

We multiply (4) by z and (5) by z^n, add from $n = 1$ to ∞, use (3), (7), and (9), and get

$$G_0(z) = \frac{\lambda z p_s}{\mu - \lambda z}\left(e^{\lambda T} + \frac{e^{\lambda D(z-1)} - 1}{z - 1} z\right). \tag{10}$$

Thus, we get

$$G_0(1) = \frac{\lambda p_s}{\mu - \lambda}\left(e^{\lambda T} + \lambda D\right). \tag{11}$$

Substituting (7), (8), (9), and (11) into the normalization equation (6) gives

$$p_s = \frac{1 - \rho}{e^{\lambda T} + (1 - \rho)(1 - e^{-\lambda D}) + \rho \lambda D}, \tag{12}$$

where $\rho = \lambda / \mu$.

Consequently,

$$\begin{aligned} p_i &= \frac{(1 - \rho)\left(e^{\lambda T} - 1\right)}{e^{\lambda T} + (1 - \rho)(1 - e^{-\lambda D}) + \rho \lambda D}, \\ p_u &= \frac{(1 - \rho)\left(1 - e^{-\lambda D}\right)}{e^{\lambda T} + (1 - \rho)(1 - e^{-\lambda D}) + \rho \lambda D}. \end{aligned} \tag{13}$$

And the utilization is

$$G_0(1) = \frac{\rho\left(e^{\lambda T} + \lambda D\right)}{e^{\lambda T} + (1 - \rho)(1 - e^{-\lambda D}) + \rho \lambda D}. \tag{14}$$

Let $L(z) = \sum_{n=1}^{\infty} n p_{0n}\, z^n$, then $L(1)$ is the total number of jobs in the system:

$$\begin{aligned} L(z) &= \sum_{n=1}^{\infty} n p_{0n}\, z^n \\ &= z \frac{d}{dz}\left(\sum_{n=1}^{\infty} P_{0n}\, z^n\right) \\ &= z \frac{d}{dz} G_0(z). \end{aligned} \tag{15}$$

Incorporating (10) into (15), we can get the total number of jobs in the system as follows:

$$L(1) = \frac{\rho}{1 - \rho} \frac{e^{\lambda T} + (1/2)(1 - \rho)\lambda^2 D^2 + (2 - \rho)\lambda D}{e^{\lambda T} + (1 - \rho)(1 - e^{-\lambda D}) + \rho \lambda D}. \tag{16}$$

According to the Little's Law, the average latency for each job is

$$\tau = \frac{L(1)}{\lambda}. \tag{17}$$

Thus, the total running time is

$$\begin{aligned} T &= \frac{N}{\lambda} + L(1)\tau \\ &= \frac{N + L(1)^2}{\lambda}, \end{aligned} \tag{18}$$

where N is the total number of jobs.

And the total energy consumption is

$$\begin{aligned} E = &\left(p_i P_{idle} + p_s P_{standby} + p_u P_{powerup} + G_0(1) P_{active}\right) \\ &\times \frac{N + L(1)^2}{\lambda}, \end{aligned} \tag{19}$$

where P_{idle}, $P_{standby}$, $P_{powerup}$, and P_{active} are the power consumption rate in the idle, standby, power up, and active states, respectively; p_i, $p_s p_u$, and $G_0(1)$ are the probability that the system stays in the corresponding state.

3.3. CPU Energy Modeling Using a Petri Net.

As shown in the last section, the development of a Markov model for even a simple CPU is mathematically cumbersome especially when dealing with deterministic transitions. Any slight modifications to the model will entail that the equations be rederived again. Petri net on the other hand offers a more flexible approach.

Figure 4 shows an open model of an extended deterministic and stochastic Petri net (EDSPN) [22] modeling a minimizing power consumption system for a processor like the Markov model described earlier. The Petri net models a CPU that starts from some "stand by" state (*Stand_By*) and moves to an "on" state (*CPU_ON*) when jobs are generated. The CPU remains in the "on" state so long as there are jobs in the CPU buffer. If there are no jobs in the CPU buffer for some time interval as given by *Power_Down_Delay*, the CPU then moves to the "stand by" mode (*Stand_By*) to conserve power.

This model uses an open workload generator because when transition $T1$ fires to deposit a task in the *CPU_Buffer*, a token is moved back to place $P0$ which enables transition *Arrival_Rate* and allows another task to be generated. Table 1 lists the parameters of all the transitions in the Petri net. The names of the transitions in Figure 4 are listed in the first column, followed by the type of transition. Transitions that have a specified time parameter are listed in the "Delay" column. The last column indicates the priority of a transition in the event that there is a tie. Transitions with higher priority fire before other transitions.

The CPU is simulated by executing the Petri net using the following steps.

(1) Jobs are generated in place $P1$, when transition *Arrival_Rate* fires randomly in the interval $[0, 1]$ using an exponential distribution. A token in the place $P0$ is moved to $P1$ and enables $T1$.

(2) Transition $T1$ is an immediate transition and fires as soon as it is enabled. Also since $T1$ has the highest priority, it will fire before any other immediate transition if multiple immediate transitions are enabled

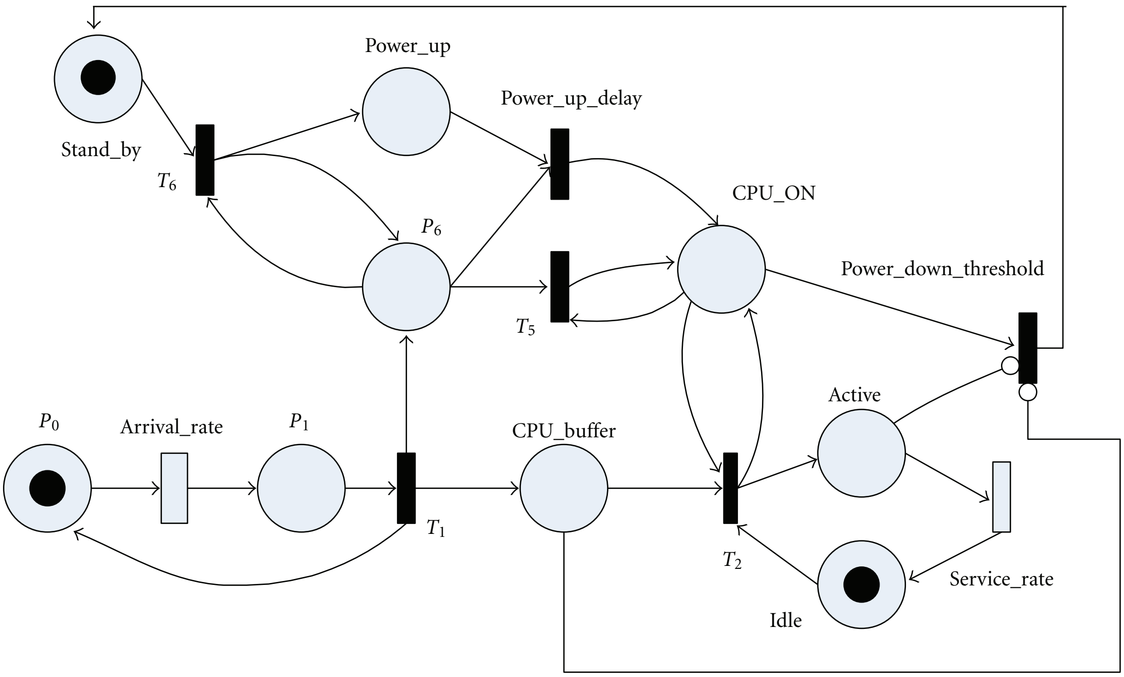


Figure 4: Petri net model of CPU.

Table 1: Petri net transition parameters for CPU jobs.

Transition	Firing distribution	Delay	Priority
AR	Exponential	ArrivalRate	NA
*T*1	Instantaneous	—	4
*T*2	Instantaneous	—	1
SR	Exponential	ServiceRate	NA
PDT	Deterministic	*PDD*	NA
*T*5	Instantaneous	—	2
*T*6	Instantaneous	—	3
PUD	Deterministic	*PUD*	NA

concurrently. When *T*1 fires, a token is removed from *P*1 and three tokens are generated and deposited in places *P*0, *P*6, and *CPU_Buffer*, respectively. Initially, CPU is in the *Stand_By* mode. When a job arrives, a token is deposited in place *P*6 and transition *T*6 is enabled.

(3) When *T*6 fires, one token from *Stand_By* and *P*6 are removed, respectively. Two new tokens are then generated, with one deposited in place *Power_Up* and the other in *P*6. The CPU has now moved to a powering up state. Transition *Power_Up_Delay* is now enabled with a token in *Power_Up* and in *P*6.

(4) Since transition *Power_Up_Delay* has a deterministic delay, the transition fires after a fixed time interval. When this happens, the two tokens from *Power_Up* and *P*6 are removed, and a token is deposited in place *CPU_ON*. The CPU is now "on" and ready to process job events.

(5) When *T*1 fires, a token is placed in *CPU_Buffer* in step 2. A token in *CPU_Buffer*, a token in *CPU_ON*, and a token in *Idle* enable the immediate transition *T*2. When *T*2 fires, one token, respectively from *CPU_Buffer*, *CPU_ON*, and *Idle* are removed and two new tokens are generated, with one deposited back in *CPU_ON* and the other in *Active*. The system is now in the processing state. With a token in *Active*, the transition *Service_Rate* is enabled.

(6) *Service_Rate* is an exponential delay and it will fire randomly after some time in the interval $[0, 0.1]$. After *Service_Rate* fires, the token is removed from *Active* and placed in *Idle*.

(7) In the event that another task arrives while the CPU is still "on" and processing other tasks (steps 1-2), a token will be deposited in *P*6 and *CPU_Buffer*. When the CPU is already "on", having a token in *P*6 will enable *T*5 to fire immediately. The tokens from both *P*6 and *CPU_ON* will be removed and a single token will be placed in *CPU_ON*. This is necessary because tokens cannot be allowed to accumulate infinitely in any place.

(8) The token deposited to *CPU_Buffer* will remain there until the CPU is idle as determined by a token in *Idle*. All jobs that arrive while the CPU is "on" will cause the Petri net to cycle through steps 7 and 8.

(9) However, in the event that the job arrival rate is very slow, the CPU may power down and move to the *Stand_By* state. This happens when there is a token in *CPU_ON* and no tokens in *Active* and *CPU_Buffer*, transition *Power_Down_Threshold* becomes enabled. The small circle at the ends of

Table 2: Simulation parameters.

Total simulated time	1000 seconds
Arrival rate	1 per second
Service rate	1 per second

the arcs from *Active* and *CPU_Buffer* specify this inverse logic. Since *Power_Down_Threshold* is a transition with deterministic delay, it will fire after a specified period *Power_Down_Delay* (*PDD*). When *Power_Down_Threshold* fires, a token from *CPU_ON* will be removed and transferred to *Stand_By*. The CPU has now moved to the *Stand_By* state.

By computing the average number of tokens in a certain place during the duration of the simulation time results in the "steady-state" percentage of time the CPU spends in that state. For example, the average number of tokens in *CPU_ON* will indicate the percentage of time the CPU was "on." The average number of tokens in *Power_Up* will indicate the "steady-state" percentage of time that the CPU was "powering up." Of course, these percentages are determined by the *ArrivalRate*, *Service_Rate*, *Power_Down_Delay*, and *Power_Up_Delay* delays. Once the percentages are obtained, they can be used to compute the total energy consumption of the system over time as given in (20):

$$\text{Total Energy} = \Big(P_{standby} \times p_{standby} + P_{powerup} \times p_{ppowerup} + P_{idle} \times p_{idle} + P_{active} \times p_{active}\Big) \times \text{Time}, \qquad (20)$$

where P_x is the power consumption rate and p_y is the steady-state probability of a specific state.

4. Comparison between Simulation, Markov Models, and Petri Net

We have developed a discrete event simulator that emulates the timings of state transitions of CPU. Equation (20) will be used to compute the total energy for the simulator as well. Table 2 lists the simulation time, arrival rate, and service rate parameters. The PXA271 Intel Processor whose power parameters are given in Table 3 [13] is used in this paper. We will compare the predicted steady-state probabilities and the energy estimates of the event simulator, the Markov model and the Petri net model while the *Power_Down_Threshold* is varied from 0.001 to 1 second and the *Power_Up_Delay* is fixed at 0.001, 0.3, and 10 seconds.

Figure 5 shows the percentage of time the CPU spends in the different states when *Power_Up_Delay* or the time for the CPU to "wake up" is fixed at 0.001 seconds. The *Power_Down_Threshold* is the length of time that the CPU waits in the *Idle* state before it transitions to the *Stand_By* mode.

Figure 5 allows us to study the effects of increasing the *Power_Down_Threshold*. Intuitively, it is obvious that as the *Power_Down_Threshold* increases, the *Idle* time increases appropriately as indicated in the figure. The amount of time in *Stand_By* decreases proportionally, and more and more time is spent in *Idle* and the CPU moves to *Stand_By* fewer times. This means that the time spent in powering up decreases as well because there are fewer times the CPU goes to *Stand_By*. Notice that the *Active* time remains constant indicating that for the most part the *Power_Down_Threshold* does not affect the utilization of the CPU.

Table 3: System model Petri net power parameters.

State	Power rate (mW)
CPU *Stand_By*	17
CPU *Idle*	88
CPU *Power_Up*	192.976
CPU *Active*	193
Radio *Stand_By*	1.44*e* − 4
Radio *Idle*	0.712
Radio *Power_Up*	0.034175
Radio *Active*	78

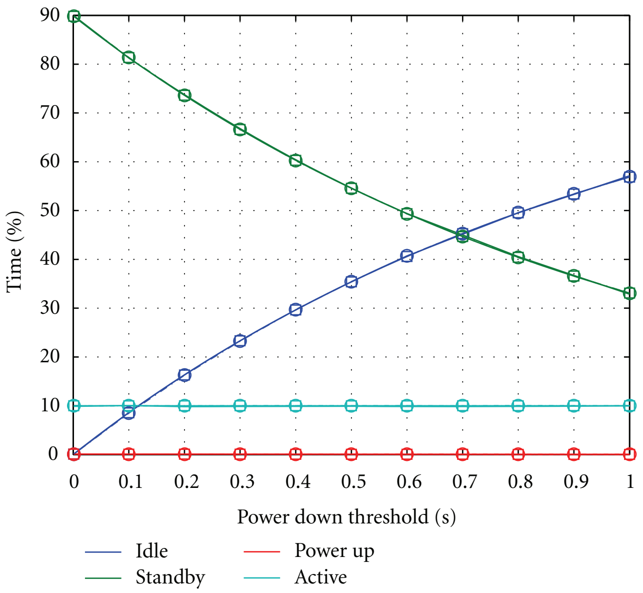


Figure 5: $Power\_Up\_Delay = 0.001$ seconds.

In all of the figures, the simulator results are given by the solid line. The Markov model is represented by the line with squares, and the Petri net by the line with circles.

Figure 8 shows the energy consumption estimates for each of the three methods. It is interesting to note that the average difference between the Markov model energy estimates compared to the simulator is equal to the average difference between the Petri net and the simulator as shown in Table 4.

Figure 6 depicts the behavior of the CPU when the *Power_Up_Delay* is fixed at 0.3 seconds. Although, the Petri net model seems to overestimate the percentages of each of the four states as compared to the simulator, it tends to be a better indicator of the system than the Markov model. Figure 9 shows the energy consumption estimates for each of the three methods. It is interesting to note that the Petri

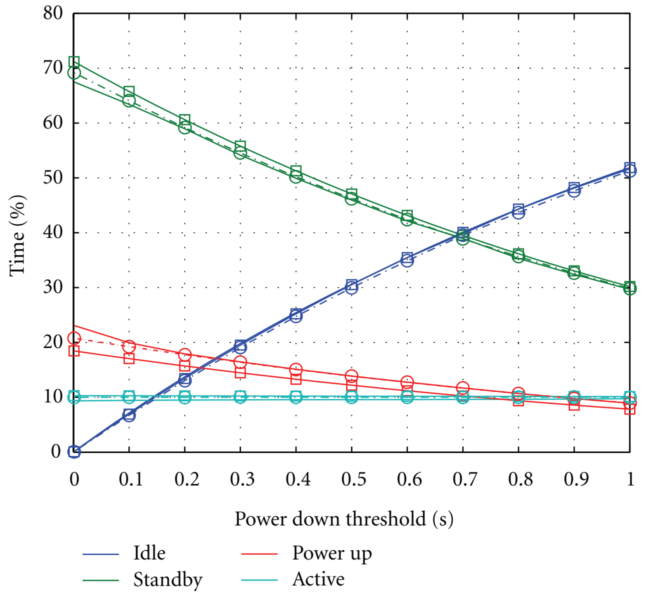


Figure 6: *Power_Up_Delay* = 0.3 seconds.

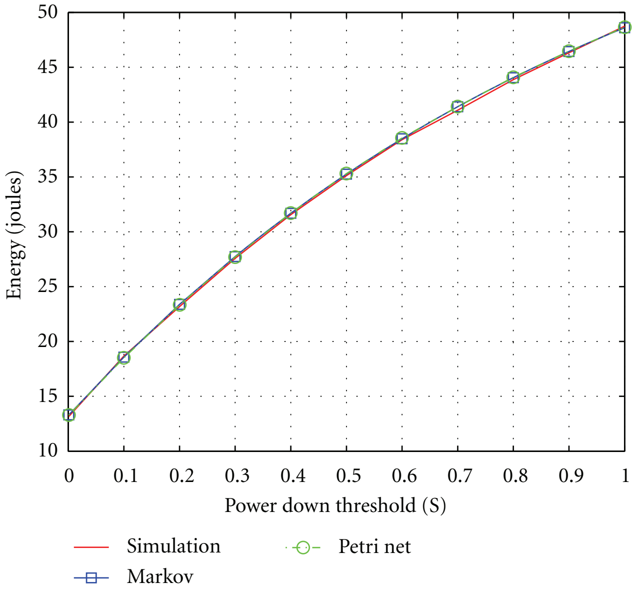


Figure 8: Energy estimates for *Power_Up_Delay* = 0.001 seconds.

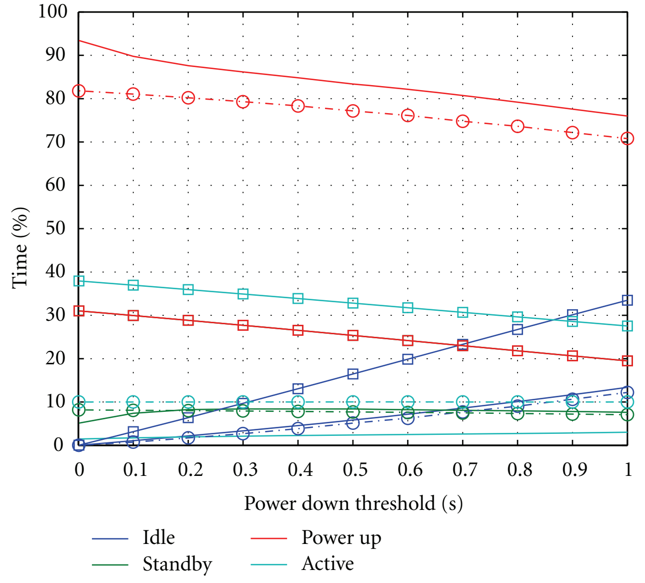


Figure 7: *Power_Up_Delay* = 10 seconds.

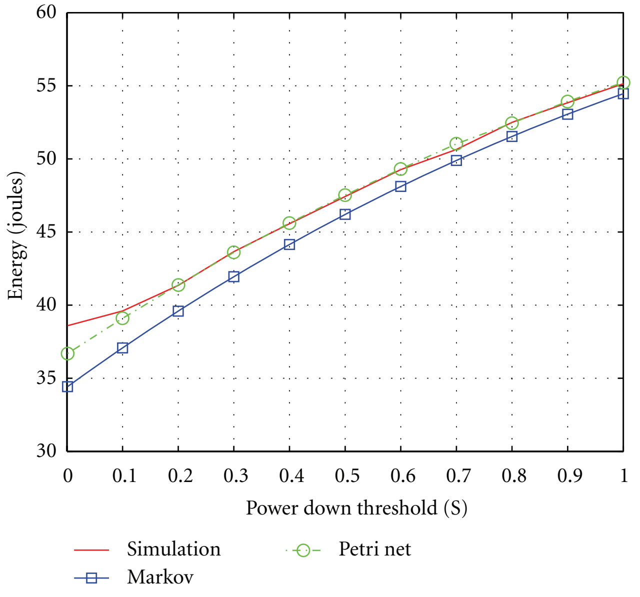


Figure 9: Energy estimates for *Power_Up_Delay* = 0.3 seconds.

Table 4: ΔEnergy (Joules) estimates (*Power_Up_Delay* = 0.001 second).

Power down	ΔSim-Markov	ΔSim-Petri net	ΔMarkov-Petri net
Avg.	7.37	7.37	0.05
Variance	11.88	12.18	0.00
STD DEV	3.45	3.49	0.03
RMSE	8.07	8.08	0.06

net energy estimates are now closer to the simulator results than the Markov model. As Table 5 shows, the average energy estimate difference for all estimates between the simulator and the Markov model is 7.28 Joules, while the difference between the simulator and the Petri net is only 4.99 Joules.

Figure 7 depicts the behavior of the CPU of an extreme case when the *Power_Up_Delay* is fixed at 10 seconds. In this scenario, the CPU spends a significant amount of time in *Power_Up* as it "wakes up." In this setting, the Markov model completely fails to estimate the behavior of the simulator. The Petri net on the other hand seems to be in lock step with the simulator results. The energy consumption comparison in Figure 10 and Table 6 further shows that the Petri net model is more accurate than the Markov model.

By comparing three different scenarios (*Power_Up_Delay* of 0.001, 0.4, and 10 seconds) and of which two were extreme

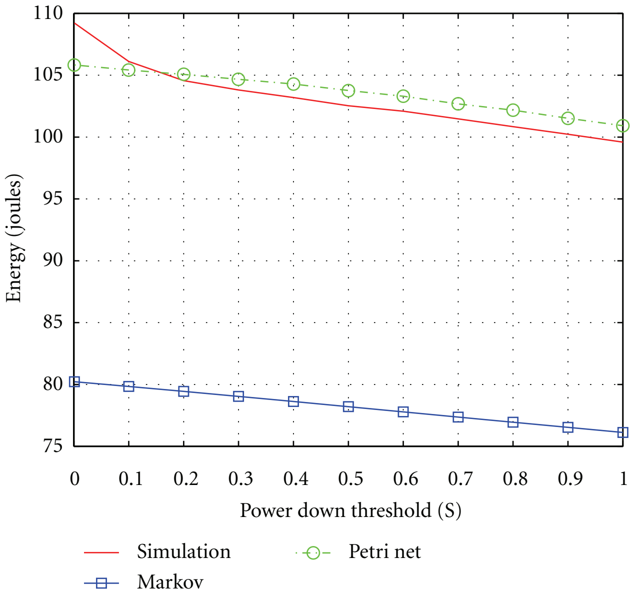


Figure 10: Energy estimates for *Power_Up_Delay* = 10 seconds.

Table 5: ΔEnergy (Joules) estimates (*Power_Up_Delay* = 0.3 second).

Power down	ΔSim-markov	ΔSim-petri net	ΔMarkov-petri net
Avg.	7.28	4.99	2.29
Variance	6.71	3.55	0.51
STD DEV	2.59	1.88	0.71
RMSE	7.69	5.30	2.39

cases (0.001, and 10 seconds), we were able to show that the Petri net is better adept at predicting the behavior of the simulator than the Markov model. The Petri net hence is a better method of modeling a CPU.

Another interesting observation from these experiments is that a *Power_Up_Delay* of 10 seconds results in an energy consumption trend that actually decreases as the *Power_Down_Threshold* increases (see Figure 10). This is because the *Power_Up* power rate is much higher than the *Idle* rate. As the *Power_Down_Threshold* increases, the time spent in *Idle* also increases, and hence decreases the number of time the CPU goes to the *Stand_By* state. As a result, the number of power up transitions decreases, leading to reduced energy usage. From this we can gather that it is more efficient to allow a CPU to idle than to have it repeatedly move from a power down state to active.

5. Evaluation of a Simple Sensor System

In this section, the energy prediction of a Petri net model for a simple sensor system will be compared against real measurements collected from an IMote2 node acting as a sensor node. Figure 11 depicts the generic operating behaviour of a sensor system node. The system remains in a wait state until a random event occurs at which point, a message is received, some computation is required, and then the results are transmitted to some other node. The transition *Job_Arrival* is the only one that fires randomly using an exponential distribution; all others are deterministic transitions. The transition *Temp* and place *Temp_Place* are required in the Petri net to account for the fact that the IMote2 node is not capable of handling events that are less than 1 second apart; the *Temp* transition fires after a fixed 1 second.

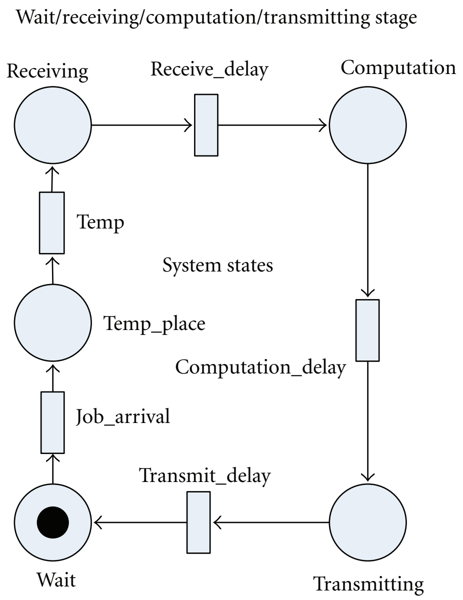


Figure 11: Simple system model of node in wireless sensor network.

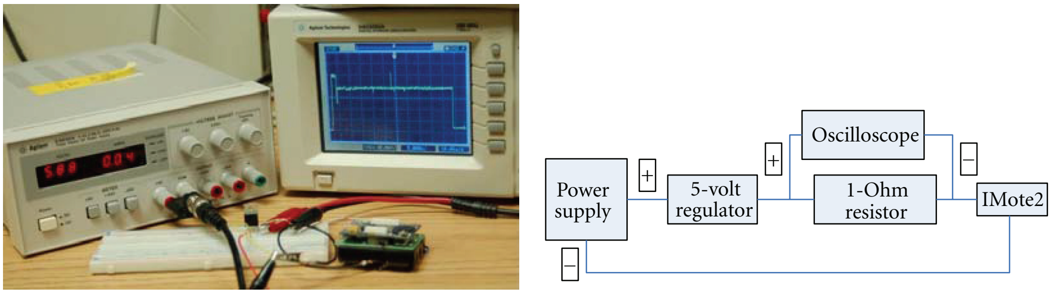
Table 6: ΔEnergy (Joules) estimates (*Power_Up_Delay* = 10 seconds).

Power down	ΔSim-markov	ΔSim-petri net	ΔMarkov-petri net
Avg.	42.41	0.12	42.41
Variance	1.85	0.00	2.00
STD DEV	1.36	0.06	1.41
RMSE	42.43	0.13	42.43

Table 7: Measured power requirements for different IMote2 states.

State	Mean power (mW)
Idle	1.216
Receiving	1.213
Computation	1.253
Transmission	1.028

Figure 12(b) depicts how a power supply was used to power the IMote2 node. The voltage across a 1.16 Ohm resistor was monitored to determine the current draw of the system. The power consumed by the IMote2 as seen at the battery terminals was measured in four different states of operation: computation, idle, transmission, and receiving. The measured power values listed in Table 7 are the average power consumed in these states. It is interesting to note that the transmission state has the least power consumption, even than that of the Idle case. Although this might seem counterintuitive, it must be borne in mind that while



(a) Setup of data collection for IMote2

(b) Block diagram of data collection for IMote2

FIGURE 12: IMote2 power collection setup.

TABLE 8: Petri net transition parameters for a simple system.

Transition	Firing distribution	Delay (sec)	Steady state probability (%)
Job_Arrival	Exponential	3.0	59.8
Temp	Deterministic	1.0	19.7
Receive_Delay	Deterministic	0.00597	0.098
Computation_Delay	Deterministic	1.0274	20.2
Transmit_Delay	Deterministic	0.0059	19.7

TABLE 9: Steady-state probabilities for a simple system.

State/place	Probability (%)
Wait	59.8
Temp place	19.7
Receiving	0.098
Computation	20.2
Transmitting	19.7

TABLE 10: Results of actual system and petri net.

IMote2 execution time	266.5 sec
Average IMote2 power	1.261 mW
IMote2 energy usage	0.336137 J
Petri net energy usage	0.326519 J
Percent difference	2.95

the IMote2 is idling, the receiver is actively "listening" (although it is not receiving anything). The datasheet for the CC2420 radio chip on the IMote2 lists the receive current consumption at 18.8 mA, whereas for transmission, the current draw is 17.4 mA. To obtain the power consumption in the nonidle states, the IMote2 node executed programs that either ran a sort routine repeatedly, transmitted packets, or received packets. This data was collected for the time it took to send or receive 50 packets.

Once the power parameters of the IMote2 were characterized, the energy consumption of the IMote2 as a node in a sensor network was found. This was done by triggering the node randomly for 100 events while the power consumption was monitored. These 100 events took 266.5 Seconds, and resulted in an average power consumption of 1.261 mW. The energy consumption of the IMote2 was found to be 0.336137 J as listed in Table 10.

Using the power parameters collected, the Petri net was simulated until steady-state probability values were obtained. This took about 10 minutes on a 2.80 GHz computer running XP. Table 8 lists the transitions in the Petri net and the delays. Table 9 lists the steady-state probabilities of the places for the given transition parameters in Table 8. Equation (21) was used to compute the energy consumption resulting from these probabilities. As Table 10 indicates, the actual energy consumed by the IMote2 and the energy predicted by the Petri net vary only by about 3 percent.

$$
\begin{aligned}
\text{Total Energy} = \Big(&P_{Wait} \times \left(p_{Wait} + p_{Temp\_Place}\right) \\
&+ P_{Receiving} \times p_{Receiving} \\
&+ P_{Computation} \times p_{Computation} \\
&+ P_{Transmitting} \times p_{Transmitting}\Big) \times \text{Time}.
\end{aligned} \tag{21}
$$

6. Modeling a Sensor Node in Wireless Sensor Networks

In this section, stochastic colored Petri nets are used to model the energy consumption of a sensor node in a wireless sensor network using open and closed workload generators as shown in Figures 13 and 14. Generally, the behavior of nodes in a wireless sensor network follows the same basic pattern. First, a node in *Idle* or *Stand_By* is "awoken" by either an external event or a message from another node. This node then proceeds to process the event or message that typically involves some computation. The resulting information is then transmitted to other sensor nodes or a centralized data collector. Finally, the node moves either to *Idle* or *Stand_By* if no more events arrive for some time period. It then "waits" for another event.

Unlike Markov models, the ease with which a Petri net can be designed allows for complicated scenarios to be modeled. Figures 13 and 14 describe Petri net models of a processor capable of servicing multiple tasks. A colored Petri net is capable of assigning numerical values or other attributes

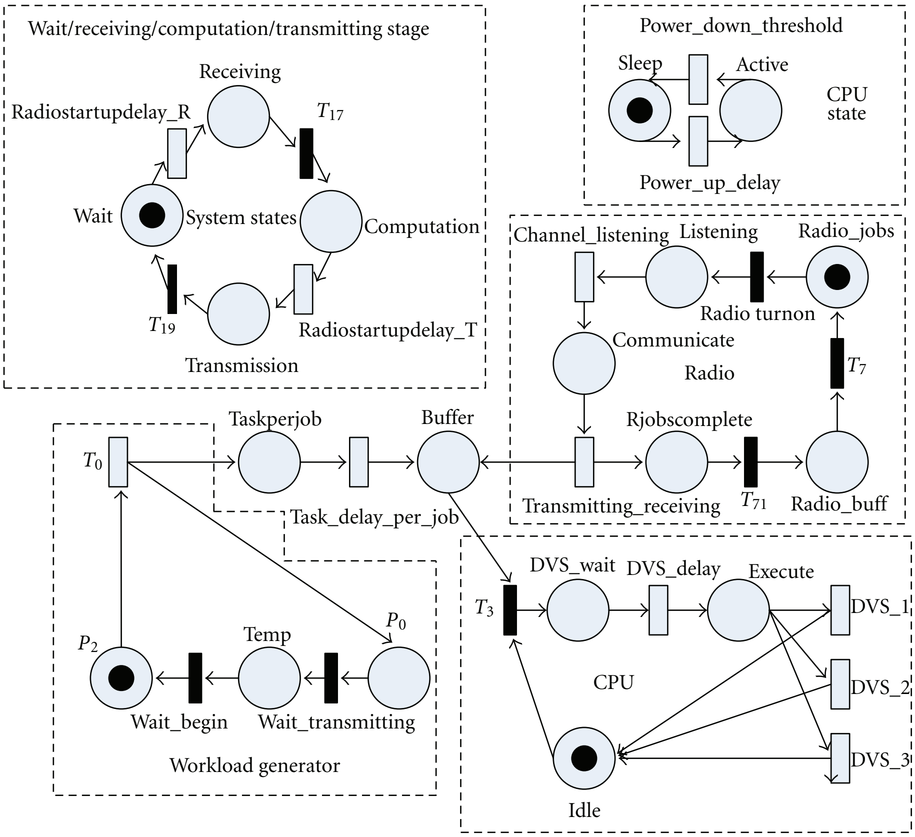


Figure 13: Closed system model of node in wireless sensor network.

to tokens to allow for enhanced decision making capabilities. The characteristics of a token can be checked, using expressions called "local guards," as the token is input to a transition. Local guards can be used to allow or deny tokens from activating a transition. For example, transitions *DVS_1*, *DVS_2*, and *DVS_3* have local guards associated with them and the appropriate transition fires only if the token in its input place (*Execute*) has a corresponding value of 1, 2, or 3 associated with it. This feature allows the model to simulate a DVS processor using a practical variable voltage system where the processor stops executing while changing operating parameters [5]. Tokens of different values result in different execution speeds simulating the change in the operating parameters.

The Petri nets in this section model a system that services jobs of a single type. As soon as a job is generated, a token is placed in either place *P*0 for the closed workload generator (Figure 13) or place *Event_Arrival* for the open workload generator (Figure 14). We use the processor and radio parameters as given in Table 3 [13] for the iMote2 sensor platform to provide realistic analysis.

6.1. Energy Model Using a Closed Workload Generator. Figure 13 demonstrates a stochastic colored Petri net (SCPN) [22] model of a sensor node using a closed workload generator. The portion of the Petri net labeled "*Workload Generator*" generates the job events, while the portions marked "*Radio*" and "*CPU*" refer to those respective components. The system is composed of four states: "*Wait*," "*Receiving*," "*Computation*," and "*Transmitting*." There are two states associated with the CPU: "*Sleep*" and "*Active*."

In addition, our model implemented based on TimeNET utilize a feature called "global guards" to specify more "global" conditions for the firing of transitions. We use global guards in the forms of expressions at the transitions that remove the need to provide connections using arcs. For example, these conditions can be used to check for the number of tokens in a given place. This simplifies the drawing of the Petri net significantly.

Simulation of the open workload Petri net given in Figure 13 results in the movement of tokens as given below.

Global guards for the Petri net in Figure 13 are given in Table 11.

The Petri nets in this section model a system that services jobs of a single type. As soon as a job is generated, a token is placed in either place *P*0 for the closed workload generator (Figure 13) or place *Event_Arrival* for the open workload generator (Figure 14).

The system then moves from the "*Wait*" state to the "*Receiving*" state using transition *RadioStartUpDelay_R* which simulates the time for the radio to start up. Once the system is in the *Receiving* state, this allows the token in *Radio_Jobs* (Radio) to move to *Listen*. The Petri net begins to simulate "*Channel_Listening*" for an available communication slot. Thereafter, the radio proceeds to "receive" information in the *Communicate* place, and after which, a token is

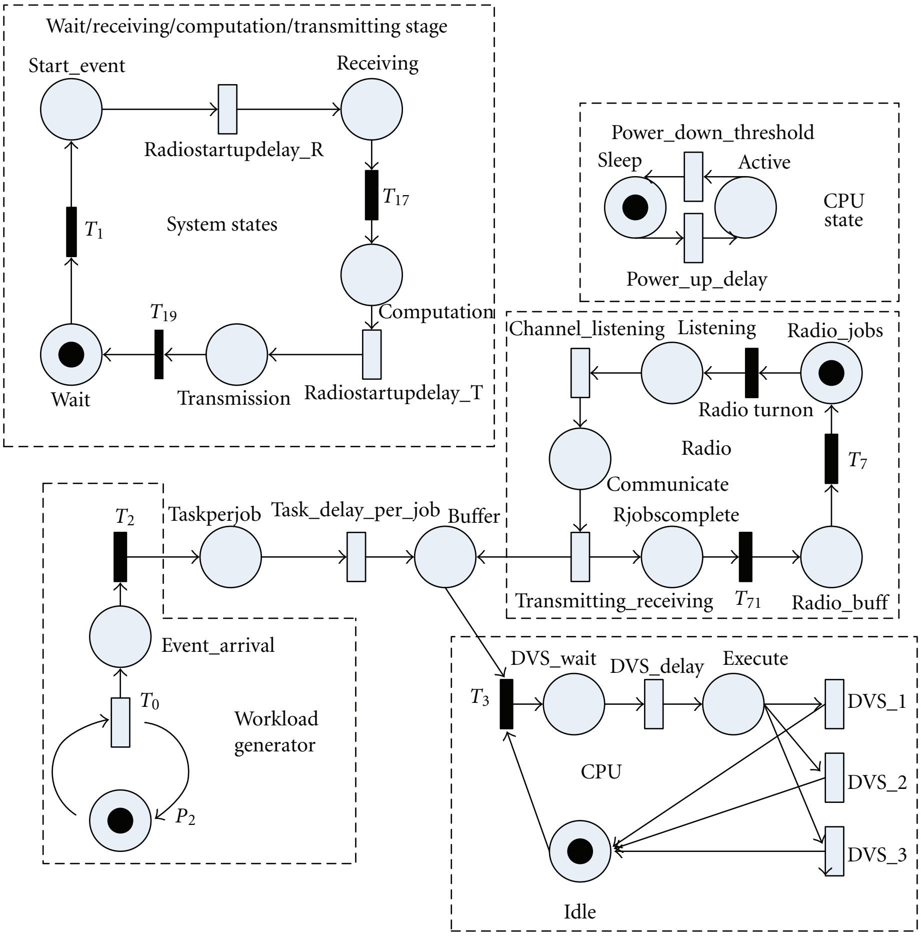


Figure 14: Open system model of node in wireless sensor network.

deposited in the CPU *Buffer* for the purpose of awakening the CPU for error checking the received packet. Any tokens (jobs) in the *Buffer* cause the CPU to transition from the *Sleep* state to the *Active* state.

If the CPU is *Idle*, the token moves from the place *Buffer* to *DVS_Wait*. The transition labeled *DVS_Delay* simulates the time for any overhead to execute a particular task. Finally, the token moves to the *Execute* place to simulate the execution of the job on the CPU. Depending on the value of the token, either *DVS_1*, *DVS_2*, or *DVS_3* is enabled. Once enabled, the transitions will fire after fixed intervals that represent the time to service the appropriate task. Thereafter, the CPU then moves to *Idle*.

Once the CPU has "processed" the received packet, the radio moves to an "idle" mode. The system then moves to the *Computation* state. The token that was generated and placed in *TaskPerJob* is moved to the *Buffer* and the CPU proceeds to service the job simulating any computation required for the event generated. The system then moves to the *Transmitting* state using transition *RadioStartUpDelay_T* to awaken the radio, where the processed information is "transmitted" to some base station using the same steps as for the *Receiving* state. Finally, the state moves back to the *Wait* state. The CPU *Sleep*/*Active* states operate independently of the system states. The CPU is "woken" from sleep if any tokens are placed in the *Buffer*; however, depending upon the CPU *PowerDownThreshold*, the CPU may go back to "sleep" during the communication stage. In which case, the CPU may need to be woken up again.

The Petri net assumes that the radio is put to sleep after the *Transmitting* state. However, between the *Receiving* and *Computation* states the radio is idle. The Petri net also assumes that the radio wake up cost is the same whether the radio is awoken from sleep to active or idle to active; *RadioStartUpDelay_R* = *RadioStartUpDelay_T* = *RadioStartUpDelay*. We will present the simulation results and analysis in Section 7.

6.2. Energy Model Using an Open Workload Generator. Figure 14 demonstrates a stochastic colored Petri net (SCPN) [22] model of a sensor node using an open workload generator. This Petri net is very similar to the one with the closed workload generator presented previously. Many of the transitions and global guards given in Table 11 are also used here. The three additional transitions unique to this model are given in Table 12.

The main difference between the close model (see Figure 13) and the open model (see Figure 14) is that events arrive independently of the state of the system in the Petri net given in the open model. When transition *T*0 fires randomly

Table 11: Closed system model Petri net transition parameters.

Transition	Type	Delay	Global guard
*T*0	DET	*AR*	(#*Wait* > 0)
RadioStartUpDelay_R	DET	0.000194	(#*P*0 > 0)
RadioTurnOn	INST (3)	—	((#*Receiving* > 0) ‖ (#*Transmitting* > 0))
Channel_Listening	DET	0.001	((#*Receiving* > 0) ‖ (#*Transmitting* > 0))
Transmitting_Receiving	DET	0.000576	NA
Power_Up_Delay	DET	0.253	(#*Buffer* > 0)
*T*3	INST (2)	—	(#*Active* > 0)
DVS_Delay	DET	0.05	NA
DVS_1	DET	0.03	*dvs*1 == 1.0 (Local Guard)
DVS_2	DET	0.01	*dvs*2 == 2.0 (Local Guard)
DVS_3	DET	0.081578	*Comm* == 3.0 (Local Guard)
*T*17	INST (3)	—	((#*Buffer* == 0) && (#*Idle* > 0) && (#*RJobsComplete* == *ComPackets*))
*T*71	INST (2)	—	((#*Buffer* == 0) && (#*Idle* > 0) && (#*RJobsComplete* == *ComPackets*))
*T*7	INST (1)	—	((#*Computation* > 0) ‖ (#*Wait* > 0))
Task_Delay_Per_Job	DET	0.000001	#*Computation* > 0
RadioStartUpDelay_T	DET	0.000194	((#*TaskPerJob* == 0) && (#*Buffer* == 0) &&(#*Idle* > 0))
*T*19	INST (3)	—	((#*Buffer* == 0) && (#*Idle* > 0) && (#*RJobsComplete* == *ComPackets*))
Power_Down_Delay	DET	*PDT*	((#*Buffer* == 0) && (#*Idle* > 0))
Wait_Transmitting	INST (3)	—	(#*Transmitting* > 0)
Wait_Begin	INST (3)	—	(#*Wait* > 0)

using an exponential distribution, a token is deposited back in place *P*2 and a new token is placed in place *Event_Arrival*. With a token in place *P*2, transition *T*0 can fire again at any time. In order to assure that multiple but closely spaced events each trigger a new system cycle, place *Start_Event* and transition *T*1 were needed.

As mentioned before, the ease of building Petri nets allows one to simulate complex behavior. The Petri net in Figure 14 describes just one particular scenario. Any variation of other scenarios can just as easily be simulated by slight modifications to the Petri net. This flexibility and ease in modeling a system can go a long way towards obtaining an understanding of the system and hence exploiting power saving features.

Using the Petri net in Figure 14, the effects of the *Power_Down_Threshold* of the CPU on the system can easily be studied. The next section explores results obtained from simulating the Petri net.

7. Energy Evaluation of a Sensor Node

Based on the Petri net models presented in the previous section, this section evaluates the energy consumption of a sensor node and discusses the potential applications of our Petri net model. For all experimental results presented in this section, we use our models to estimate the total energy consumption for a time interval of 15 minutes unless specified otherwise.

7.1. Analysis Using Closed Workload. Figure 15 describes the energy characteristics of a wireless sensor node with a closed generator as *Power_Down_Threshold* increases. We aim to use our model to address the question that was posed in Section 1: what is the optimum *Power_Down_Threshold* that yields minimum energy consumption in a wireless sensor network?

In Figure 15, powering down the CPU immediately after the computation does not result in the minimal energy consumption, neither does always keeping the CPU active achieve the optimal energy efficiency. The optimum energy consumption of approximately 2432 Joules occurs at a *Power_Down_Threshold* of 0.00177 seconds. This is a 35% decrease in energy consumption that occurs when the CPU is immediately powered down to a low power state. This is also a 29% decrease in energy consumption that occurs when the CPU is never powered down. Interestingly, it is no coincidence that the minimal energy consumption occurs at this point as will be illustrated.

In the closed model, all transitions are deterministic including transition *T*0 for generating jobs as well as transition *Channel_Listening*. Although, the results of the closed model are predictable, the results from this model can be used to identify four classes of energy values due to three boundaries that arise. These three boundaries delineate shifts in energy consumption trends as *Power_Down_Threshold* increases. The boundaries are generated because there are three points in the system where the CPU can power

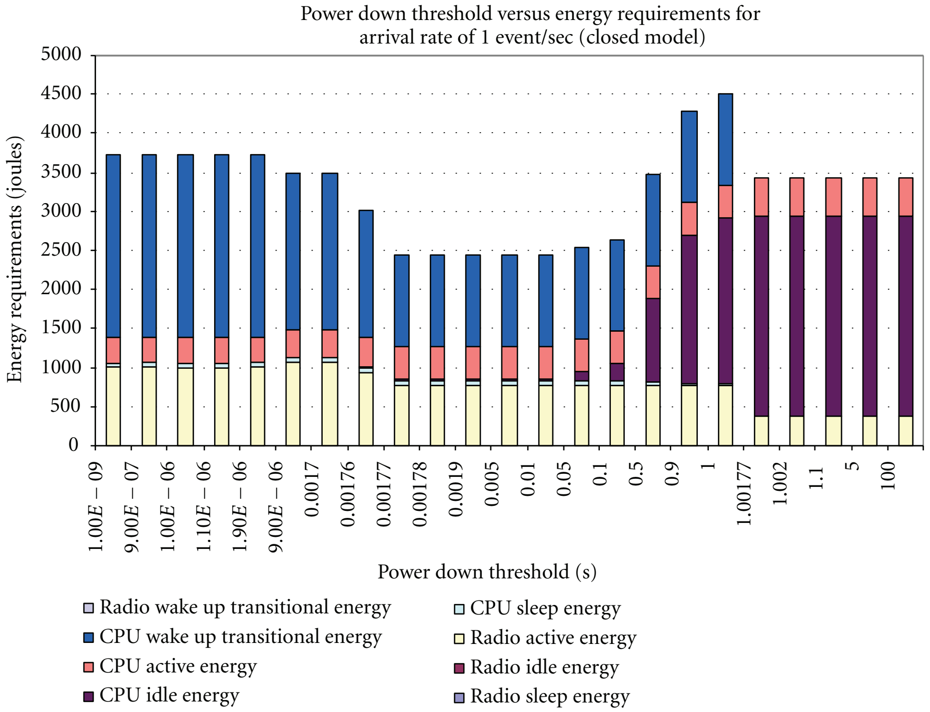


Figure 15: Power down threshold versus energy requirement for a closed model with a job arrival rate of 1 event/second.

down after having been powered up. In other words, the boundaries are the results of three different intervals in the system where the CPU can power down after having been powered up.

The first class of values are those associated when the CPU is powered down after computation at a threshold less than the sum of the least transition times between consecutive CPU usages. Hence, the system is required to power up three times in one system cycle. From Table 11, this happens when *Power_Down_Threshold* < *Tasks_Delay_Per_Job* since this is the transition with the least time between consecutive CPU usages. The second class results when the CPU powers down after some larger threshold and as a result is required to power up twice during the system cycle. In the third class, the CPU powers up once, and in the fourth class the CPU powers up and never powers down again. Each will be examined in detail in the following.

When *Power_Down_Threshold* is smaller than *Tasks_Delay_Per_Job*, that is, it is less than the time between the *Receiving* and *Computation* states, the CPU powers down and is then forced to power up 3 times per system cycle as seen in Figure 16. The red depicts the CPU *Power Up* energy cost, the green depicts the CPU *Active* energy cost, and the aqua depicts the CPU *Idle* energy cost. The purple depicts the radio energy costs.

However, when *Power_Down_Threshold* is larger than *Tasks_Delay_Per_Job*, then the CPU will not power down and will not need to be powered up again between the *Receiving* and *Computation* states. This deterministic transition determines how long the CPU remains idle between the *Receiving* and *Computation* state. At this point, the CPU remains powered up between the Receiving and Computation states, and as a result there is one less CPU power up.

Table 12: Open system model Petri net transition parameters.

Transition	Type	Delay	Global guard
$T1$	INST (2)	NA	$(\#TaskPerJob > 0)$
$T2$	INST (1)	NA	$(\#Wait > 0)$
$T171$	INST (3)	NA	$(\#Wait == 0)$

Since *Power_Down_Threshold* is larger than *Tasks_Delay_Per_Job*, the CPU is powering down fewer times. Hence, the time it takes to power up the CPU is being saved from the cycle time as depicted in Figure 17.

Again, applying the same principal as above, select the next minimum sum of transitions of deterministic delays. From Table 12, this happens for *RadioStartUpDelay* + *Channel_Listening* + *Transmitting_Receiving* or the sum of time when the radio is awoken, a wireless communication slot found, and data is transmitted.

So when

$$\begin{aligned} Power\_Down\_Threshold > {} & RadioStartUpDelay \\ & + Channel\_Listening \\ & + Transmitting\_Receiving, \end{aligned} \tag{22}$$

the CPU now remains powered up between the *Computation* and *Transmitting* states. There is no *Power_Up_Delay* in the *Transmitting* state as shown in Figure 18.

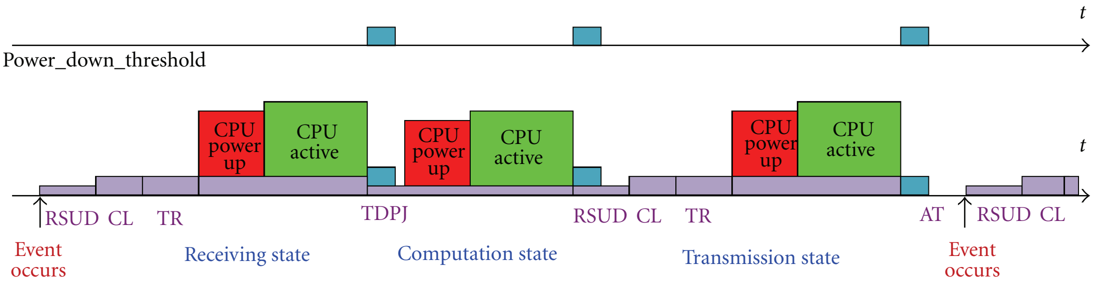


Figure 16: Energy diagram of 3 CPU power ups in one system cycle for closed model (not drawn to scale).

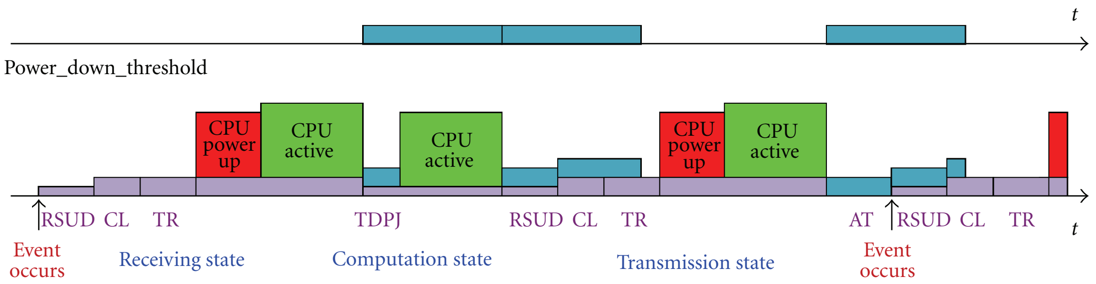


Figure 17: Energy diagram of 2 CPU power ups in one system cycle for closed model (not drawn to scale).

The final set of transitions between two consecutive CPU usage requests are between the end of the current *Transmission* state and the next *Receiving* state. This means that when

$$\begin{aligned} Power\_Down\_Threshold > Job\_Arrival\_Rate & \\ + RadioStartUpDelay & \\ + Channel\_Listening & \\ + Transmitting\_Receiving, & \end{aligned} \quad (23)$$

the CPU now is always powered up. The CPU no longer powers down between the end of the previous *Transmitting* state and the next subsequent event.

The difference between this and the previous cases is that *Power_Up_Delay* is no longer a factor. *Power_Down_Threshold* is sufficiently large that the CPU does not power down after it powers up for the first time between events for this arrival rate as shown in Figure 19.

Each of these cases with the appropriate cycle times is presented in Table 13.

Table 13: *Power_Down_Threshold* cases for a closed system and their associated cycle times.

Power down threshold criteria	Delay (second)	Power up delays	Cycle time (second)
=0	0	3	2.00254
>*TDPJ*	0.000001	2	1.74954
>*RSUD*+*CL*+*TR*	0.00177	1	1.49654
>*AT*+*RSUD*+*CL*+*TR*	1.00177	0	1.24354

As Table 13 indicates, a sudden shift in energy consumption can be expected when

$$\begin{aligned} Power\_Down\_Threshold > RadioStartUpDelay & \\ + Channel\_Listening & \\ + Transmitting, & \end{aligned} \quad (24)$$

that is, *Power_Down_Threshold* > 0.00177 seconds. This can be verified in Figure 15. When *Power_Down_Threshold* is 0.00176 second, the system energy consumption is 3007.827 Joules. However, when *Power_Down_Threshold* is set to 0.00177 second, the system energy consumption is 2431.95 Joules. This is a drop of 19.15% in energy consumption just by increasing *Power_Down_Threshold* by 0.00001 seconds. This decrease in energy consumption results from the fact that the CPU is powering up only once per cycle as opposed to twice. The CPU is saving the heavy *Power Up* energy penalty.

However, as *Power_Down_Threshold* continues to increase, the system energy consumption increases steadily. Upon closer inspection, the CPU *Power Up* energy remains constant; however, it is the CPU *Idle* energy that contributes to the overall increase in the system consumption. As *Power_Down_Threshold* increases, the one *Power Up* penalty is being avoided; however, the time the CPU is in the *Idle* state is being increased and hence the greater system energy consumption is.

This increase continues and even results in a maximum energy consumption of 4501.96 Joules at a *Power_Down_Threshold* of 1 second and then drops suddenly to 3429.92 Joules when *Power_Down_Threshold* increases by 0.00177 seconds. This abrupt change is, of course, due to the CPU

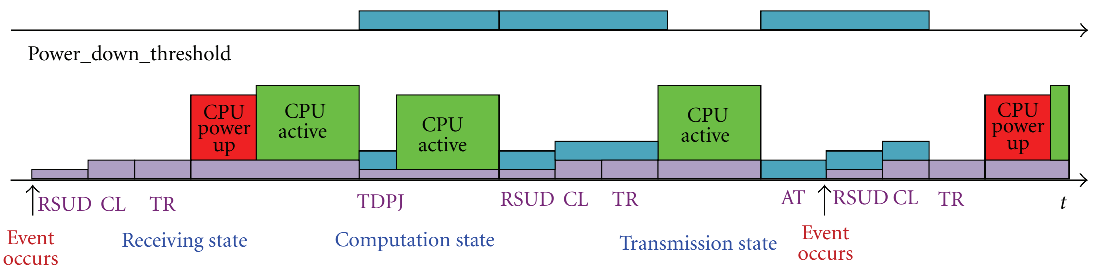


Figure 18: Energy diagram of 1 CPU power up in one system cycle for closed model (not drawn to scale).

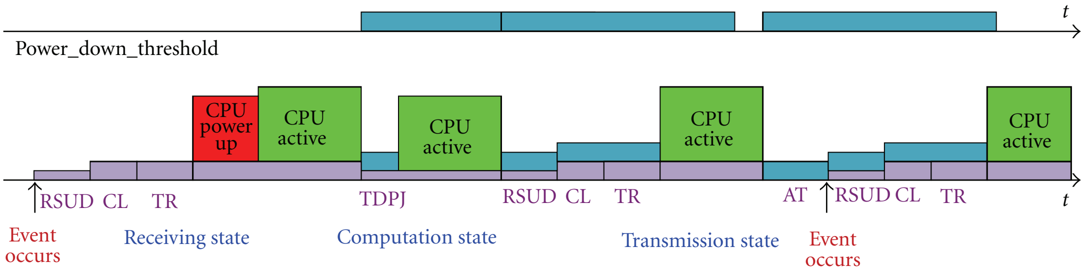


Figure 19: Energy diagram of no CPU power ups (after the first one) in one system cycle for closed model (not drawn to scale).

remaining always on as opposed to powering down once per cycle as indicated in Table 13. Although the CPU *Idle* energy increased during this transition, the saving from the heavy powering up penalty is sufficient to result in the system energy consumption that is less than when the CPU was powering down immediately after it finished computation or when *Power_Down_Threshold* was zero.

We can also see how the tradeoff between the CPU *Power Up* and *Idle* energy consumption of only one power up delay results in the minimum energy consumption at a *Power_Down_Threshold* of 0.00177 seconds. This is when

$$\begin{aligned} Power\_Down\_Threshold > {} & RadioStartUpDelay \\ & + Channel\_Listening \\ & + Transmitting. \end{aligned} \tag{25}$$

We can see from this exercise that powering down the CPU immediately after it completes processing is not the best solution. A careful analysis of the system parameters should be conducted before deciding upon a *Power_Down_Threshold* that will minimize overall energy consumption.

An interesting point to note is that as *Power_Down_Threshold* is steadily increased, the *Radio Active Energy* decreases. The reason for this is apparent because the energy consumption of the *Radio* in the active state is the time spent in the *Receiving* and *Transmitting* states. However, during these states, after communication, the "received" packet needs to be processed by the CPU for integrity checking of the checksum. If the CPU has powered down by then, the *Radio* needs to wait in an active state consuming the active energy rate while the CPU powers up and then processes the packet. This decrease in the *RadioActiveEnergy* as *Power_Down_Threshold* is decreased is indicative of the fact that the CPU is powering down fewer and fewer times during each cycle, and as a result the *Radio* needs to spend less time in an active state while the CPU powers up. When *Power_Down_Threshold* increases to the point where the CPU is always on, the energy consumption of 368.24 Joules is just for the communication involved without the powering up time overhead.

Although the cases where the system CPU moves from powering up twice to only one time

$$\begin{aligned} Power\_Down\_Threshold > {} & RadioStartUpDelay \\ & + Channel\_Listening \\ & + Transmitting\_Receiving \end{aligned} \tag{26}$$

and powering up once to always staying on

$$\begin{aligned} Power\_Down\_Threshold > {} & Arrival\_Time \\ & + RadioStartUpDelay \\ & + Channel\_Listening \\ & + Transmitting\_Receiving \end{aligned} \tag{27}$$

can be clearly identified in Figure 15, it is a little harder to identify the first case where *Power_Down_Threshold* > *Task_Delay_Per_Job* or $1e^{-6}$ Seconds. This is because Time NET uses the float type and since it is a very small value used during simulation, the value of the floating value and the rounding error may have skewed the results of the simulation.

By analyzing the closed model case, an understanding of the internal workings of the Petri net and its resulting

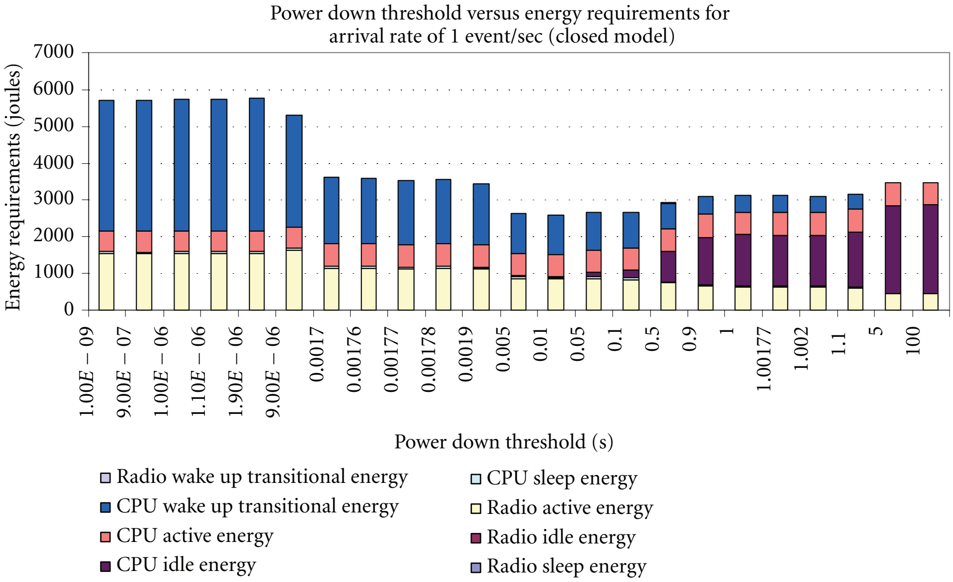


Figure 20: *Power_Down_Threshold* versus energy requirements of an open model for an arrival rate of 1 event/second.

behavior was obtained. This understanding can be applied to the open model as well, however, while acknowledging that the four cases in Table 13 are not as distinct for the open model.

7.2. Analysis Using Open Workload. As Figure 20 shows, having the CPU go to sleep immediately after processing a task is not beneficial for this system either. This can be seen when the *Power_Down_Threshold* is 1.00$e - 9$ seconds or almost zero. The *Power Up* transitional energy to wake the CPU becomes prohibitive. However, it is not beneficial to always keep the CPU "on" either as indicated when *Power_Down_Threshold* is five seconds or more. However, when *Power_Down_Threshold* is approximately 0.01 seconds, the energy requirement is approximately 2589 Joules which is almost 55% less than the energy consumed when the CPU is shut down immediately after every task. This is also less than the energy consumed when the CPU is always on by almost 26%.

Figure 20 is not meant to present the minimum energy level that can possibly be obtained, but this figure does indicate the energy consumption in the four distinct classes. Although Figure 20 is not plotted at even intervals of *Power_Down_Threshold*, the energy trends seem to be relatively smooth.

Except for transitions $T0$ and *Channel_Listening*, all other transitions are deterministic that fire after fixed intervals. An exponential distribution was used for transitions $T0$ and *Channel_Listening* due to the random and sporadic nature of these events. Assuming that the firing time of these exponent transition can be treated as a random variable, this will result in an average firing rate of $(1/ArrivalRate)$. Since the *Channel_Listening* time is so small, it will be used as it is.

In the open model, transitions $T0$ and *Channel_Listening* fire randomly in the specified time interval as given by the exponential distribution. A randomness is introduced that blurs the sharp boundaries in the energy consumption as *Power_Down_Threshold* increases as was seen for the closed model.

Table 14: *Power_Down_Threshold* cases for an open system and their associated cycle times.

Power down threshold criteria	Power up delays	Cycle time (second)
0	3	1.00254
$>TDPJ$	2	0.74954 (1.0)
$>RSUD + CL + TR$	1	0.49654 (1.0)
$>AT + RSUD + CL + TR$	0	0.24354 (1.0)

With open workload, there may not be a delay between the current cycle and the next cycle because the transition that dictates the arrival rate now fires randomly according to an exponential distribution. As Figure 14 and Table 12 indicates after transition $T0$ fires, the token is returned back to place $P2$ again which enables the transition again. $T0$ may fire again immediately after it has just fired or it may fire after waiting the full length of the interval. When $T0$ will fire will be determined by the probabilistic nature of the exponential distribution.

It is possible that when the system finishes servicing the current "event" a new event is waiting for it. Therefore, there is no *Arrival_Time* delay between the current and the next "event." This can happen when the *Arrival_Time* is less than the *Cycle Time* minus the *Arrival_Time* as was expressed before. With an *Arrival_Rate* of 1, event occurs every 1.0 seconds on average $E[X] = 1/\lambda$ where λ is the *Arrival_Rate*. Table 13 can be used and the "Cycle Time" values can be modified by removing the average *Arrival_Rate* from the original cycle times.

The cycle times have been adjusted in Table 14 to show the average cycle times of the system given

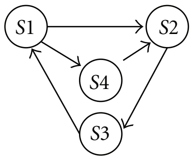


Figure 21: Wireless sensor network.

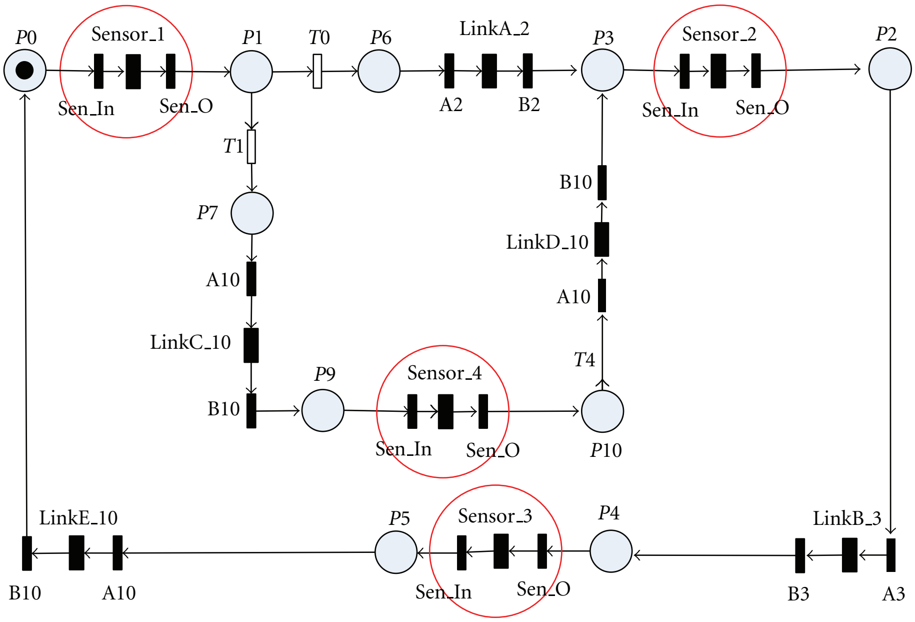


Figure 22: Petri net of wireless sensor network.

Table 15: *Power_Down_Threshold* cases for an open system and their associated down times.

Power down threshold criteria	Cycle time (second)	Down time (second)
0	1.00254	0
$>TDPJ$	0.74954 (1.0)	0.25046
$>RSUD + CL + TR$	0.49654 (1.0)	0.50346
$>AT + RSUD + CL + TR$	0.24354 (1.0)	0.75646

Power_Down_Threshold. However, it should be noted, that since transition $T0$ fires every 1.0 seconds on average, when *Power_Down_Threshold* is equal to 0, events are being generated at a faster rate than can be serviced as indicated by the cycle time. The system is slightly saturated.

In the next three cases, the cycle time is shorter than the average *Arrival_Rate* implying that the average *Arrival_Rate* dictates how often events are serviced. In the latter three cases, the system completes servicing the previous event and waits for the next event to be generated. The length of the time that the system waits in idle is given in the last column "Down Time" in Table 15. In the closed model of the system, since events always arrive after a fixed interval after the system has completed servicing the previous event, the "Down Time" is always 1.0 seconds. Unlike the closed model results in Figure 15 where the minimum energy consumption at the corresponding *Power_Down_Threshold* is apparent, identification of this point for an open model system can be challenging and can fall at a *Power_Down_Threshold* that has not been simulated. The open model of the system presented in this paper contains only two transitions that use exponentially distributed firing rate, and already their results are unpredictable in terms of arriving at an expression to analytically explain them.

If a larger combination of transitions using deterministic and exponential distribution firing rates are used, the resulting outputs can be expected to be even more difficult to analytically derive. This indicates that Petri nets are a very important tool in modeling and analyzing systems.

8. Modeling Wireless Sensor Networks

The Petri net models constructed for the sensor nodes can also be used to create a network of sensors. TimeNET has the capability of modularizing a Petri net so that a hierarchy of Petri nets can be designed. For example, Figure 21 depicts a sensor network composed of four nodes $S1$, $S2$, $S3$, and $S4$ with communication links as described. Slightly modifying the Petri net shown in Figure 13 and substituting it for each of the nodes the sensor network can be created as shown in Figure 22. The filled-in box labeled *Sensor_1* corresponds to $S1$, and so forth. The box labeled *LinkA_2* corresponds to a Petri net simulating the distance link between $S1$ and $S2$

which also incorporates bit error rate, propagation delay, and channel availability.

Using this platform, the energy consumption of a wider range of areas such as routing, network communications, and even high level sensing applications can be modeled and simulated. With such a detailed model from the physical layer to the network layer, our platform allows for research that require knowledge at all these levels such as energy-aware cross-layer routing. In fact, this platform can be used for almost an endless number of research areas.

9. Conclusion

Using stochastic colored Petri nets, this paper develops a detailed and flexible energy model for a wireless sensor node. The experimental results indicate that this model is more accurate than the one based on Markov models. This is due to the fact that a Markov model requires the modeled systems have memoryless states. A wireless sensor node that relies on time to dynamically change its power state does not satisfy the Markov chain's memoryless requirements. In addition, the Petri net model is much more flexible than the Markov model and can easily accommodate changes.

Further, in this paper, we have successfully demonstrated using our model that immediately powering down a CPU after every computation is not an energy minimal option and nor is never powering down the CPU. However, using our model, it is possible to identify a *Power_Down_Threshold* that results in large power savings. Through this example, we were able to show that our model is very useful and provides a valuable platform for energy optimization in wireless sensor networks.

The drawbacks of our Petri net model is that simulating Petri nets can be computationally intensive and require relatively long simulation time to achieve steady-state probabilities. The models presented in this paper required between 10 and 15 minutes of simulation on a 2.8 GHz computer running Windows XP to stabilize. In comparison, evaluation of closed-form Markov equations is almost instantaneous.

Acknowledgments

This work was supported in part by NSF Grants (GK-12 no. 0538457, IGERT no. 0504494, CNS no. 1117032, EAR no. 1027809, IIS no. 091663, CCF no. 0937988, CCF no. 0737583, CCF no. 0621493).

References

[1] A. Seyedi and B. Sikdar, "Modeling and analysis of energy harvesting nodes in wireless sensor networks," in *Proceedings of the 46th Annual Allerton Conference on Communication, Control, and Computing*, pp. 67–71, September 2008.

[2] J. Liu and P. H. Chou, "Idle energy minimization by mode sequence optimization," *ACM Transactions on Design Automation of Electronic Systems*, vol. 12, no. 4, article 38, 2007.

[3] T. Burd, T. Pering, A. Stratakos, and R. Brodersen, "A dynamic voltage scaled microprocessor system," in *Proceedings of the IEEE International Solid-State Circuits Conference 47th Annual (ISSCC '00)*, pp. 294–295, February 2000.

[4] T. Pering, T. Burd, and R. Brodersen, "Simulation and evaluation of dynamic voltage scaling algorithms," in *Proceedings of the 1998 International Symposium on Low Power Electronics and Design*, pp. 76–81, August 1998.

[5] G. Qu, "What is the limit of energy saving by dynamic voltage scaling?" in *Proceedings of the International Conference on Computer-Aided Design (ICCAD '01)*, pp. 560–563, November 2001.

[6] T. D. Burd, T. A. Pering, A. J. Stratakos, and R. W. Brodersen, "Dynamic voltage scaled microprocessor system," *IEEE Journal of Solid-State Circuits*, vol. 35, no. 11, pp. 1571–1580, 2000.

[7] A. Shareef and Y. Zhu, "Energy modeling of processors in wireless sensor networks based on petri nets," in *Proceedings of the 37th International Conference on Parallel Processing Workshops (ICPP '08)*, pp. 129–134, Portland, Ore, USA, September 2008.

[8] A. Shareef and Y. Zhu, "Energy modeling of wireless sensor nodes based on Petri nets," in *Proceedings of the 39th International Conference on Parallel Processing (ICPP '10)*, pp. 101–110, San Diego, Calif, USA, September 2010.

[9] A. Zimmermann, M. Knoke, A. Huck, and G. Hommel, "Towards version 4.0 of TimeNET," in *Proceedings of the 13th GI/ITG Conference on Measurement, Modeling, and Evaluation of Computer and Communication Systems (MMB '06)*, pp. 477–480, 2006.

[10] P. Hu, Z. Zhou, Q. Liu, and F. Li, "The HMM-based modeling for the energy level prediction in wireless sensor networks," in *Proceedings of the 2007 2nd IEEE Conference on Industrial Electronics and Applications (ICIEA '07)*, pp. 2253–2258, May 2007.

[11] Y. Zhang and W. Li, "An energy-based stochastic model for wireless sensor networks," in *Proceedings of the Wireless Sensor Network*, 2011.

[12] Y. Wang, M. C. Vuran, and S. Goddard, "Stochastic analysis of energy consumption in wireless sensor networks," in *Proceedings of the 7th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '10)*, Boston, Mass, USA, June 2010.

[13] D. Jung, T. Teixeira, A. Barton-Sweeney, and A. Savvides, "Model-based design exploration of wireless sensor node lifetimes," in *Proceedings of the 41rth European Conference on Wireless Sensor Networks (EWSN '07)*, 2007.

[14] S. Coleri, M. Ergen, and T. J. Koo, "Lifetime analysis of a sensor network with hybrid automata modelling," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA '02)*, pp. 98–104, ACM, New York, NY, USA, September 2002.

[15] S. Kellner, M. Pink, D. Meier, and E. O. Blaß, "Towards a realistic energy model for wireless sensor networks," in *Proceedings of the 5th Annual Conference on Wireless on Demand Network Systems and Services (WONS '08)*, pp. 97–100, January 2008.

[16] N. Kamyabpour and D. Hoang, "A task based sensor-centric model for overall energy consumption," in *Proceedings of the 12th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT '11)*, pp. 237–244, 2011.

[17] N. Kamyabpour and D. B. Hoang, "Modeling overall energy consumption in wireless sensor networks," submitted, http://arxiv.org/abs/1112.5800.

[18] M. Korkalainen, M. Sallinen, N. Kärkkäinen, and P. Tukeva, "Survey of wireless sensor networks simulation tools for demanding applications," in *Proceedings of the 5th International Conference on Networking and Services (ICNS '09)*, pp. 102–106, April 2009.

[19] X. Fu, Z. Ma, Z. Yu, and G. Fu, “On wireless sensor networks formal modeling based on petri nets,” in *Proceedings of the 7th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM ’11)*, pp. 1–4, 2011.

[20] D. R. Cox, “The analysis of non-markovian stochastic processes by the inclusion of supplementary variables,” *Proceedings Cambridge Philosophical Society*, vol. 51, no. 3, pp. 433–441, 1955.

[21] R. German, “Transient analysis of deterministic and stochastic petri nets by the method of supplementary variables,” in *Proceedings of the 3rd International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS ’95)*, pp. 394–398, IEEE Computer Society, Washington, DC, USA, 1995.

[22] TimeNET 4.0 A Software Tool for the Performability Evaluation with Stochastic and Colored Petri Nets, User Manual.

Hindawi Publishing Corporation
Journal of Computer Networks and Communications
Volume 2012, Article ID 767920, 8 pages
doi:10.1155/2012/767920

Research Article

Energy Efficiency Adaptation for Multihop Routing in Wireless Sensor Networks

Abdellah Chehri and Hussein T. Mouftah

School of Information Technology and Engineering, University of Ottawa, 800 King Edward Avenue, Ottawa, ON, Canada K1N 6N5

Correspondence should be addressed to Abdellah Chehri, chehri@gel.ulaval.ca

Received 18 June 2012; Accepted 17 September 2012

Academic Editor: Runhua Chen

Wireless sensor networks (WSNs) are considered as a suitable solution for long-time and large-scale outdoor environmental monitoring. However, an important feature that distinguishes the WSNs from traditional monitoring systems is their energy constraints. In fact, these nodes have often a limited and usually nonreplenishable power source. Therefore managing these limited resources is a key challenge. In this paper we use an optimization scheme based on adaptive modulation and power control for a green routing protocol. The optimization mechanism is subject to certain QoS requirements in terms of total end-to-end delay time and bit error rate. The simulation results show that the proposed algorithm can, theoretically, reduce the consumed energy of the sensor nodes almost to half.

1. Introduction

Wireless sensor networks (WSNs) are one of today's most interesting emerging technologies. WSN is made up of a large number of inexpensive devices that are networked via low-power wireless communications [1]. The development of this attractive network has open many doors for a several number of "new and exciting" applications, in which flexibility, easy deployment, and configuration are important properties. Amongst a diverse set of applications where WSN can be used we can quote precision agriculture, environment monitoring, fire detection, smart home, intrusion detection, localization, medicine, and many others.

In addition, the wireless sensor networks are working in ad hoc fashion. The self-configuring, self-healing characteristics make a WSN autonomous wireless network, and therefore allow them a great advantage in a large number of situations.

A large number of WSN applications are composed of stationary nodes. These nodes transmit the collected data at relatively low rates, with a focus to route the data to a central base station.

However, one important feature that distinguishes sensor networks from traditional monitoring systems is their nonreplenishable energy, where the sensor nodes are battery-power limited.

It is commonly accepted that the main design goal in wireless sensor networks is the reduction of energy costs. In fact, energy efficiency is an important point emphasized in the vast majority of relevant publications, since energy reserves are usually considered to be finite.

Energy consumption can be affected by all layers of the network, ranging from physical to application layer. Several energy management strategies have been investigated of a WSN [2–6]. The main works focus on efficient topology design (tier layer networks versus flat network) [2], link adaptation [3], energy-efficient MAC [4], energy-aware routing protocol [5], or suitable power preserving physical layer (by using green modulation and power control) [6].

However, this traditional layer-wise approach leads to independent design of different layers and results in higher margins. Interesting approaches, called cross-layer, have been proposed to improve the whole network performance.

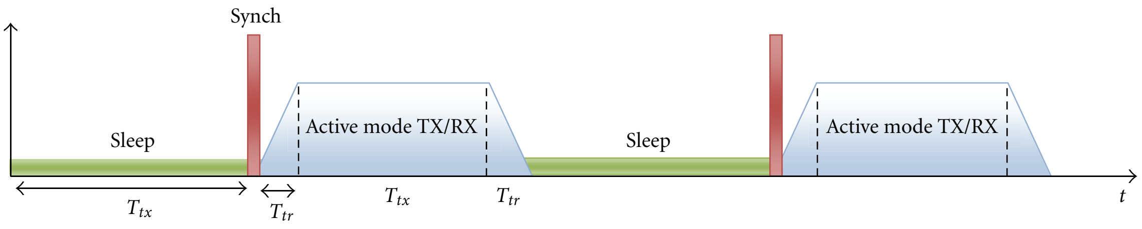


Figure 1: Flexible duty-cycling process.

Su and Lim [7] are the first who proposed a cross-layer design and optimization framework, by providing experimental results and analysis. The research done by the authors shows promising results, and they strongly recommend cross-layer design as a new methodology for designing and optimizing the performance of future wireless networks.

The main idea behind the cross-layer design is the transparency behavior between two or more layers. The interactions and information exchange between various protocol layers can achieve, generally, a good network performance.

In this paper we study design of a joint power and the adaptive modulation scheme applied to a green energy routing protocol. In addition, the proposed scheme used a suitable power-saving MAC protocol with a sleep mode mechanism.

The remainder of this paper is organized as follows. In Section 2, several efforts on energy-related aspects at different layers in WSNs are addressed. The Section 3 deals with the proposed cross-layer assumptions and network configuration. The Section 4 gives the mathematical formulation of the proposed algorithm. Section 5 provides numerical evaluations using the random WSN topology as well as realistic node circuit sensor models to confirm our analysis. We consider first energy-efficient communication over a single communication link between the nodes. Multihop routing will be analyzed using energy efficient routing protocol. Finally Section 6 concludes the paper with some direction for future works.

2. Related Works

Mainly, the sensor energy can be spent in two categories: (1) the device consumption and (2) transmission energy consumption. Even if the power consumption of wireless link represents approximately 70% of the total power in a sensor network, we take into consideration the hardware (or circuit) energy consumption [8]. To optimize the communication power consumption, much work has been conducted either for PHY, MAC, or routing layer. In this section we define the main areas where the sensor energy consumption can be reduced.

2.1. Energy-Aware Physical Layer Protocols. At the physical layer, the modulation and the power transmission are the main adjustment parameters for an energy efficiency communication. One of the pioneering works on energy-efficient modulation has been presented in [9]. The authors focused only on minimizing the energy consumption of transmitting one bit.

The modulation scaling (which exhibits benefits similar to those of voltage scaling) has been also presented. In [10] an adaptive modulation scheme for WSN with an Additive White Gaussian Noise (AWGN) channel has been investigated.

Link adaptation can also achieve a good performance in terms of energy saving in WSN. The results have shown that the link adaptation can improve the performance of such networks [11].

2.2. MAC Protocols. Many energy efficiency MAC protocols have been proposed in literature. One of the primary mechanisms for achieving low-energy operation in energy-constrained WSNs is duty cycling [12].

In these approaches, each sensor node periodically cycles between an active state and a sleep state. Key parameters that characterize the duty cycle include sleep time, wake time, and the energy consumed during the active state and the sleep state. The period of a duty cycle is equivalent to its sleep time plus the active time. On the basis of this reduced duty cycle several protocols have been analyzed (S-MAC, B-MAC, RI-MAC, X-MAC, and T-MAC).

Figure 1 shows the duty cycle mechanism, and the communicating nodes are synchronized with one another.

For this purpose, we use the system model according to a flexible duty-cycling process utilized in practical sensor devices.

Each node is at any time in one of three different states: active (transmitting or receiving), sleeping (not communicating), or transient (powering up from sleep to active).

We denote T_{tr} as the transient mode duration consisting of the switching time from sleep mode to active mode. This model will be adopted in this work.

We should take into consideration that the circuit power consumption is dependent on the modulation order. However, this power is different for each of the three states. When the node is transmitting (i.e., active mode), the total power consumption is found as the sum of the transmission and circuit power consumption.

In addition we note that T_{ac} is a key factor in choosing the energy-efficient modulation, since it directly affects the total energy consumption as we will show later.

2.3. Routing Protocol. Routing problems are usually considered to be the core of wireless sensor network design. However, in most application scenarios, WSN nodes are powered by small batteries, which are often in practice nonrechargeable. Therefore, a good routing protocol should be able to select the routes that consume minimal global energy. Hence, a software solution which combines cross-layer and energy-aware system design to increase energy efficiency is a promising way worth exploring.

A plethora of routing protocols has been developed for WSNs and can be studied in detail in [13]. One reason for such a huge number of protocols for WSNs is that such networks are application specific, and a particular routing protocol can only satisfy a class of WSN application requirements. The shortest path and min-energy algorithm are a typical WSN routing policy.

In [14], the authors have studied network life time maximization of WSN through joint routing and sleep scheduling. The authors tackled the problem by transforming the nonconvex optimization routine into a special form of signomial program (SP). The problem is then solved through an IGP algorithm. Their proposed algorithm drastically outperforms the performance of optimal iterative separate routing and sleep scheduling method by an average of 29% over a large range of traffic rates.

Yanwei et al. proposed an energy-efficient wake-up scheduling for data collection [15]. They have also proposed an efficient method to construct energy-efficient data gathering tree, whose energy cost and time span of the scheduling are both within constant times of the optimum.

In [16], the authors have formulated the joint routing and power control problem in wireless multihop networks as a markov decision process. On the basis of the factorization of the state transition probability, the authors derived a distributed computation method for finding the optimal policy.

3. System Model and Assumptions

Our work has been informed and influenced by a variety of other research efforts [1, 3, 6, 8, 10, 12]. There are various proposals in the area of power management, mostly validated either using theoretical analysis or simulation.

In this work, we extend our previous work [8], to provide, optimize, and analyze a cross-layer energy efficiency model by jointly considering routing (maximum remaining energy routing protocol), MAC, physical layer, and device energy consumption. The works presented in [3, 6, 8, 10] (physical layer), T-MAC (MAC layer), will be combined and applied to a randomly generated multihop route in a network of a given size and density; that is, the nodes in a multihop route will be configured so that the total energy consumption is minimized, subject to QoS requirements.

It is demonstrated that among various sinusoidal carrier-based modulations, the noncoherent M-ary frequency shift keying (NC-MFSK) is the most energy-efficient scheme [6]. Therefore, this modulation scheme was selected in this work.

Our aim is to develop an energy efficiency algorithm for WSN over multi-hop route. The route is selected based on the battery energy levels of each node. Each node uses a modulation scaling and dynamic power transmission. The both parameters are able to vary dynamically depending on the QoS requirements (in terms of maximum end-to-end delay and BER, bit error rate). In order to solve it, we proposed a scenario with the following characteristics.

(i) The underlying wireless sensor network is going to be modeled as multi-hop communication paths. In each path, N sensor nodes will be involved.

(ii) Each node has the possibility of being part of several paths at the same time. Therefore different nodes will have more energy consumption per unit of time than others.

(iii) Each node has several parameters. Most important parameters in our analysis are node remaining energy, modulation order, and transmission power.

(iv) Packet of size Q is going to be transmitted from the sensor to sink with a given end-to-end latency constraint.

(v) The inference among nodes is assumed to be solved by a synchronized T-MAC layer protocol.

(vi) We selected the NC-MFSK modulation scheme. However the research made can be easily extended to other modulation schemes.

4. Problem Formulation

We consider a communication link between adjacent nodes. So that the analysis reflected in the best reality, the theoretical energy consumption models in the transmitter and receiver nodes along the route, for both communication and hardware consumed energy, are taken into account.

4.1. Radio Link Model. The wireless channel model between the sensors is assumed to be Rayleigh flat fading with path-loss, which is a feasible model in static WSNs [6].

We denote the fading channel coefficient correspond to the transmitted symbol i as h_i where the amplitude $|h_i|$ is Rayleigh distributed with probability density function given according to

$$f_{|h_i|} = \frac{2r}{\Omega} \exp\left(\frac{-r^2}{\Omega}\right), \quad r > 0, \tag{1}$$

where

$$\Omega = E\left[|h_i|^2\right]. \tag{2}$$

When transmitting over a ith link with transmission power P_{tx} the received signal power, P_{rx}, is given by

$$P_{rx,i} = \frac{P_{tx,i}}{G_1 d^{\eta} M_l}, \quad \forall i, \tag{3}$$

where d is the separation distance, G_1 is the gain factor at $d = 1\,\text{m}$, which is calculated by

$$G_1 = 10 \log_{10}\left(\frac{(4\pi)^2}{G_t G_r (\lambda)^2}\right). \tag{4}$$

η is the path loss exponent, and M_l the link margin, which are all assumed to be the same for all hops.

The corresponding average signal-to-noise ratio SNR ($\overline{\gamma}$)

$$\overline{\gamma} = \frac{\Omega}{G_1 d^{\eta} M_l} \frac{P_{tx}}{2\sigma^2 B \cdot N_f}, \tag{5}$$

where $P_{tx,i}$ is the energy of the transmitted signal, B is the system bandwidth, σ^2 the power spectral density of the AWGN on the channel, and N_f the receiver noise figure. The measured SNR indicates the quality of a single wireless link.

4.2. Modulation Scheme. Actually the two schemes are playing different and complementary roles while the M-ary quadrature amplitude modulation M-QAM provides better spectral efficiency, and M-ary frequency shift keying the M-FSK assures power efficiency. The Noncoherent M-FSK (NCMFSK) with a small order of constellation size M has been considered the most energy-efficient modulation in WSNs over Rayleigh and Rician fading channels [6].

In addition, using MFSK appears to be the most attractive solution for sensor networks as it is the modulation scheme that requires the least circuit complexity and apart from that increasing M-ary results in small SNR and thus less-required transmitted power needed for a given SER. On the Basis of this criterion, NC M-FSK has been selected in this work.

It is shown that the bit error rate BER, P_b, conditioned upon γ_i of a M-ary FSK is obtained as

$$P_b(\gamma_i) = \int_0^{\infty} u I_0\left(\sqrt{2\gamma_i} u\right)\left[1 - \left(1 - e^{-u^2/2}\right)^{M-1}\right] e^{-(u^2+2\gamma_i)/2} du, \tag{6}$$

where $I_0(x)$ is the zeroth-order modified Bessel function. The average SER of NC-MFSK is given by

$$P_s = \int_0^{\infty} P_s(\gamma_i) f_{\gamma_i} d\gamma_i. \tag{7}$$

It is shown that P_s is upper bounded by

$$P_s \leq 1 - \left(1 - \frac{1}{2 + \overline{\gamma}}\right)^{M-1}. \tag{8}$$

Since it aims to obtain the maximum energy consumption, we approximate the above upper bound as equality:

$$\overline{\gamma}_{\min} \approx \left[\left(1 - (1 - P_s)^{1/(M-1)}\right)^{-1} - 2\right]. \tag{9}$$

Combining (7) and (12) yields the expression for the required transmission power using modulation order M over a link:

$$P_{tx} \approx \left[\left(1 - (1 - P_s)^{1/(M-1)}\right)^{-1} - 2\right] \cdot \frac{G_1 d^{\eta} M_l 2\sigma^2 B \cdot N_f}{\Omega}. \tag{10}$$

4.3. Total Energy Consumption. As we described earlier, the energy consumed of the transmitter nodes is caused by two sources: one part is due to the RF signal generation, which mostly depends on choosing modulation and target distance and therefore on the transmission power P_{tx}. A second part of consumed energy (called P_{hr}) is due to the hardware electronic components (filters, frequency synthesis and amplifiers...).

In this work, we used an energy consumption model for a generic (canonical) packet-based node-to node wireless communication link. Although there are extensive assumptions and validations of these models by measurements in various scenarios, this is beyond the scope of this work.

Since sensor nodes in a typical WSN are densely deployed, the distance between nodes is normally short. Thus, the total circuit power consumption, defined by $P_{\text{hr}} = P_{\text{ct}} + P_{\text{cr}}$, is comparable to the RF transmit power consumption denoted by P_{tx}, where P_{ct} and P_{cr} represent the circuit power consumptions for the transmitting sensor and receiving node, respectively.

To modeling the P_{hr}, we use the similar model described by Abouei et al. [6]. The optimum NC-MFSK demodulator consists of a bank of M-matched filters, each followed by an envelope detector. For the sensor node with the NC-MFSK modulator, we denote the power consumption of frequency synthesizer, filters, and power amplifier as P_{Sys}, P_{Filt} and P_{Amp}, respectively. In this case

$$P_{\text{ct}} = P_{\text{Sy}} + P_{\text{Filt}} + P_{\text{Amp}}. \tag{11}$$

For transmitting circuit's part, we considered that the circuit consumption is depending on the modulation order. In other words, an increase of modulation order yields an increase of power consumption of node circuit. In addition, we assume that the receiving node uses a low-noise amplifier (LNA), which is generally placed at the front-end of an RF receiver circuit, an intermediate-frequency amplifier (IFA), and an ADC, regardless of type of deployed modulation. Thus, denoting P_{LNA}, P_{Filt}, P_{ED}, P_{IFA}, and P_{ADC} denoting power consumption of LNA, filters, envelope detector, IF amplifier, and ADC, respectively, the circuit power consumption of the node is given by

$$P_{\text{cr}} = P_{\text{LNA}} + M \cdot (P_{\text{Filt}} + P_{\text{ED}}) + P_{\text{IFA}} + P_{\text{ADC}}. \tag{12}$$

In addition, it is shown that the energy consumption during transit time (T_{tr}) is obtained as $P_{\text{tr}} = 2P_{\text{syn}}$.

4.3.1. The Peak Power Constraint. The nodes are peak power limited, that is, the instantaneous power that can be drained from the batteries is limited by the constraint P_{Max}. The peak power constraint P_{Max} limits the total power consumption in each node, since $(1 + \alpha)P_{tx} + P_{\text{ct}} \leq P_{\text{Max}}$ is required. For a given modulation order M, the output transmission power is then limited as follows:

$$P_{tx,\max} = \frac{P_{\text{Max}} - P_{\text{ct}}}{(1 + \alpha)} \quad \forall i, \tag{13}$$

where factor $\alpha = 1/\zeta - 1$ depends on the drain efficiency ζ of the RF power amplifier and the peak-to-average power ratio (PAPR) ζ of the transmitted signal as described in [6].

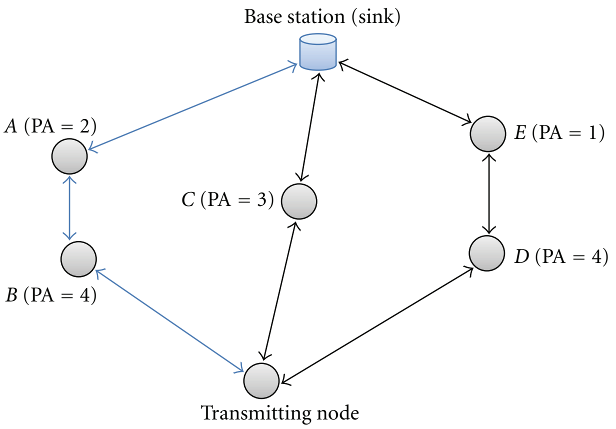


FIGURE 2: Selection for the route based on maximum-minimum available-energy routing protocol.

4.3.2. Active Transmitting Time. The time duration of the transmission of one packet of length Q bits (including all headers) can be modeled as

$$T_{\text{act}} = \frac{QM}{B\log_2 M}, \quad \forall i, \tag{14}$$

where B is channel bandwidth.

4.3.3. Link Energy Consumption. The total energy consumption for a packet transmission over the ith hop can now be expressed as

$$E_i = T_{\text{act},i}\left(P_{\text{hr}} + P_{\text{tx},i}\right) + T_{\text{tr}}P_{\text{tr}}. \tag{15}$$

E_i, T_{act}, P_c, and P_{tx} are all functions of the modulation order M For easy writing, and this dependence will not be indicated explicitly.

4.3.4. Route Selection. To this point, we have simply assumed that a reliable path from source to sink has been found, without discussing how. Let us consider a source node try to forward the collected data to the sink node, and let us assume, without loss of generality, that there are N disjoint routes from source to sink (see Figure 1). Each route R_i, $i = 1, \ldots, X$ contains intermediate nodes n_{ij} $(1, \ldots, N)$. Let us call ξ_{ij}, with $0 \leq \xi_{ij} \leq 1$, the relative residual energy of node n_{ij}. The available energy E_i on route R_i is the lowest energy level of all nodes forming this route, that is, $E_i = \min(\xi_{ij})$. Maximum-minimum available-energy routing selects the route with maximum E_i, that is, route R_k whose k satisfies

$$k = \underset{1\leq i\leq N}{\operatorname{argmax}}\left\{\min_{1\leq i\leq N}\left(\xi_{ij}\right)\right\}. \tag{16}$$

The route that has maximum-minimum available power is preferred. Using (16), the total available energy is calculated by summing the PAs of each node along the route. On the basis of this approach, route (B-A) is selected in Figure 2.

4.3.5. QoS Requirements. The QoS requirements are given per end-to-end packet transmission, in terms of the maximum allowed delay $T_{\max}$ and P_b. The total transmission time is calculated as the sum of the transmission and transit times for all nodes. In addition, there might be a queuing delay at each node, denoted $T_{q,i}$ $i = 1, 2, \ldots, L$. If no other traffic exists on the route, and all processing and buffering delays associated with the forwarding of the packet are ignored.

This assumption is made in [17] and will be adopted here. Although this assumption is not possible to fulfill in an implementable system, it does provide a theoretical upper performance bound.

For an end-to-end delay constraint $T_{\max}$, a valid configuration must satisfy

$$T_i = \sum_{i=1}^{L}\left(T_{\text{act}} + 2T_{\text{tr}}\right) \leq T_{\max} \tag{17}$$

The probability for a successful end-to-end transmission over L hops can be found as the product of the probability for successful transmission for each hop, assuming that bit error events are independent between the hops; that is, a decode-and-forward scheme is used. The following product must be satisfied:

$$\prod_{i=1}^{L}\left(1 - P_i\right) \geq \left(1 - P_{\text{req}}\right). \tag{18}$$

The dedicated time to share key information between nodes such as a routing protocol mechanism, remaining node's energy, and so forth could be transmitted during a fixed time called T_{overhead} (i.e., packet headers). Since this time is constant and not functions of any variables, it can be ignored for purposes of optimization.

4.3.6. Optimization Algorithm. We consider a communication link between adjacent nodes. So that the analysis could reflect the best reality, the theoretical energy consumption models in the transmitter and receiver nodes along the route, for both communication and hardware consumed energy, are taken into account. Consider

$$\begin{aligned}
&\text{Given}: && L, d_{ij}, T_{\max}, P_b \\
&\text{Find}: && p_{tx,\text{opt}}, b_{\text{opt}} \\
&\text{Minimise}: && E_{\text{tot}} = \sum_{i=1}^{L}\frac{QM}{M\log_2 M}\left(P_{\text{hr}} + P_{tx,i}\right) + 2T_{\text{tr}}P_t
\end{aligned} \tag{19}$$

subject to:

$$\sum_{i=1}^{L}\left(T_{\text{act}} + 2T_{\text{tr}}\right) \leq T_{\max}, \tag{20a}$$

$$\gamma_i \geq \gamma_{\min} \quad \forall i \tag{20b}$$

$$P_{tx,i} \leq P_{\max} \quad \forall i \tag{20c}$$

$$\xi_1 > \xi_2 > \xi_3 \cdots > \xi_L, \tag{20d}$$

where d_{ij}, is the distance between the nodes i and j. $T_{\max}$ is the end-to-end delay requirement, as we explain in the next subsection, which is imposed on the whole path as a constraint. P_b is the required BER, and this parameter is also imposed on the whole path as a constraint. L is the route length. ξ_j is the function of the current residual energy of node j.

The first expression gives the total energy consumption for each transmitting packet over all accessible routes (described by (15)). The inequality (20a) ensures that the packet should transmit through the route with minimum time $T_{\max}$, while T_{act} is the individual transmission time of a node. It should be noted that, in some sensor network applications, the latency is very crucial (monitoring, risk, or fire detection). The third expression is the SNR constraint which is proportional to the BER constraint (9). The third one is the SNIR constraint which is proportional to the BER constraint.

The 4th line of (20c) means that each node transmits with an adaptive power which is less than a certain fixed power $P_{\max}$, where $P_{\max}$ is the peak power constraint that limits the total power consumption in each node (given in (13)). The 5th line of (20d) denotes the energy consumption of a node as a function of remaining energy (from rich node to poor node).

The immediate solution for finding the optimal modulation for each node is to study all possibilities and then to determine the optimal modulation order. However, it is very expensive in processing time to find the optimal modulation by the examination of all possibilities. As solution which is more efficient consists of finding the optimal modulation of each node that respect all constraints without processing all possibilities. Therefore, suboptimal solution by using a greedy algorithm has been adopted to resolve the system of (19).

5. Simulations Results

In this paper, we assume that each node is stationary and knows its location via GPS or other localization services. Localization is a basic service essential for many applications that need to know the physical location of sensor readings. We also assume that radios can adjust their transmission power and modulation order. For example, the Chipcon CC1000 radio can vary its transmission power between −20 dbm and 10 dbm. RPAR is designed to work with existing simple CSMA/CA protocols such as the T-MAC protocol in TinyOS.

5.1. Route Selection. We evaluate the performance of the algorithm for dense WSN network. Therefore, we deployed 100 nodes over a flat square area of 100 m × 100 m according to a uniform distribution. In order to represent a realistic multihop WSN network, an exponential distribution of node's remaining energy is used.

Compared to the far-away nodes, the nodes close to the sink use often their energies to relay the data. With the times, these nodes drain overall energy. An exponential distribution of energy consummation of a multi-hop path can represent perfectly this scenario. To alleviate this responsibility, the other nodes should help to balance the expenses of the energy.

Table 1: The summarizes the system parameters for the simulation.

Parameter	Value
$P_{\max}$	100 mW
G_1	30 dB
η	2
M_l	40 dB
α^2	−174 dBm/Hz
N_f	10 dB
B	62.5 Khz
Q	1024 ∗ 8 bit
ζ	0.35
Ω	1
P_{Ed}	3 mW
P_{Syn}	10 mW
P_{fil}	3 mW
P_{ADC}	7 mW
P_{LNA}	9 mW
P_{IFA}	3 mW
t_{tr}	5 μs

For our routing protocol, we use the maximum-minimum available-power scheme (Section 4.3.4). First, the energy was assigned to all nodes (using an exponential distribution). The second step is to find all possible routes between a node and the sink. From these founded routes, we chose the path that respects (16). The selected green route will be selected for our simulation. So, at this step, the data transmitting node have all the information about the number of hops, and the selected intermediate nodes relay the data.

5.2. Case of Single-Hop Communication. First, we present some numerical results to evaluate the energy efficiency for single hop. The system described has been simulated taking as a reference IEEE 802.15.4 standard in order to make a realistic choice of the simulation parameter values. For this purpose, we assume that all modulation schemes operate in B = 65.5 KHz and the f_0 = 2.4 GHz Industrial Scientist and Medical (ISM) band has been utilized for WSNs. According to the Federal Communications Commission (FCC 15.247 RSS-210 standard) for United States/Canada, the maximum allowed antenna gain is 6 dBi [7]. In this work, we assume that Gt = Gr = 5 dBi.

To gain more insight to the above optimization problem, we use a specific numerical example with the simulation parameters summarized in Table 1 (for more details on the simulation parameters, we refer the reader to [6] and its references.). In addition, we assume the end-to-end bit error probability of the route is $P_b = 10^{-3}$ and path loss exponent η equal to 2.

Figure 3 shows the consumed energy over single-hop communication, in terms of energy per bit, as function of modulation order and hop length, for hop lengths 1–50 m,

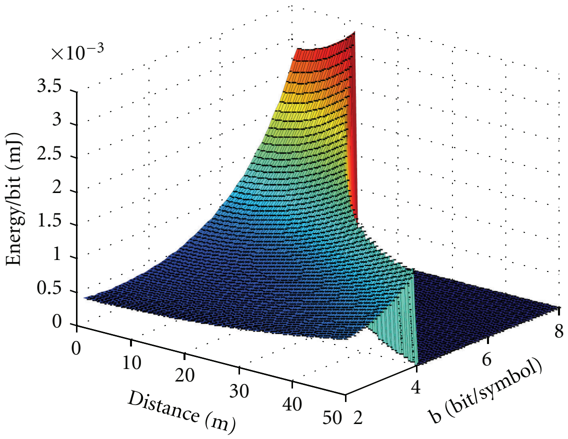


FIGURE 3: Energy E_i for single-hop transmission as function of modulation order b (b/symbol) and hop length.

modulation order 1–8 bits/symbol, and target BER 10^{-3}. The dark-blue area below the sharp threshold marks the region for which no valid configuration exists, due to the peak-power constraint.

From this figure, we can see that when the transmission range increases the required energy to transmit a bit increase, this seems evident. However, if we increase the modulation order this would waste valuable energy for transmission over shorter distances, where a higher modulation is higher energy consumption. This important point will be useful in multihop scenario. Generally speaking, the optimal configuration for each hop is found as the best tradeoff between saving circuit energy at the cost of transmitting energy, and vice versa.

5.3. Case of Multihop Communication. The solution found in the Section 5.1 is encouraging; we analyze the impact of our algorithm in the case of a multihop. We select one of the routing algorithms given in Section 4.3 for the global study.

The problem is to find the optimally modulation order of each node to balance the consumed energy between all nodes across route without violating the end-to end mean latency and packet loss constraints of the application. This should, naturally, take into account the remaining battery level for each node.

The simulation analises end-to-end packet transmissions over multihop route. The folloing three important optimization schemes are investigated:

(i) fixed modulation order (fixed rate) with a fixed transmission power,

(ii) variable rate (variable modulation order) with a fixed transmission power,

(iii) variable rate (variable modulation order) with a variable transmission power.

For basic comparison we first evaluate the impact of route length on the performance of each optimization scheme.

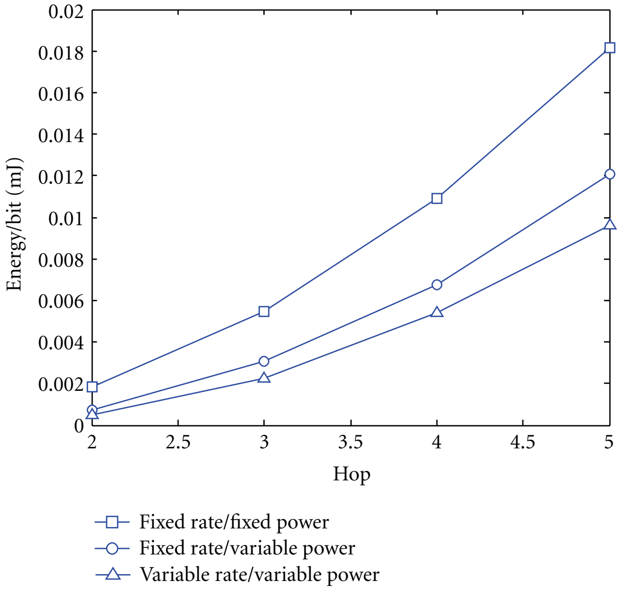


FIGURE 4: Simulation results for multihop communication averaged over 1000 runs.

Averaging the results over a large number of iterations gives a better picture of the system's performance. Hence, we use a Monte Carlo simulation with (the simulation was evaluated for 1000 times) and varying the rough lengths from 2 to 5.

First of all, the simulation results show (Figure 4) that the adaptive schemes can give huge energy savings. As one might expect, transmitting at a fixed rate with maximum power yields the largest energy consumption. For a route length of 4 hops, the results show that, by allowing power control (channel inversion), 37% of energy savings per bit can be obtained over fixed rate and fixed transmission power.

However, the system can achieve up to 49% by using adaptation modulation and power control. The variable rate (variable modulation order) with a variable transmission power adaptation scheme yields 19% of energy savings compared to a fixed rate with power control.

5.4. Impact of Delay Constraint. We have seen that the transmitting at high modulation order requires high consumed energy compared to transmission with a low order of modulation. The problem does not arise if there is not end-to-end delay constraint (the user simply transmits with a low modulation order). So it seems important to see the impact of end-to-end delay on the performance of the system.

Figure 5 shows the results of a number of simulations, where the delay constraint has gradually increased, while all other parameters are kept the same ($P_b = 10^{-3}$, and number of hop = 5). The fixed-rate modulation was fixed to $b = 2$. Form the same figure we note that setup error (where there is not solution or when the constraints are not respected) is much smaller for adaptive modulation than fixed modulation. However, when the time delay constraint

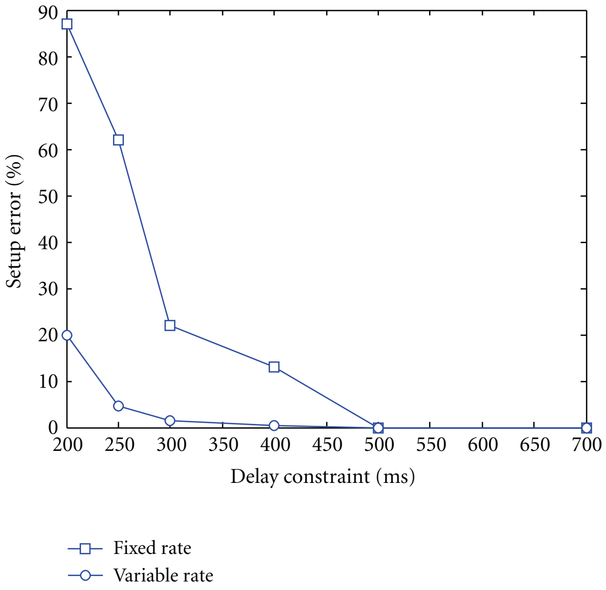


Figure 5: The impact delay constraint on the performance of the system.

increases, the error decreases in both cases. After delay constraint over 500 msec, the error setup becomes zeros.

6. Conclusion

The idea behind the energy-efficient wireless sensor networks is that each node can only transmit to a limited number of other nodes directly. The limited resources of nodes imply that the transmission range is limited. In order to transfer the data to the final destination, the traffic must be relayed using intermediate nodes, creating a multihop route. The total energy consumption associated with an end-to-end transmission over such a route can be significantly reduced if the nodes are correctly configured.

In this work, we investigate on the route optimization problem for wireless sensor networks from the energy management perspective. Our goal is to adapt all routes in the networks to transmit the data from the node to the sink with the best energy performance. This is calculated according to certain quality of service (QoS) in order to provide an energy-efficiency routing strategy.

The simulation shows that we can reduce the consumed energy of a route up to 49% if the node selects the optimal transmission and modulation order.

References

[1] F. Weiwei, L. Feng, Y. Liu, Y. Fangnan, S. Lei, and S. Nishio, "Energy-efficient cooperative communication for data transmission in wireless sensor networks," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 4, pp. 2185–2192, 2010.

[2] T. Kaur and J. Baek, "A strategic deployment and cluster-header selection for wireless sensor networks," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 4, pp. 1890–1897, 2009.

[3] S. Cui, A. J. Goldsmith, and A. Bahai, "Energy-constrained modulation optimization," *IEEE Transactions on Wireless Communications*, vol. 4, no. 5, pp. 2349–2360, 2005.

[4] H. S. Kim and S. W. Lee, "Enhanced novel access control protocol over wireless sensor networks," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 2, pp. 492–498, 2009.

[5] E. Lee, S. Park, F. Yu, and S. H. Kim, "Data gathering mechanism with local sink in geographic routing for wireless sensor networks," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 3, pp. 1433–1441, 2010.

[6] J. Abouei, J. D. Brown, K. N. Plataniotis, and S. Pasupathy, "On the energy efficiency of LT codes in proactive wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 59, no. 3, pp. 1116–1127, 2011.

[7] W. Su and T. L. Lim, "Cross-layer design and optimization for wireless sensor networks," in *Proceedings of the 7th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD '06)*, pp. 278–284, Washington, DC, USA, June 2006.

[8] A. Chehri and H. Mouftah, "Energy-aware multi-hop transmission for sensor networks based on adaptive modulation," in *Proceedings of the 6th Annual IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob '10)*, October 2010.

[9] C. Schurgers, O. Aberborne, and M. Srivastava, "Modulation scaling for energy-aware communication systems," in *International Symposium on Low Power Electronics and Design (ISLPED '01)*, pp. 96–99, Huntington Beach, CA, USA, August 2001.

[10] S. Cui, A. J. Goldsmith, and A. Bahai, "Energy-constrained modulation optimization," *IEEE Transactions on Wireless Communications*, vol. 4, no. 5, pp. 2349–2360, 2005.

[11] C. V. Phan, Y. Park, H. H. Choi, J. Cho, and J. G. Kim, "An energy-efficient transmission strategy for wireless sensor networks," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 2, pp. 597–605, 2010.

[12] S. H. Hong and H. K. Kim, "A multi-hop reservation method for end-to-end latency performance improvement in asynchronous MAC-based wireless sensor networks," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 3, pp. 1214–1220, 2009.

[13] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: a survey," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6–28, 2004.

[14] F. Liu, C. Y. Tsui, and Y. J. Zhang, "Joint routing and sleep scheduling for lifetime maximization of wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 9, no. 7, pp. 2258–2267, 2010.

[15] W. Yanwei, L. Wu, L. Xiang-Yang, L. Yun-Hao, and L. Wei, "Energy-efficient wake-up scheduling for data collection and aggregation," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 2, pp. 275–287, 2010.

[16] L. Zhichu Lin and M. van der Schaar, "Autonomic and distributed joint routing and power control for delay-sensitive applications in multi-hop wireless networks," *IEEE Transactions on Wireless Communications*, vol. 10, no. 1, pp. 102–113, 2011.

[17] H. C. Yang, L. Yang, and K. Wu, "Minimum-energy route configuration for wireless ad hoc networks," in *Proceedings of the 25th IEEE International Performance, Computing, and Communications Conference (IPCCC '06)*, Phoenix, Arizona, USA, April 2006.

Hindawi Publishing Corporation
Journal of Computer Networks and Communications
Volume 2012, Article ID 757291, 10 pages
doi:10.1155/2012/757291

Research Article

Power-Management Techniques for Wireless Sensor Networks and Similar Low-Power Communication Devices Based on Nonrechargeable Batteries

Agnelo Silva,[1] Mingyan Liu,[2] and Mahta Moghaddam[1]

[1] *Department of Electrical Engineering/Electrophysics (EE), University of Southern California, Los Angeles, CA 90089, USA*
[2] *Electrical Engineering and Computer Science (EECS), University of Michigan, Ann Arbor, MI 48109, USA*

Correspondence should be addressed to Agnelo Silva, agnelors@gmail.com

Received 15 June 2012; Accepted 22 August 2012

Academic Editor: Hongxiang Li

Despite the well-known advantages of communication solutions based on energy harvesting, there are scenarios where the absence of batteries (supercapacitor only) or the use of rechargeable batteries is not a realistic option. Therefore, the alternative is to extend as much as possible the lifetime of primary cells (nonrechargeable batteries). By assuming low duty-cycle applications, three power-management techniques are combined in a novel way to provide an efficient energy solution for wireless sensor networks nodes or similar communication devices powered by primary cells. Accordingly, a customized node is designed and long-term experiments in laboratory and outdoors are realized. Simulated and empirical results show that the battery lifetime can be drastically enhanced. However, two trade-offs are identified: a significant increase of both data latency and hardware/software complexity. Unattended nodes deployed in outdoors under extreme temperatures, buried sensors (underground communication), and nodes embedded in the structure of buildings, bridges, and roads are some of the target scenarios for this work. Part of the provided guidelines can be used to extend the battery lifetime of communication devices in general.

1. Introduction

Energy harvesting has been an intensive research area in wireless sensor networks (WSNs). However, for many important WSN scenarios, such energy option is not feasible, and specific power-management strategies are necessary for WSN nodes that are powered by nonrechargeable batteries. For instance, when extreme temperatures are involved, the charging process of rechargeable cells is strongly impacted, as empirically demonstrated in our work. The behavior observed in Figure 1 for a specific node was repeated by many others and the network was impacted for many periods of consecutive days.

Also, buried nodes used in wireless underground sensor networks [1] and nodes embedded inside the walls of buildings [2], in the roads, or in the internal structures of a bridge, typically cannot employ rechargeable cells. On the other hand, when nonrechargeable batteries (or primary cells) are considered for WSNs, a high operational cost is usually expected [3]. This is typically the case even for very low duty-cycle WSN applications. Two design areas can be exploited in order to increase the lifetime of a battery in this scenario: networking protocols and power management. The focus of this work is on the latter area. More specifically, after analyzing several current WSN node designs, we systematically identified three power management aspects that allow a significant extension of the battery lifetime.

After implementing a customized WSN solution following the techniques proposed in this work, we observed that the battery's lifetime can be enhanced to almost 300% for some cases. More specifically, when the role of a WSN node is just taking and transmitting few measurements per day, the battery exchange will realistically occur closer to its age limit, typically between 5 to 10 years. With such results, a new generation of WSN applications for low-cost and unattended nodes becomes a reality. However, a trade-off to be considered is the increase on the high software/hardware complexity. Although the proposed techniques are tailored

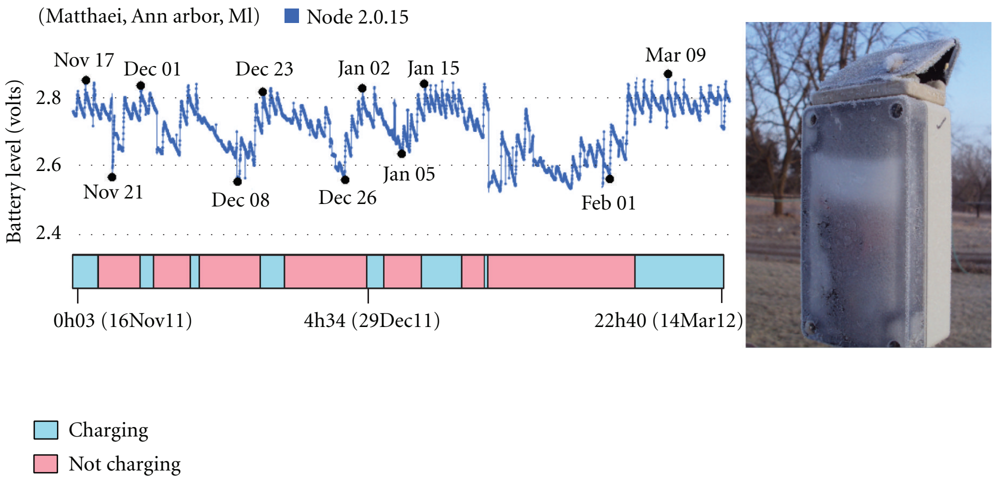


Figure 1: Even with sufficient solar irradiation in some days, the efficiency of the recharging process of two NiMH cells (1.5 V nominal voltage each) is drastically impacted by extreme low temperatures (<0°C). Also, the efficiency of a solar panel, in particular a small one, is strongly impacted by snow and ice.

to some classes of WSN applications, some of the design principles in this work can still be considered for generic WSNs and ultra-low-power devices in general.

The rest of this paper is organized as follows. In Section 2, the power-gating technique is investigated for the WSN domain. In Section 3, another power-management strategy for WSN nodes powered by nonrechargeable batteries is presented: the avoidance of voltage regulators while the WSN node is sleeping. In Section 4, the design of a basic power-matching circuitry for WSN nodes, based on supercapacitors, is presented. Analytical and empirical results combining the 3 techniques are discussed. In Section 5, a comparison between our approach and traditional power-managment techniques is presented. The paper is concluded in Section 6.

2. Power-Gating Technique

In this section, the power-gating technique is compared with the conventional power-saving approach. Next, the novel concept of *leakage gating* is introduced as an extension of the power-gating technique for any interconnection line that can potentially allow leakage current while a module is inactive.

The power-gating technique applied to WSNs has been proposed as a way to save energy for both active and sleep modes of a device [4]. Such technique is basically the introduction of an electronic switch between an electronic module or chip and the power-supply line. Typically, a power gate is implemented by a PMOS or NMOS transistor for cutting the ground or power lines, respectively. Therefore, it is possible to temporarily *shut down* blocks of circuitries that are not in use. For instance, while the sensor node is taking measurements usually, the radio transceiver is not required and it can be turned off.

Although the majority of the modules in a WSN node have some sort of standby pin (sometimes called shutdown or power save), such control typically cannot effectively switch off all the internal circuitries of the module. In general, the quiescent current for the power-saving mode is small but still higher than 1 μA. The main reason why manufacturers usually do not adopt a power-gating solution is simple: using power-saving, the module can quickly return to normal operation as soon as the standby pin returns to disabled mode. However, if the delay caused by completely turning off/on a device (e.g., 1 s) is not so critical for a given WSN application, an impressive energy reduction can be achieved by means of power-gating. This technique can be implemented externally to the module provided that the module does not have this functionality, as illustrated by the switches SW_A, SW_B, and SW_C in Figure 5.

2.1. Simulation Analysis. To highlight the significant energy reduction achieved with the power-gating (PG) technique, a simple architecture for a WSN node is considered. This node is formed by a processor, a sensing module, and a radio transceiver. It periodically wakes up, performs some processing (1 s), takes measurements (5 s), sends/receives data to/from the sink node (3 s), performs more processing (1 s), and finally sleeps again. Two scenarios are compared: (a) the power-saving technique, that is, the use of standby/sleep pins already available at the sensing and radio modules, and (b) the power-gating technique, implemented externally. It is assumed that the power source (e.g., batteries) is directly connected to the devices, that is, no additional loss due the existence of a voltage regulator or DC-DC converter. The values for the power consumed in each task/state are shown in Table 1. These values are typical ones based on off-the-shelf analog switches and modules used in WSN nodes.

In this simulation, we compare the total energy consumption levels for a 24 h period when different application duty cycles (expressed as number of measurements per day) are considered, as shown in Figure 2. Note that, for the scope of this work, we disregard the network impact on the energy consumption. In Figure 2, the total energy consumption per day for the power-saving mode can be inferred as the sum

Table 1: Typical power profile for a WSN node.

	MCU	Sensors	Radio
Active (regular operation)	5 mW	30 mW	350 mW
Inactive (power saving)	2 μW	5 μW	20 μW
Inactive (power gating)	2 μW	1 μW	1 μW

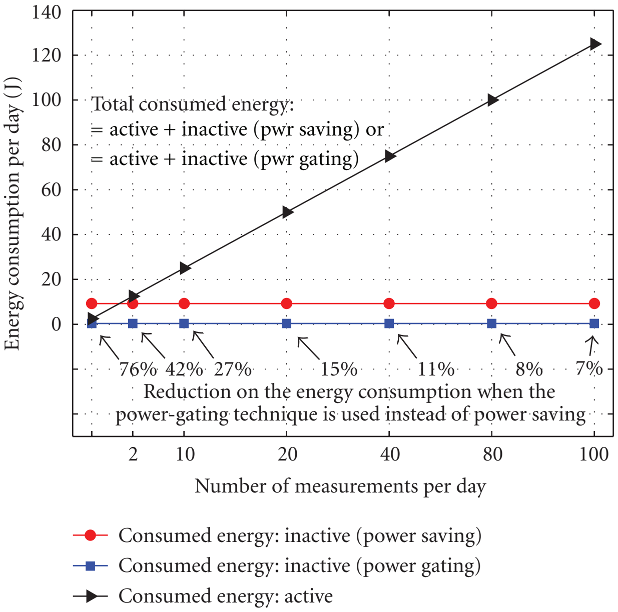


Figure 2: Energy consumption of a WSN node in an ideal scenario for different application duty cycles. Modules not being used are disabled using two techniques: power-saving and power-gating.

of the active and the inactive (power-saving) values. For the power-gating mode, similarly it is the sum of active and the inactive (power-gating) values.

As shown in Figure 2, as the frequency of measurements increases, the active energy also increases, as expected. Similarly, the inactive mode energy, corresponding to the intervals when the node is sleeping or has some of its modules disabled, must decrease when the number of cycles increases. Although this is true, Figure 2 does not clearly show this fact because such variations are very small compared to the active energy. Such observation also give us the intuition that the proposed PG technique does have a great impact on the energy when the duty-cycle goes beyond a certain point.

The most important aspect to highlight in this simulation is the significant energy reduction achieved with the power-gating technique. When an extremely low duty-cycle is considered, such as 0.02%, which corresponds to 2 measurements per day in Figure 2, such reduction is close to 76%. As the application duty-cycle increases, the advantages of the power-gating technique becomes less pronounced. For instance, for 100 measurements per day (1.16% duty-cycle), the energy consumption is only reduced by 7%.

Nonetheless, such low duty-cycle sense-send-sleep applications are not rare cases. Soil moisture measurements [5, 6] and antimold or similar solutions to be installed inside the walls of a building are some of such applications. Even when a higher sampling rate is still required, multiple sensors located at the same physical area can properly divide the workload at the time domain resulting in an individual low duty cycle. Although the concept of power-gating is not a novelty, none of the current off-the-shelf WSN nodes use such approach, with the exception of the Waspmote [7]. In the next section, we provide the explanations for this fact in conjunction with design guidelines.

2.2. Leakage Gating and WSNs.

We investigated the possibility of adopting the power-gating technique on typical off-the-shelf WSN nodes. The process is not simple because the complexity of the hardware and software significantly increases. For instance, due to the high time delay to effectively activate the radio, the current WSN communication protocols are typically impacted by the PG technique. After designing the proper software modules, the proposed technique proved to be feasible. By using external analog switches, the expected energy achievements shown in Figure 2 are confirmed.

Another factor that increases the complexity of the solution is the fact that not only power lines are associated with leakage: if two modules are connected, such as MCU and radio transceiver, any interface line between these modules can potentially be a leakage point even when one of them is completely powered off. The issue can occur with I/O lines and ADC channels. Typically, the interfaces lines must be empirically measured, one-by-one, in relation to the existence of leakage currents. To reach this goal, special low-power instrumentation is necessary. In our work, we evaluated the leakage influence of each MCU line using the special instrumentation in order to detect current variations on the order of nanoamperes [8].

Once a line with significant current leakage is detected, an additional analog switch can be added to that circuitry. We call such technique *leakage gating*, as an extension for the power-gating not restricted to power lines. Unfortunately, the choice of a PMOS or an NMOS transistor may not be straightforward as in the case of power lines. It is necessary to carefully verify the effects of the switch to the circuitry. Switch parameters such as $R_{\text{DS\_ON}}$ may be not so important in leakage gating as it is for a radio transceiver power gate. In many cases, a less expensive analog switch can be effectively used for leakage gating.

Nonetheless, the most important parameter to consider for the switch is its own current leakage (I_{LEAK}), also called *gate leakage*. Ideally, this current must be a small fraction of the leakage current which is expected to be controlled. In our experiments, we investigated 21 different types of switches and we opted for the use of a single optobased one with $I_{\text{LEAK}} < 50$ nA and $R_{\text{DS\_ON}} < 80\,\text{m}\Omega$.

As expected, with the addition of more switches, the complexity of the overall system significantly increases because its proper operation depends on the correct sequence and timing to switch on/off multiple leakage gates. In some cases, power lines and other correlated lines can be switched by a single pin/command. However, in other cases, the timing sequence is required, aggravating even more the time delay necessary for the full operation of the system. In fact, in our experiments using off-the-shelf MCU and radio transceiver, the required time delay reaches values as high as 0.5 s. Therefore, such leakage-control technique cannot be used in WSN nodes without substantial modification of the existing network software.

Fortunately, most I/O lines are connected to high impedance loads and the associated leakage is negligible, not justifying the leakage-gating switch. However, it was not the case for one the transceiver modules we tested: XBee-Pro (MaxStream Inc.). The reception line of this module has a leakage current of around 30 μA, more than 40 times the overall sleeping current of our final implementation of the node. As already explained, for very low duty-cycle applications (e.g., <0.5%), such additional loss is significant and we finally opted by using a switch for this line.

3. Avoiding Voltage Regulators

Voltage regulators are typically used for the proper operation of radio transceivers and sensor modules in WSN nodes. The energy efficiency of some of these regulators is very impressive and reaches values higher than 95%. Therefore, the inclusion of such regulators in the design of power modules for WSN nodes is a well-known practice. With an increasing emphasis on energy harvesting, rechargeable batteries usually are the main choice for energy storage. In this scenario, a voltage regulator for the processor (MCU) is typically a requirement due to the dynamic range of the voltage level of the cells and also to optimize the usage of the energy remaining at the batteries. Once we turn our attention to nonrechargeable batteries, it is not clear in the WSN literature if the discharging behavior of such primary cells also requires a voltage regulator for the MCU. In this section, we will demonstrate that in general this is not the case and the avoidance of such regulators for powering the MCU is an option to be evaluated.

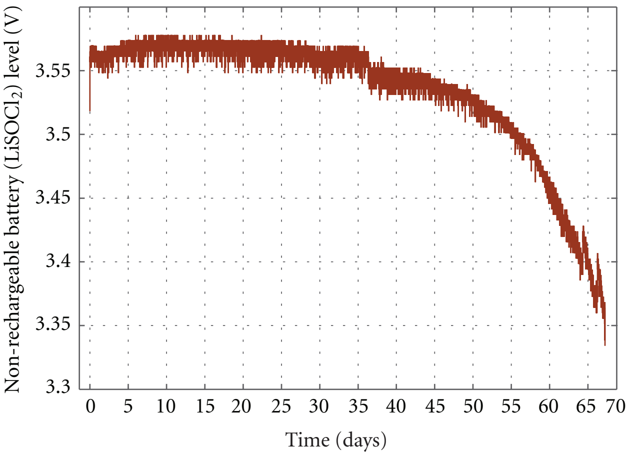


Figure 3: Battery level discharging profile. Experiment with a typical WSN node taking measurements and sending data every minute (17 dBm TX_{pwr}).

If we temporarily disregard the effect of pulse currents, a topic to be considered in Section 4, the voltage level of a primary cell has a small variation during the lifetime of the cell when compared with a rechargeable cell. For instance, we intensively used in our experiments high-capacity/low-current lithium thionyl chloride (LiSOCl_2) cells [9]. With an initial voltage level of around 3.65 V, the cell reaches its end of life with a voltage level of 3.33 V, as shown in Figure 3. Observe that such variation is typically acceptable for the majority of low-power MCUs and processors available in the industry. Therefore, no voltage regulator is required for this specific module.

The long-term experiment in our work related to Figure 3 involves node performing and transmitting soil moisture measurements (3 distinct sensors) every minute. Such high duty cycle was used to speed up the experiment. We can observe that the behavior of the battery is very stable. Also, we can infer that in a typical WSN application with 15-min cycle the battery used here would last more than 2 years. In this controlled laboratory experiment, the MCU is directly connected to the battery and the power-gating technique is used. A single-size-D ER34615 (3.6 V), 19 Ah cell is used in a WSN node with a power profile similar to the one in Table 1. To avoid influence from the instrumentation, all sensing and battery data are collected via wireless.

In contrast with a primary cell, the dynamic voltage range of a rechargeable cell is critical enough to require a voltage regulator for the MCU, as shown in Figure 4. In Table 1, we consider that the power consumption of the MCU in sleep mode is only 2 μW. Although this value is in agreement with the datasheet of many low-power MCUs, it is important to highlight that *no* voltage regulator is being considered in that scenario. If the regulator is included, only the power consumption of this component would typically be 1 or 2 orders of magnitude higher than the sleeping power consumption of the MCU. In short, while in sleep mode, the regulator typically dominates the power budget of the node. It is important to highlight that in the specifications of WSN nodes usually the sleeping power is given without adding the contribution of a voltage regulator. Unfortunately, due to the crescent focus on harvesting systems for WSN nodes, typically the voltage regulator for the MCU is always present and such approach is repeated in many WSN designs even when the node is powered by primary cells.

The avoidance of voltage regulators can be summarized as an effort to use MCUs/processors that support the voltage level variation of a nonrechargeable battery during all the lifetime of the cell. In our studies, we figured out that the implementation of this technique is relatively easy to be achieved. Even when voltage regulators are still required, such as the ones used for radio transceiver and for ADC measurements (analog sensors), the power-gating technique can be promptly used and these regulators will not contribute significantly to the power consumption of inactive devices.

The main drawback of such technique is the potential reduction of reliability of the system. For the solution related to the Figure 3, a careful design is considered. More specifically, it is very important to monitor if an expected low battery level can cause problems to the modules of the node, that is, to the stability of the system. For instance, in our first design, we used an external real-time clock (RTC) chip that presented problems when the battery level was close to its end-of-life. That problem was not promptly identified based on the datasheet information but we eventually solved the issue by selecting a different model for the RTC device.

A second design limitation for the proposed approach is to maintain all loads (i.e., MCU, sensor(s), radio transceiver), directly or indirectly connected to the battery system, under low-power level. For instance, due to the absence of a voltage regulator, a short high-current pulse from any load may restart the MCU. However, radio transceivers and sensors modules can potentially have an intermittent high-power profile. Moreover, when a low-current non-rechargeable battery drives high-current pulses, the lifetime of the cell is drastically reduced [10]. Unfortunately, it is not possible to accurately obtain such information from the battery's datasheet [9]. In our outdoors experiments, we faced lifetime reductions varying from 50 to 90%. Although high-power primary cells are available, they are expensive and their energy capacity are still drastically reduced. We tested 9 different models of $LiSOCl_2$ cells and concluded that a low-current model [9] can still be effectively adopted for WSNs. However, an additional technique is required to extend the lifetime of the system which is subject to high-current pulses. This aspect is considered in the next section.

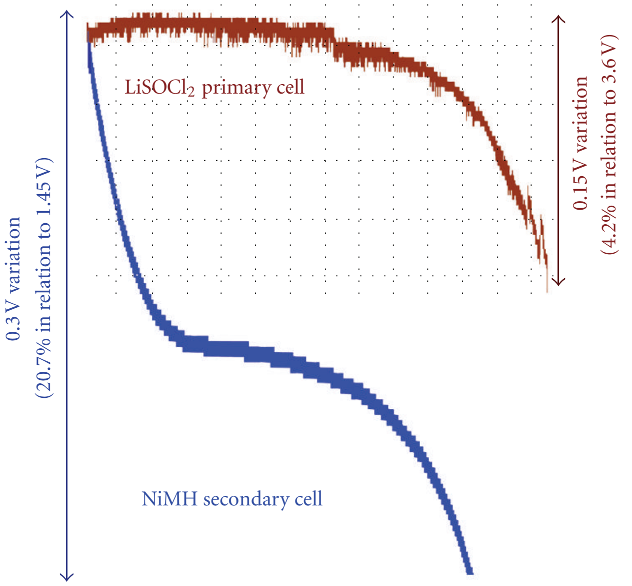


FIGURE 4: Typical-normalized discharging curves for primary and secondary cells in WSN nodes.

So far, the proposed power-management techniques considered a scenario where the WSN nodes operate in very low duty-cycles and with primary cells. The main goal behind those techniques is to reduce the active and inactive power consumption as much as possible. By *inactive* power, we are referring to the intervals when the module is not being used, in special when the node is sleeping. However, intermittent high currents from the sensors and radio modules are expected, as shown in Table 1, and can potentially decrease the lifetime of the primary cells. Therefore, the third technique presented in this section is not exactly a way to reduce the energy consumption, but a low-cost strategy to avoid the reduction of the nominal energy capacity of a primary cell.

The fact that the nominal lifetime of a primary cell can be affected by pulse currents [10] can be confirmed by analyzing the Table 1 in conjunction with Figure 3. Assume a 15-minute sampling rate, a total sleeping power of 4 μW, and that the first 2 mentioned techniques are in use. Therefore, the energy consumption for each cycle is around 1,114 mWs. For a 19 Ah primary cell and 3.53 V average voltage level, the theoretical maximum lifetime of the system is 2,258 days. Compare this value with 1,005 days extrapolated from Figure 3 for a 15-minute sampling rate ($67 * 15$). Therefore, a lifetime reduction of around 55% is expected even for a scenario where the temperature, self-discharging, and battery aging effects are highly minimized. In fact, based on the datasheet of the manufacturers [9], intermittent high-discharging currents from low-current primary cells can extremely reduce the lifetime of the cell (*pulse current* effect).

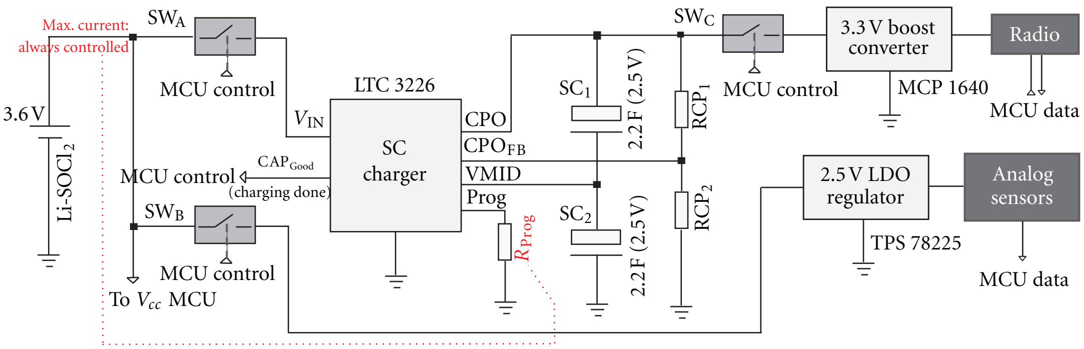


Figure 5: Basic power-matching circuitry for a typical WSN node: intermittent high-power load (bold lines) with a low-current/high-capacity nonrechargeable battery.

4. Power-Matching Technique

As a rule of thumb, the higher is the energy capacity of a cell, the smaller is its power capacity. For instance, an alkaline battery can provide higher currents but its energy capacity is many times smaller than the energy of a $\mathrm{LiSOCl_2}$ cell for the same physical volume. For a very long lifetime, a low-current profile cell (i.e., high energy capacity) is usually the proper option. Due to the pulse current effect, a *power matching* technique is proposed in order to allow the use of continuous low currents no matter if high-power loads are intermittently activated.

The typical component used for power matching is the supercapacitor (SC) [11]. A longer time to charge the SC (low-current charge) is traded in favor of a shorter high-current discharge. The smaller is the charging current, typically limited to 5–35 mA for $\mathrm{LiSOCl_2}$ cells, the longer is the charging process. By controlling that the maximum current never goes beyond the recommended *nominal, continuous* battery current as defined by the cell manufacturer, the mentioned pulse effect does not occur. However, as previously mentioned, a significant time delay (e.g., seconds to minutes) is introduced due to the need of slow-charging the SC(s). As a result, such technique cannot be effectively employed for WSNs or any other communication scenario where high data rate and low data delays are required. Moreover, the underlying network protocols must be very efficient and highly deterministic: the usage of the radio transceiver must be short and well defined.

Our proposed model for a power-matching circuitry is shown in Figure 5. This configuration is highly optimized for WSN nodes while maintaining relative low hardware complexity. The first aspect to consider in this model is the configuration for the SCs. The typical maximum voltage-level range for SCs (2.5 to 2.7 V) is usually not sufficient to power WSN radios. Even adopting a boost-converter between the SC and the radio, the operation of the converter would be mainly on a region with small energy efficiency. Therefore, 2 SCs connected in series is proposed as the ideal circuitry for this scenario. Although the effective capacitance decreases by 50%, the autodischarging issue is highly minimized, as explained latter in this section. Observe that the power-gating technique is applied for both SC charger and the radio transceiver. The SC charger can be simply implemented with a resistor. However, in order to reduce the charging time and increase the efficiency of the charging process, off-the-shelf solutions are recommended. In our experiments, we opted by using a commercial 2-cell SC charger (LTC3226 [12]).

Besides the introduction of a significant time delay, the power-matching circuitry also presents inefficiencies in terms of energy. There are multiple causes of losses even for the relative simple circuitry shown in Figure 5. These losses are very related to each other and also with the characteristics of the application. Therefore, a simplified (worst-case scenario) and systematic design approach is proposed in this section specifically for low-date rate and sense-send-sleep WSN applications.

Three Techniques Combined. As shown in Figure 5, the 3 techniques are combined for our final solution. The power-gating is achieved with 3 analog switches. Omitted for clarity, additional switches are used for some external lines of the MCU (leakage-gating). Note that the MCU is directly connected to the battery, that is, no voltage regulator is used for the MCU. Finally, 2 SCs, the SCs charger, and a 3.3 V boost-converter represents the power-matching circuitry.

Overall Strategy. Our ultimate goal is to verify if the total energy losses due to the power-matching technique is still smaller when compared with the loss due to the pulse current effect. Accordingly, a systematic approach is proposed in Figure 6. Next, we follow this approach using specific design options as a complete example.

Steps 1 and 2. In our case, the maximum voltage across the 2 SCs in series is 5 V because we used 2.5 V SCs. Sometimes, this value is limited by a smaller available battery level. This is not our case because SC charger we selected (LTC3226) has a charge pump: it charges SCs from 2.5 to 5.5 V. Therefore, in this example, the maximum SC voltage ($V_{\mathrm{SC}}^{\max}$) is constrained by 5 V. Next, we are interested in finding the optimum value for $V_{\mathrm{SC}}^{\max}$ and the answer is directed related to the choice of the voltage converter. We selected the MCP1640 3.3 V boost-converter [13] and based on its empirical performance we

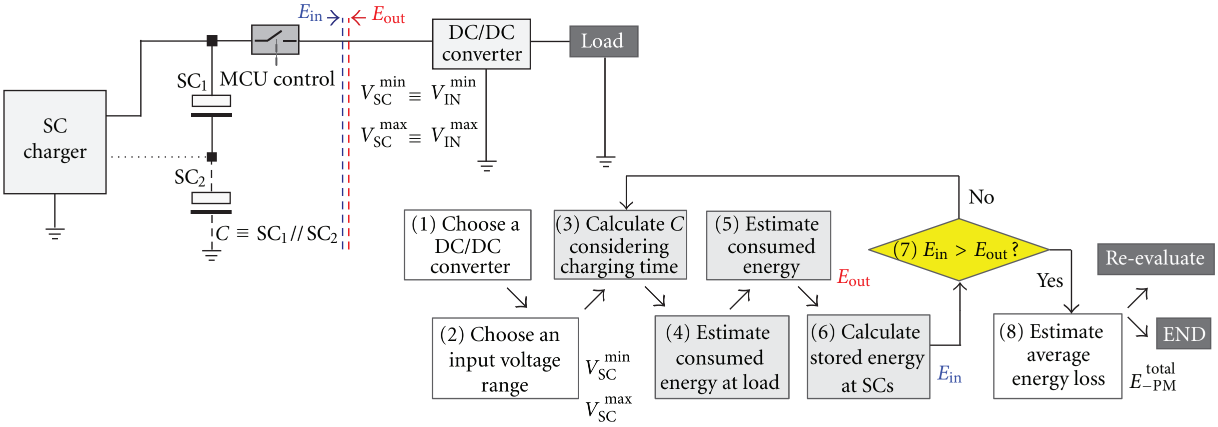

E_{in}
E_{out}
DC/DC converter
Load
SC charger
SC$_1$
MCU control
SC$_2$
$C \equiv$ SC$_1$//SC$_2$
$V_{\text{SC}}^{\min} \equiv V_{\text{IN}}^{\min}$
$V_{\text{SC}}^{\max} \equiv V_{\text{IN}}^{\max}$
(1) Choose a DC/DC converter
(2) Choose an input voltage range
$V_{\text{SC}}^{\min}$
$V_{\text{SC}}^{\max}$
(3) Calculate C considering charging time
(4) Estimate consumed energy at load
(5) Estimate consumed energy
E_{out}
(6) Calculate stored energy at SCs
E_{in}
(7) $E_{\text{in}} > E_{\text{out}}$?
No
Yes
(8) Estimate average energy loss
$E_{-\text{PM}}^{\text{total}}$
Re-evaluate
END

Figure 6: Strategy for the power-matching circuitry design.

defined its dynamic input voltage range as 1 V ($V_{\text{SC}}^{\min}$) to 3 V ($V_{\text{SC}}^{\max}$). The latter value can be adjusted by RCP$_1$ and RCP$_2$ shown in Figure 5.

Step 3. A small value of C may be not sufficient to store the required energy for the load. However, a higher C may require an excessive longer time for the SC charge. In this step, we would like to select a high value for C associated with a reasonable time delay for the application. The theoretical maximum time to charge 2 SCs in series, each one with capacitance C is $5R(C/2)$, where R is the virtual series resistance of the charging system. We just need to evaluate the worse case for this R, as expressed next. According to the final SC charging implementation, the charging time can be only a fraction of what we are estimating in this step. Considering our primary cell, we opted by a maximum charging current of 35 mA ($I_{\text{charge}}^{\max}$) which can be adjusted by R_{PROG} shown in Figure 5. Assuming a maximum voltage of 3 V ($V_{\text{SC}}^{\max}$), $T_{\text{charge}}^{\max}$ is given, in seconds, by

$$T_{\text{charge}}^{\max} = 5\frac{\text{VSC}_{\max}}{I_{\text{charge}}^{\max}}\left(\frac{C}{2}\right) = 2.5C\frac{3}{.035} \cong 214C. \qquad (1)$$

In our design, we finally selected $C = 2.2$ F, thus, $T_{\text{charge}}^{\max} = 470$ s. However, we empirically determined that, for completed discharged SCs, $T_{\text{charge}}^{\text{real}} = 165$ s. Moreover, if the SCs are not completely discharged, which is usually the case, this time is much smaller. Nonetheless, it is already clear that if we opted for $C = 100$ F, 15-minute cycle WSN application would not be feasible. Therefore, establishing reasonable boundaries for C is the goal of this step.

Step 4. We want to store sufficient energy in the SCs for properly powering the load for a certain time. However, an excessive nonused energy remained in the SCs represents an important loss (i.e., autodischarge) that we want to minimize. To this end, a worst-case scenario for the load behavior is defined and the maximum consumed energy at the load ($E_{\text{load}}^{\max}$) is estimated accordingly. In our case, we concluded that in the event of a communication failure, an additional transmission must be supported. In other words, the communication time must double, from 3 to 6 s. Therefore, $E_{\text{load}}^{\max}$ is estimated as follows:

$$E_{\text{load}}^{\max} = P_{\text{load}} * \text{Time}_{\text{active}} = (355\,\text{mW})(6\,\text{s}) = 2.13\,\text{J}. \qquad (2)$$

Step 5. Sometimes, it is necessary to empirically determine the average efficiency of the converter ($\eta_{\text{conv}}^{\text{avg}}$) for the entire input voltage range defined at Step 2. Alternatively, the information from the converter datasheet can be used to extrapolate $\eta_{\text{conv}}^{\text{avg}}$. We used the second approach to estimate $\eta_{\text{conv}}^{\text{avg}} \cong 85\%$. Small errors in this estimation can be corrected latter by adjusting the value of C (thus, $T_{\text{charge}}^{\max}$ also). Therefore, the maximum effective energy required from the SCs (E_{out}) is given by

$$E_{\text{out}} = \frac{E_{\text{load}}^{\max}}{\eta_{\text{conv}}^{\text{avg}}} = \frac{2.13}{0.85} = 2.5\,\text{J}. \qquad (3)$$

Step 6. From Step 3, a value for C was already selected. The associated energy stored (E_{in}) at the 2 capacitors, as a whole, is given by

$$E_{\text{in}} = \frac{1}{2}C_{\text{eq}}\left(V_{\text{SC}^{\max}}^2 - V_{\text{SC}^{\min}}^2\right) = \frac{1}{2}(1.1)\left(3^2 - 1^2\right) = 4.4\,\text{J}. \qquad (4)$$

Step 7. E_{in} must be higher than E_{out}, otherwise, we must return to Step 3 and choose a higher value for C (thus, increasing the time to charge the SCs). In this example, the goal is satisfied even if we consider a variation of the value of C as high as 40%. A relative high gap between E_{in} and E_{out} is usually necessary because the temperature and fabrication process can cause fluctuations on the value of C.

Step 8. At Section 3, we figured out that lifetime of the primary cell is reduced by 55%. It means that, for a 19 Ah 3.6 V cell, the pulse current effect causes an energy loss of 135 KJ. If during the battery's lifetime, the losses due to the power-matching circuitry becomes closer to this value, the technique loses its attractiveness. In this step, we want to estimate these losses ($E_{-\mathrm{PM}}$) to confirm if the power-matching technique is an option. One can empirically determine with accuracy $E_{-\mathrm{PM}}$, but this approach is not practical for a scenario involving multiple years. An alternative is to estimate the worst case ($E_{-\mathrm{PM}}^{\max}$): assume that all the energy at the SCs related to the voltages 0 to $V_{\mathrm{SC}}^{\min}$ are completely lost due to the autodischarge of the SCs. Therefore, if we assume that the discharge occurs as soon as the SCs are charged, $E_{-\mathrm{PM}}^{\max}$ is given by

$$E_{-\mathrm{PM}}^{\max} = \frac{1}{2} C_{\mathrm{eq}} V_{\mathrm{SC}^{\min}}^{2} = \frac{1}{2}(1.1)(1^2) = 0.55\,\mathrm{J}. \quad (5)$$

In our experiments, we actually verified that $E_{-\mathrm{PM}} < 0.2\,\mathrm{J}$ for each cycle, on the average. In fact, two factors contribute to the low autodischarge effect in this scenario. First, the value of C is relatively small and such effect is minimized. Second, although the maximum voltage at each SC is 2.5 V, the actual value is never beyond 1.5 V ($V_{\mathrm{SC}}^{\max}/2$). The closer the voltage level is to the upper limit, the higher the autodischarging. However, we still need to estimate the cumulative losses for all cycles ($E_{-\mathrm{PM}}^{\mathrm{total}}$), which is given by (for 15-minute cycles):

$$E_{-\mathrm{PM}}^{\mathrm{total}} = N_{\mathrm{cycles}} * E_{-\mathrm{PM}}^{\max} = \frac{1005\ \mathrm{days}}{15\,\mathrm{min}}(0.55) = 53\,\mathrm{KJ}. \quad (6)$$

Even with a conservative approach, the power-matching technique will reduce at least 60% of the pulse current effect on primary cells. However, if this technique is found not to be an option, multiple cells in parallel is a simplistic solution that can potentially mitigate the mentioned issue. The trade-offs of this alternate option are the higher cost and higher physical volume of the node.

5. Comparison with Typical Power Management Approaches

Due to the fact that some of the techniques presented in this paper have been used in the industry in different scenarios, the goal of this section is to highlight the contributions of our work in particular for solutions based on WSNs.

Power-Gating Aspects. Although is a well-known technique, for the best of our knowledge, we present the first real-world implementation of the PG technique for a WSN node also associated with the extension of this technique, called *leakage gating* (LG). This latter technique is the application of the PG concept to any other line that presents significant current leakage when the node is in sleep mode. An important trade-off of the PG/LG techniques is the larger form factor (and higher cost) introduced with the additional power/signal switches. However, because the total cost of a WSN node is also associated with its support, the ultimate question behind the PG/LG is if the higher cost of a node justify the expected costs reduction associated to the frequent need of exchanging batteries. For the antimold example mentioned in this paper, the higher deployment cost due to PG/LG seems to be justified. The same for a node deployed in an area of difficult (or expensive) access. However, for certain cases, the technique may not be justified or, at least, just a small number of PG/LG switches can be realistically important to save energy.

Regulator Avoidance Aspects. The proposed technique in this paper is specifically associated with the main MCU and its RTC, if available. In a WSN node, these two components must be continuously connected to a power source when the node is sleeping/hibernating. If a voltage regulator is used in series with the power source, such regulator will always contribute for the sleeping power of the WSN node. Therefore, by eliminating this components we are minimizing the total sleeping power of the node. We highlighted in this paper that such approach is potentially feasible with primary cells but not with secondary cells. Nonetheless, the ultimate application of this technique is dependent on the choice of the primary cell, MCU, and RTC components of the WSN node. For instance, if the maximum supply voltage for an ultra-low-power MCU is 1.8 V, it is clear that the inclusion of a voltage regulator is a wise decision if a 3.6 V primary cell is involved. With other chemical types of cells, different voltage levels are provided and 3.6 V must not be considered a fixed value for the mentioned technique. If the technique is expected to be used, it is important to find the proper battery type/configuration that is best tailored to the MCU and RTC.

Supercapacitor Charger. At the time this paper was written, there were few off-the-shelf SC charger chips available. However, the efficiency of the charger device used in our work clearly justifies its adoption rather than designing the circuitry for this task. In particular, the chip we used in our implementation (LTC3226 from Linear Circuits, Inc.) takes care of the charge balance between the SCs without the risk of having a voltage across an SC which is higher than its nominal maximum voltage.

Quiescent Current of a Voltage Regulator. When considering the voltage regulator avoidance technique, one can highlight that modern regulators have a very small quiescent current (e.g., <1 μA) which prevents the adoption of the technique. Such argument is potentially valid in relation to the PG technique. For instance, if we assume that the 2.5 V LDO in Figure 5 can be disabled and still maintain a very low quiescent/shutdown current, then the PG switch SW_B is not justified. However, if we consider the argument again for the voltage regulator technique, it is rarely valid. The goal of this technique is to connect the MCU (and/or RTC) components directly to the power source, such as a nonrechargeable battery in order to reduce the *sleeping* power of the node. Typically, the quiescent current in the datasheet of a voltage regulator assumes no load or, in some cases, the complete deactivation of the regulator. However, because the MCU still requires a small current while sleeping, the nominal value of the quiescent current may not be applied in this case and tests must be conducted. On the other hand, if an active regulator still has high efficiency (e.g., >80%) while supplying very small currents to MCU and RTC modules (the former in sleep mode), then the argument holds and the adoption of the voltage regulator is justified. Therefore, such aspect must

Table 2: Overview of the proposed power management techniques.

Technique	Costs	Software complexity	Intrinsic energy inefficiencies	Main trade-off	Usage indication
Power gating (PG)	Add + 10%	High	Very low	High data latency	Very low duty-cycle app
Voltage regulator avoidance	None	None	MCU w/high volt. can be energy-inefficient	MCU can be unstable	Primary cells + low duty-cycle app
Power matching (PM)	Add + 15%	High	May have + 30% energy losses	High data latency	Primary cells + low duty-cycle app

be evaluated case-by-case. In our implementation, we did not find a regulator which satisfied the mentioned requirements and it was not adopted, as shown in Figure 5.

Reducing Operating Voltage of the MCU. A technique so far not mentioned in this work, but highlighted by the processors/MCUs manufacturers, is the reduction of the operating voltage. Typically, this approach is also associated with a smaller clock speed. Nonetheless, such technique basically reduces the power consumption while the device is in *active* mode. In general, the technique does not provide significant advantages when the processor is in sleep mode. In other words, if low duty cycle applications are *not* assumed, the adoption of this technique is strongly recommended. That is, a value such as 5 mW in Table 1, could be reduced to smaller value, such as 1 mW. On the other hand, the adoption of a smaller MCU voltage may require a voltage regulator between the power source and the MCU. Because such device can introduce significant loss when the MCU is sleeping, there is a minimum application duty-cycle that supports the adoption of the technique. In other words, the mentioned voltage reduction technique would decrease the height of the line *Consumed Energy: active* in Figure 2 but also would increase the height of the other two lines (inactive) due to the additional losses caused by the introduction of a voltage regulator for the MCU. For the context of this work (low duty-cycle applications) and the selected components used in our sensor node implementation, the reduction of the operating voltage of the MCU was not considered a proper option.

Real-World Implementations in WSN Nodes. to the date, there is no report about a WSN node that implements the combined techniques mentioned in this work. The Waspmode device (Libelium Comunicaciones Distribuidas S.L.) [7] implements PG switches for multiple modules. Even related to the avoidance of a voltage regulator, the models of WSN modules that target rechargeable batteries, employ a voltage regulator for the MCU. In our work, we designed three different WSN nodes: Ripple-2A, Ripple-2D, and Ripple-2D+. The first one is based on rechargeable batteries and solar panel (Figure 1 and implements the PG technique in 4 modules and also the regulator avoidance technique. The second device is based on nonrechargeable battery (3.6 V) and implements the PG and regulator avoidance techniques. The third node, the default node for our real-world deployments, implements the three techniques analyzed in this paper. The first deployment (Matthaei Bothanical Gardens, Ann Arbor, MI) was realized on August 2011 and involved 26 sensor nodes. The second deployment (Canton, OK) was realized on September 2011 and involved 21 nodes. The third deployment (Sacramento, CA) has started on August 2012 and involves 150 nodes spread in an area larger than 3×3 km^2 [14].

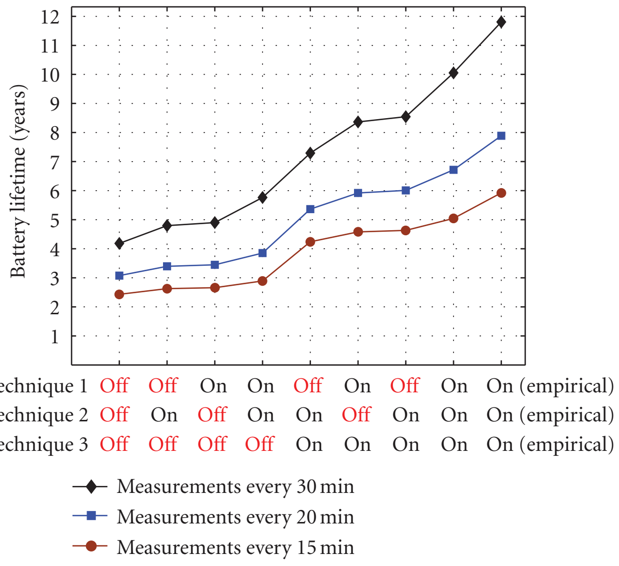


Figure 7: Lifetime of a WSN node using a 19 Ah battery according to different power techniques.

This third deployment mentioned above is solely based on Ripple-2D+ which employs PG by means of 4 analog switches PS710-B (NEC Inc.). Beside the *Turn-On Time* (TOT) due to the switch (up to 5 ms), each load has its own TOT characteristic. For instance, we empirically determined that the TOT of the radio transceiver XBee Pro SBP1 (Digi Inc.) is higher than 200 ms. Such value is high enough to prevent the use of the PG technique for the majority, if not all, of the WSN protocols and new networking protocols must be designed considering PG. Similarly, the PM technique can potentially increase the transmit delay because

SC(s) must be charged before the realization of a radio communication or similar high-power task. Therefore, a very deterministic network behavior is expected in order to properly design the amount of charge regularly stored in the SCs. In our sensor node implementation, more than 15% of the code is related to the management of the PM technique. Also, the SC circuitry corresponds to more 15% of the costs of our sensor node.

6. Conclusions

Three power-management techniques are proposed specifically for sensor nodes used in low duty-cycle WSN applications and powered by primary cells. Accordingly, a customized node is designed and both laboratory and outdoor experiments are performed. Semiempirical models are used for the simulations and the results are summarized in Figure 7. Disregarding network aspects, these simulations consider worst-case scenarios and our empirical results show significantly better results. In all cases, we conclude that the lifetime of a node can be strongly extended by multiple folds with the approaches proposed here.

The main trade-offs of the solution are the increase of both data-latency and complexity. The third technique, in particular, imposes operational delays on the order of minutes constraining its application to some communication scenarios. Also, the effects due to aging and extreme temperature variations are not considered in our models. Accordingly, we envision two directions for our future work: the development of novel WSN protocols to take advantage of the proposed techniques and the realization of long-term experiments (multiple years).

In general, only low (or very low) duty-cycle applications can be indicated for these techniques, as shown in Table 2. Nonetheless, we believe that the future of many networks reside in the capability of dynamically changing between multiple operational modes, being low duty-cycle one of these modes. While in this mode, all the power-management techniques discussed in this paper can be potentially can be adopted as hardware modules that can be dynamically activated and deactivated by software making the final WSN node design very flexible for different and temporary needs.

References

[1] A. R. Silva and M. C. Vuran, "Empirical evaluation of wireless underground-to-underground communication in wireless underground sensor networks," in *Proceedings of the 5th IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS '09)*, Marina Del Rey, Calif, USA, June 2009.

[2] W. Q, P. Angove, J. Buckley, J. Barrett, and G. Kelly, "Design and performance analysis of an embedded wireless sensor for monitoring concrete curing and structural health," *Journal of Civil Structural Health Monitoring*, vol. 1, pp. 47–59, 2011.

[3] A. S. Weddell, N. R. Harris, and M. N. White, "Alternative energy sources for sensor nodes: rationalized design for long-term deployment," in *Proceedings of the IEEE International Instrumentation and Measurement Technology Conference (IMTC '08)*, pp. 1370–1375, British Columbia, Canada, May 2008.

[4] M. A. Pasha, S. Derrien, and O. Sentieys, "Toward ultra low-power hardware specialization of a wireless sensor network node," in *Proceedings of the IEEE 13th International Multitopic Conference (INMIC '09)*, pp. 1–6, Islamabad, Pakistan, December 2009.

[5] H. R. Bogena, J. A. Huisman, C. Oberdörster, and H. Vereecken, "Evaluation of a low-cost soil water content sensor for wireless network applications," *Journal of Hydrology*, vol. 344, no. 1-2, pp. 32–42, 2007.

[6] M. Moghaddam, D. Entekhabi, Y. Goykhman et al., "A wireless soil moisture smart sensor web using physics-based optimal control: concept and initial demonstrations," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 3, no. 4, pp. 522–535, December 2010.

[7] "Waspmote," http://www.libelium.com/products/waspmote.

[8] "2400 Sourcemeter," http://www.keithley.com/.

[9] "Primary lithium batteries ER34615 (lithium thionyl chloride)," http://www.batteryspace.com/prod-specs/ER34615-BS-R.pdf.

[10] M. Doyle, T. Fuller, and J. Newman, "Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell," *Journal of the Electrochemical Society*, vol. 140, no. 6, pp. 1526–1533, 1993.

[11] C. Menachem and H. Yamin, "High-energy, high-power pulses plus battery for long-term applications," *Journal of Power Sources*, vol. 136, no. 2, pp. 268–275, 2004.

[12] "LTC3226—2-Cell supercapacitor charger," http://www.linear.com/product/LTC3226.

[13] "MCP1640 Data Sheet," http://www.microchip.com/.

[14] A. R. Silva, Mingyan Liu, and Mahta Moghaddam, "Ripple-2: a non-collaborative, asynchronous, and open architecture for highly-scalable and low duty-cycle WSNs," in *ACM Intl Workshop on Mission-Oriented WSN (MiSeNet '12)*, Instanbul, Turkey, August 2012.