Summer 2005

An analytical model of MAC protocol dependant power consumption in multi-hop ad hoc wireless sensor networks

Komlan Egoh
New Jersey Institute of Technology

ABSTRACT

AN ANALYTICAL MODEL OF MAC PROTOCOL DEPENDENT POWER CONSUMPTION IN MULTI-HOP AD HOC WIRELESS SENSOR NETWORKS

by
Komlan Egoh

Power efficiency is the most constraining requirement for viable operation of battery-powered networked sensors. Conventionally, *dynamic power management* (DPM) is used to put sensor nodes into different states such as *active*, *idle*, and *sleep*, each consuming a certain level of power. Within the active state, communication operational states, such as *receive* and *transmit* consume different levels of nodal power. This thesis shows how DPM states and protocol operational states can be combined into a single stochastic model to finely evaluate the power consumption performance of a medium access control (MAC) protocol. The model is formulated as a *semi-Markov decision process* (SMDP) wherein the node's states, sojourn times, and transition probabilities are controlled by a *virtual node controller*. The overall operation of a communication protocol is viewed as a randomized policy for the SMDP, and the long-run average cost per unit time measures the energy efficiency of the protocol.

AN ANALYTICAL MODEL OF MAC PROTOCOL DEPENDENT POWER CONSUMPTION IN MULTI-HOP AD HOC WIRELESS SENSOR NETWORKS

by
Komlan Egoh

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Internet Engineering

Department of Electrical and Computer Engineering

August 2005

APPROVAL PAGE

AN ANALYTICAL MODEL OF MAC PROTOCOL DEPENDANT POWER CONSUMPTION IN MULTI-HOP AD HOC WIRELESS SENSOR NETWORKS

Komlan Egoh

Swades De, Dissertation Advisor — Date
Assistant Professor of Electrical and Computer Engineering, NJIT

Nirwan Ansari, Committee Member — Date
Professor of Electrical and Computer Engineering, NJIT

Sirin Tekinay, Committee Member — Date
Associate Professor of Electrical and Computer Engineering, NJIT

BIOGRAPHICAL SKETCH

Author: Komlan Egoh

Degree: Master of Science

Date: August 2005

Undergraduate and Graduate Education:

- Master's of Science in Internet Engineering
  New Jersey Institute of Technology, Newark, NJ, 2005

- Engineering Diploma in Electrical Engineering,
  Université de Lomé, Lomé, Togo, 2001

Major: Internet Engineering

Presentations and Publications:

K. Egoh and S. De "Stochastic Modeling of Power Consumption in Ad Hoc Wireless Sensor Networks," Submitted to *IEEE INFCOM* Conference, Barcelona, 2006.

K. Egoh "A Client-Server Solution for Access Control and Management of Public Internet Centers," A Thesis submitted to the faculty of *ECOLE NATIONAL SUPERIEUR D'INGENIEURS* in Partial Fulfillment of the Requirements for the Degree of *Diplome d'Ingenieur* in Electrical Engineering, Lomé, Togo 2001.

The mere formulation of a problem is far more essential than its solution, which may be merely a matter of mathematical or experimental skills. To raise new questions, new possibilities, to regard old problems from a new angle requires creative imagination and marks real advances in science.

– Albert Einstein

ACKNOWLEDGMENT

I would like to express my deepest appreciation to Dr. Swades De, who not only served as my research supervisor, providing valuable and countless resources, insight, and intuition, but also constantly gave me support, encouragement, and reassurance. Special thanks are given to Dr. Nirwan Ansari and Dr. Sirin Tekinay for actively participating in my committee.

Many of my fellow graduate students in the New Jersey Center for Wireless Communication and Internet Security (NJWINS) are deserving of recognition for their companionship and support.

I also wish to thank my beloved family, my mother, my brothers and sisters in Togo, Josiane and all my friends for their continual support over the years.

TABLE OF CONTENTS

Chapter	Page
1 INTRODUCTION	1
1.1 Motivation	1
1.2 Related Work	4
1.3 Background	6
1.3.1 Controlled Stochastic Systems	6
1.3.2 Semi-Markov Decision Processes (SMDP)	7
1.3.3 Network Topology	8
1.3.4 Traffic Forwarding Scheme	9
2 POWER CONSUMPTION MODEL	12
2.1 Receive State Analysis	12
2.1.1 Action/Event Selection Probabilities	12
2.1.2 Sojourn Times	15
2.1.3 Transition Probabilities	17
2.2 Idle State Analysis	19
2.2.1 Action/Event Selection Probabilities	19
2.2.2 Sojourn Times	21
2.2.3 Transition Probabilities	21
2.3 Transmit State Analysis	22
2.3.1 Action/Event Selection Probabilities	23
2.3.2 Transition Probabilities	23
2.3.3 Sojourn Times	23
2.4 Sleep State Analysis	24
2.4.1 Action/Event Selection Probabilities	25
2.4.2 Sojourn Times	25
2.5 MAC Contention	26

TABLE OF CONTENTS
(Continued)

Chapter **Page**

2.6 Cost Structure 26

3 NUMERICAL RESULTS 28

4 CONCLUSION 33

REFERENCES 34

LIST OF TABLES

Table **Page**

1.1 Typical Power Consumption Data 3

LIST OF FIGURES

Figure		Page
1.1	Functional diagram of a network node	4
1.2	Data packets and ACK packet follow opposite directions	9
1.3	Schematic for calculating the relay traffic.	10
2.1	Schematic of a sample path of a node as a dynamic stochastic system.	13
2.2	Decision rule in *receive* (*Rx*) state.	13
2.3	Data transmit and receive process	15
2.4	Decision rule in *idle* state.	19
2.5	Decision rule in *transmit* (*Tx*) state.	22
2.6	Decision rule in *sleep* state.	24
2.7	Decision rule in MAC contention state.	26
3.1	Energy and throughput performance	29
3.2	Effect of sleep cycle on energy and throughput performance ($d = 40$, $\lambda_n = 0.01$)	29
3.3	Energy consumption map of the sensor field with low traffic load	31
3.4	Power consumption map of the sensor field with high traffic load	31
3.5	Power consumption at various sleep time. $\lambda_n = 0.1, d = 40$.	31

CHAPTER 1

INTRODUCTION

1.1 Motivation

In designing network protocols for sensor networks, while the constraints of limited channel bandwidth, memory resources, and processing capabilities play major roles, a fundamental constraint that needs paramount attention is the limited power/energy resource. In many sensor network applications, the sensor nodes are battery powered and cannot always be replenished once they are drained. Therefore, it is very important that the limited energy resource is utilized judiciously.

Although there have been a lot of ongoing research on power/energy efficiency, many of the proposed protocols for sensor networks are based on the observations from wireless LANs (WLANs). However, power consumption properties of the tiny sensor nodes are vastly different from that observed in WLAN devices. Particularly, unlike in WLANs,

- the receive power consumption dominates over the consumptions due to other activities [4],
- nodal range of sensors being significantly smaller (10 to 30 meters), power saving by transmit power control may not be significant, and
- idling power consumption could be significant in sensing applications where the data exchange rate is low.

In particular, due to miniature size of the nodes, even little functionalities, such as overhearing, idle listening, etc. consume significant fraction of the total energy reserve of a node [1], [2], [3].

There also have been a few recent low power aware sensor network protocols that take into account the uniqueness of the sensors' power consumption properties. Yet, in seeking optimum network performance at the minimum energy cost, and to devise efficient new protocols or assess the effectiveness of the current ones, it is very important to have a detailed understanding of communication related power consumption properties of a sensor node in different operational conditions.

Since in cellular networks and WLANs for example coverage range is relatively large, up to a few kilometers, total power consumption at a node is dominated by the consumption due to signal transmission. The power consumption due to transmitter and receiver electronics are relatively small. Therefore, in a cellular/WLAN node with transmission power control capability, nodal power consumption was used to be considered dependent on the transmit-receive distance only and not on the electronic processing [5]. As the nodal coverage range decreases, the above method of determining nodal power consumption does not hold good. As an example, in Cisco Aironet PCM 350 series client adapter cards, the transmit power consumption (measured in terms of current drawn) at the maximum power output (20 dBm) is 450 mA, whereas the receiver electronics power consumption is 270 mA and idle mode electronics power consumption is 15 mA [17]. This implies that power consumption due to electronic processing can no longer be ignored. Indeed, energy consumption measurements by Feeney and Nilsson [18] on Lucent 802.11 WaveLAN cards showed how in addition to the *transmit* mode, *receive* and *idle* modes contribute to the total energy consumption of a node. Another series of measurements on Cisco Aironet PC 4800B PCMCIA 802.11 WLAN cards were conducted by Burns and Ebert [19] that took receive power consumption into consideration.

As further miniaturized devices are considered, which is the case with the sensor nodes, the nature of power consumption is in fact dominated by electronics power consumption. Typical data of a Chipcon sensor transceiver CC2420 [4] in Table 1.1 indicate that receiver processing power consumption is the major component.

Table 1.1 Typical power consumption data

Mode	Current drawn
Transmit:	
0 dBm (max) ··· ··· ··· ··· ···	17.4 mA
−25 dBm (min) ··· ··· ··· ··· ···	8.5 mA
Receive ··· ··· ··· ··· ··· ··· ···	19.7 mA
Idle (oscillator and voltage regulator on)	426 μA
Sleep (voltage regulator on) ··· ··· ···	20 μA

Moreover, in contrast with the cellular/WLAN devices, the nodal range of sensors being much smaller (10 to 30 meter), per node power saving by transmit power control may not be significant in sensor networks.

In this work, the power consumption of peer-to-peer communicating sensor nodes is investigated and modeled. Along with a manufacturer provided transceiver's power consumption data (see for example, Chipcon CC2420 datasheet, [4, p.13]), it is also important to take into account the protocol-dependent power consumption overheads in different modes, that will make it possible to compare the performances of different medium access control (MAC) protocols. According to dynamic power management (DPM) strategy [6], in the interest of energy saving, at any point in time an operational sensor tends to remain in a low power consuming state. It is assumed that a networked sensor remains in one of the five states: *transmit* (*Tx*), *receive* (*Rx*), *MAC contention*, *idle*, and *sleep*. A virtual control entity is introduced with functionally similar the Power Manager in a DPM model. While the PM is a software or hardware entity implemented in the MAC or PHY Layer, the virtual controller is a conceptual central controller. DPM decisions and communication events are assumed *a priori* decisions of the virtual controller. To capture the power consumption properties of a dynamically power managed sensor node, the MAC protocol dependent state transition probabilities and sojourn times in different states will first be identified. The power consumption model is formulated as a semi-Markov decision process (SMDP). The sojourn time in a state and the state transition probabilities are decided by a virtual controller. Thus, a sensor node is treated as a controlled stochastic dynamic system (see Figure 1.1). The sojourn times and state transitions are associated with a device-specific

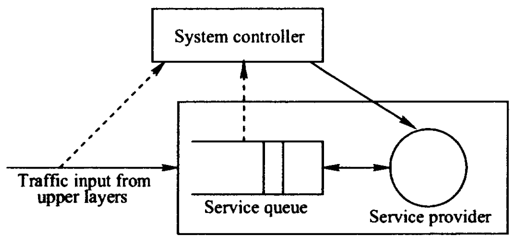

Figure 1.1 Functional diagram of a network node, represented as a controlled stochastic dynamic system.

cost. Since the states and transitions are nodal activity and channel condition dependent, the overall operation of a communication protocol is characterized as a randomized policy for the SMDP. The energy efficiency of the protocol is measured as the long-run average cost per unit time.

The energy consumption model in this report is developed with the assumption of a preamble sampling based low power protocol [2], [7], [8]. The objective is to study different parametric relationships with the average nodal energy consumption. It is shown that the developed model can be used as an optimization tool for a given communication protocol. For example, it would lead us to capture the optimum sleep time of nodes and data packet size for maximizing transmission success rate with a minimum energy cost. The model can also be used in comparing energy efficiency of different communication protocols in sensor networks, and can provide a guideline to determining energy consumption pattern for a given node and sink deployment strategy and vice versa.

The rest of this report is presented as follows: The concepts of SMDP and the assumed network model are introduced in the rest of Chapter 1. The state-dependent power consumption model is developed in Chapter 2. Chapter 3 contains an application of the power consumption model and presents results obtained trough numerical computation. Concluding remarks, related works and future extension of this work are outlined in Chapter 4.

1.2 Related Work

Although there is a lot of literature on power efficient and distributed communication protocol design for sensor networks there have been few work on protocol related power consumption modeling. Dynamic power management scheme proposed in [11] introduced the concept of turning on and off the operating system intelligently to minimize the energy wastage in power aware sensor nodes. Bollow are highlighted the works related to the current study. It has been observed in [1] and [3] that overhearing is a major source of energy consumption in sensor network, and its effect on sensor network energy efficiency is characterized in [3]. S-MAC [1] extends the concept of RTS-CTS in 802.11, and reduces the data transmission coordination related energy consumption by locally synchronizing at the beginning of every activity cycle. It has been demonstrated in [2], [7] that preamble sampling based Aloha-like access protocol operates power efficiently by reducing idle listening energy cost. Based on the understandings of power consumption in various nodal states, B-MAC [8] has been recently proposed. B-MAC operates on preamble sampling based approach and provides the system designers the flexibility of choosing between energy awareness and service criteria.

A few recent works have focussed on system performance and energy optimization via power consumption modeling. System level power optimization have been considered [9], [13]. The authors uses continuous-time Markov decision process based model to characterize power consumption in different states and obtain the optimum state transitions. While continuous-time MDP are suitable for optimal control problems in which the primary goal is to find an optimal policy for a real power manger implemented as software or hardware component, the limitation of exponentially distributed inter-decision time of the continuous-time model makes it less applicable to the modeling of complex system under the influence of random events or the combined action of different system controllers. Discrete-time Markov process based state modeling has been used in [14], where a sensor is considered to have two states: sleep and active. It was shown that by controlling the state of nodes, the optimal interference and routing performances can be achieved. In both continuous-time and discrete-time Markov process based model, the dwell time in a state (also called sojourn time) is assumed to have memoryless distribution. In [15], optimal node placement strategy in a one-dimensional sensor network is studied in order to achieve energy optimized system performance.

The proposed approach is focused on power consumption related to the operation of given protocols in the communication stack. By using SMDP model the proposed methodology offers a general framework that enables protocol designers gain in-depth understanding the node and network behavior under complex operational condition.

1.3 Background

1.3.1 Controlled Stochastic Systems

Dynamic stochastic systems are the systems that evolve with time under the influence of a control entity or random events. Power-managed sensor nodes are examples of a dynamic stochastic system under the influence of the combined action of communication protocols and power management policies. In general, dynamic stochastic system can be classified as (i) continuous-time and (ii) discrete-time, if the system is observed in continuous and discrete-time respectively. Another system classification is (i) controlled and (ii) uncontrolled, depending on the existence or not of an identifiable system controller.

In all classifications, the system is observed at the initial time $t_0 = 0$ and is found in one of a finite (possibly countably infinite) number of states $X_0 = i,\ i = 0, 1, 2, \cdots, I\ (I \leq \infty)$. After a sojourn time of τ_0, the system jumps into another state j $(X_1 = j)$ at the time instant $t_1 = t_0 + \tau_0$ with probability p_{ij}, stays for τ_1 before jumping to another state k

$(X_2 = k)$ at time $t_2 = t_1 + \tau_1$ with probability p_{jk}, and so fourth. The two processes $\{X_\ell\}$ and $\{\tau_\ell\}$, $\ell = 0, 1, 2, \cdots$, constitute the stochastic model of the dynamic system.

In a controlled dynamic system, a controller or a decision maker is allowed to influence the system by choosing at each decision epoch one of a finite (possibly countably infinite) number of actions a_κ, $\kappa = 0, 1, 2, \cdots, K$ $(K \leq \infty)$ The evolution of the system among states and the corresponding sojourn times within states are therefore determined by the actions of the decision maker. In other words, the distribution of the two processes $\{X_\ell\}$ and $\{\tau_\ell\}$, $\ell = 0, 1, 2, \cdots$, become action dependent.

The theory of controlled stochastic systems offers practical analytical frameworks for the study of (i) optimal control problem, and (ii) performance evaluation problem. While the earlier is the primary concern of DPM solutions [6], [9], [10] which seek to find an optimal decision strategy (or optimal policy) among all feasible decision strategies, the later focuses on the evaluation of a given decision strategy. In both cases, a cost (or reward) structure is associated with a strategy, where each state and state transition of the system incurs a different level of system cost. The primary focus in this thesis work is the performance evaluation of a given strategy (communication protocol).

Before modeling the different states of a sensor under a given MAC protocol, formal presentation of SMDP and the outline the assumed network architecture and multi-hop communication protocol that would affect the nodal states and state transitions are first given.

1.3.2 Semi-Markov Decision Processes (SMDP)

Consider a controlled stochastic dynamic system as introduced earlier. The system with state space $\mathcal{I} = \{0, 1, 2, \cdots, I\}$ is controlled by a sequential decision maker with action space $\mathcal{A} = \{a_0, a_1, a_2, \cdots, a_K\}$. The decision maker reviews the state of the system at given (possibly random) epochs and must take a decision. In each state $i \in \mathcal{I}$, a set of actions $A(i) \subset \mathcal{A}$ is allowed. As a consequence of selecting an action at a decision epoch, a reward (or cost) is incurred.

Denote the system state X_0 at time t_0, and let t_ℓ and X_ℓ, $\ell = 0, 1, 2, \cdots$, be the subsequent decision epochs and the corresponding system states, respectively.

Definition 1 *The above defined model is said to be an SMDP if the embedded process* $\{X_\ell\}$, $\ell = 0, 1, 2, \cdots$, *has the Markov property, i.e., the state of the system at the next decision epoch depends on the history of the system only through the current state.*

In other words, at each decision epoch, the time until the next decision and the next state of the system only depends on the current state and decision currently chosen by the decision maker. Note that there is no restriction on the distribution of the inter-decision time. Discrete-time Markov decision process (DTMDP) and continuous-time Markov decision process (CTMDP) are special cases of SMDP with fix inter-decision time and exponentially distributed inter-decision time, respectively.

Below, additional definitions on SMDP and the notational conventions used in the report are introduced.

Definition 2 *A* decision rule *of an SMDP specifies the rule for selection of actions in each state at a specified decision epoch.*

- *Deterministic decision rules are formally defined as a function* $D_t : \mathcal{I} \rightarrow \mathcal{A}$ *which specifies that the action* $D_t(i)$ *be chosen with certainty if the system is found in state* i *at the decision epoch* t, *for all* i.

- *Randomized decision rules prescribe that, when the system is found in state* i, *action* $a_\kappa \in A(i)$ *be chosen with probability* $p_i(a_\kappa, t)$, *where*

$$\sum_{a_\kappa \in A(i)} p_i(a_\kappa, t) = 1.$$

With stationary randomized policies, $p_i(a_\kappa, t)$ is simply $p_i(a_\kappa)$. correspondingly, action-dependent state transition probability from state i to j is denoted as $p_i(a_\kappa, j)$.

Definition 3 *A* decision policy *specifies the decision rule of the decision model at all decision epochs. Denote $f = \{D_0, D_1, D_2, \cdots\}$ the decision policy that apply decision D_ℓ at the ℓ^{th} decision epoch t_ℓ.*

- *A decision policy is said to be deterministic (respectively, randomized) if it applies a deterministic (respectively, randomized) decision rule at all decision epochs.*
- *A decision policy is said to be stationary if it applies the same decision rule at all decision epochs $f = \{D, D, D, \cdots\}$.*

1.3.3 Network Topology

Consider a network in a circular location space with randomly uniform distributed nodes. The single data sink is located at the center of the space, which receives all sensed data and is typically a data processing center. All nodes generate packets according to a homogeneous Poisson process. In addition, each node also routes the traffic from its neighboring nodes towards the direction of the sink. For capturing the effect of channel access conflict, the wireless channel is assumed ideal. The transmission failure or link error is assumed to occur due to collision. For tractability of the analysis, no queuing at nodes is assumed. Until the currently waiting or lost (backlogged) packet is successfully (re)transmitted, new arrivals are discarded. This traffic model will ensure high throughput but at the cost of unpredictable delay. Note that, instead, any other desired strategy could be adopted. The net new packet arrival rate at each node is assumed Poisson and denoted by λ_n. An approximate amount of relay traffic that a node handles will be computed after the traffic forwarding scheme is discussed.

1.3.4 Traffic Forwarding Scheme

A simple *random forwarding* scheme is assumed where the field nodes send data packets towards the sink by randomly selecting a forward direction neighbor as a relaying node in a multi-hop fashion. This simple forwarding approach provides a natural way of traffic load distribution as well as a tractable way to study the effect of the network multi-hop dynamics on the node control model.

Assume there are N nodes, of which $N-1$ are data sources. The nodes are denoted by IDs 0 to $N-1$, the sink node being node 0. We denote d_{xy} as the distance between nodes x and y (see Figure 1.2). $d_{0x} \triangleq d_x$ represents the distance of node x to the sink.

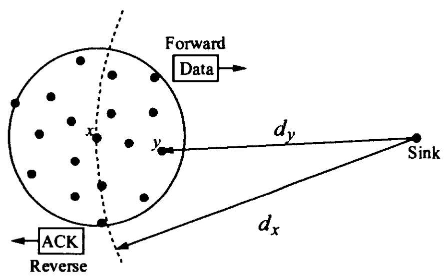

Figure 1.2 Data packets are sent towards the forward direction neighbors. ACK packets are sent towards the reverse direction neighbors.

All nodes transmit at nominal power and have isotropic radio communication range r. The neighborhood relationships are defined as follow:

(i) Node y is said to be neighbor of node x (and vise versa) if and only if $d_{xy} \leq r$;

(ii) Node y is said to be forward direction neighbor (FDN) of node x if the following conditions are satisfied:

$$d_x > r,\ d_{xy} \leq r,\ \text{and } d_y \leq d_x$$

$\mathcal{N}_{\text{FDN}}(x)$ denote the set of all FDNs of node x and $|\mathcal{N}_{\text{FDN}}(x)|$ the number of FDNs of node x.

(iii) Node x is said to be reverse direction neighbor (RDN) of node y if the following conditions are satisfied:

$$d_y > r,\ d_{yx} \leq r,\ \text{and}\ d_x > d_y$$

Similarly, $\mathcal{N}_{\text{RDN}}(y)$ denotes the set of all RDNs of node y and $|\mathcal{N}_{\text{FDN}}(x)|$ the number of RDNs of node y.

Note that (ii) and (iii) imply that directional (forward and reverse) neighborhoods are not meaningful for nodes at a distance less than r from the sink node. As defined, FDN and RDN are reciprocal, i.e.,

$$y \in \mathcal{N}_{\text{FDN}}(x) \Leftrightarrow x \in \mathcal{N}_{\text{RDN}}(y). \tag{1.1}$$

In the random forwarding scheme, neighborhood information is assumed available, and for relaying traffic a node selects the nodes in its forward direction neighborhood with equal probability. With these settings, data packets always travel in the forward direction while acknowledgment (ACK) packets are sent in the reverse direction.

The Relay Traffic The relay traffic at a node depends on the topology and multi-hop forwarding strategy. Since the node distribution in the network is uniformly random and there is only one sink, the closer a node is to the sink, the greater the volume of relay traffic. Since there is no queueing at a node and new data packet arrivals are assumed Poisson distributed, the relay data traffic at a node is also Poisson distributed, however with a distance-to-the-sink dependent rate $\lambda_r(d)$, where d is the distance of the node under consideration to the sink.

To compute $\lambda_r(d)$, the circular network space of radius R is divided into concentric coronas of width r (nodal radio range) around the sink, and consider a node, say x, at a distance d from the sink (see Figure 1.3). Let the average number of active neighbors of a node be n. (Note that the number of active neighbors is equal to the total number of alive

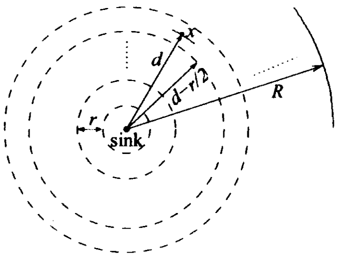

Figure 1.3 Schematic for calculating the relay traffic.

neighbors, because data arrival process at a node is independent of the state of an alive node [11].) Then, the number of nodes in the corona at distance d is $\frac{2nd}{r}$. These nodes will have to carry the new traffic load $\frac{2nd\lambda_n}{r}$ plus the traffic relayed from outside rings $\frac{n\lambda_n}{r^2}\left[R^2 - (d + r/2)^2\right]$. In the assumed random forwarding, a packet could take multiple hops to cover the distance r (nodal radio range), which increases the traffic load of an intermediate node (say, the node x) further. Denoting the near field distance of a node as d_0, the number of hops within a nodal range would vary within $[1, r/d_0]$, where it is assumed that r/d_0 is an integer, > 1. This within-range multi-hopping possibility approximately increases the locally generated traffic $\frac{2nd\lambda_n}{r}$ by a factor of $\frac{1}{2}\left(1 + \frac{r}{2d_0}\right)$ and the relay traffic by a factor of $\frac{r}{2d_0}$. Thus, a node at a distance $(d \pm r/2)$ carries approximately $\frac{r}{2nd}\left\{\frac{2nd\lambda_n}{r}\cdot\frac{1}{2}\left(1 + \frac{r}{2d_0}\right) + \frac{n\lambda_n}{r^2}\left[R^2 - (d + r/2)^2\right]\cdot\frac{r}{2d_0}\right\} =$ $\lambda_n + \frac{\lambda_n}{2}\left(\frac{r}{2d_0} - 1\right) + \frac{\lambda_n}{4dd_0}\left[R^2 - (d + r/2)^2\right]$ data traffic. Hence, effectively, the approximate relay traffic carried by a node at a distance d (and in its neighborhood) is:

$$\lambda_r(d) \approx \frac{\lambda_n}{2}\left(\frac{r}{2d_0} - 1\right) + \frac{\lambda_n}{4dd_0}\left[R^2 - (d + r/2)^2\right] \tag{1.2}$$

In the energy consumption modeling in the following Section, for simplicity of the analysis, it is assumed that the traffic relayed by the neighbors of a node is approximately the same as that relayed by the node itself. In addition, the distance dependence term in the relay traffic rate will be omitted; the term λ_r will be used, which implicitly means $\lambda_r(d)$. In Section 3, while computing numerical results, the distance dependence is taken into account to capture the energy consumption patterns of different nodes.

CHAPTER 2

POWER CONSUMPTION MODEL

With the concepts of SMDP and networking model outlined, the power consumption at different states of a node will be analyzed. The controlled stochastic system in the present case follows a randomized decision rule.

A schematic showing the decision paths at the two communicating neighbors during a transmit-receive operation is shown in Figure 2.1. The basis of decision paths, along with transition probabilities and the associated cost structure, will be elaborated in the subsequent development.

2.1 Receive State Analysis

When a node is in *receive* (*Rx*) state, three alternative events are possible: it is either receiving data, or receiving ACK, or overhearing communication from neighboring nodes. After successfully receiving a data packet, the node immediately sends an ACK packet; otherwise it simply idles. If an ACK packet is received, the node goes to *idle* state. When a node overhears a transmission intended to other, it listens to the transmission long enough to determine that the packet is not destined to itself and then switches to *sleep* state.

The state transitions described above (also see Figure 2.2) outline the decision rule followed by the virtual controller in the *Rx* state. The set of allowable actions in the *Rx* state is therefore

$$A(\mathrm{Rx}) = \{\text{“receive data”}, \text{“receive ACK”}, \text{“overhearing”}\}.$$

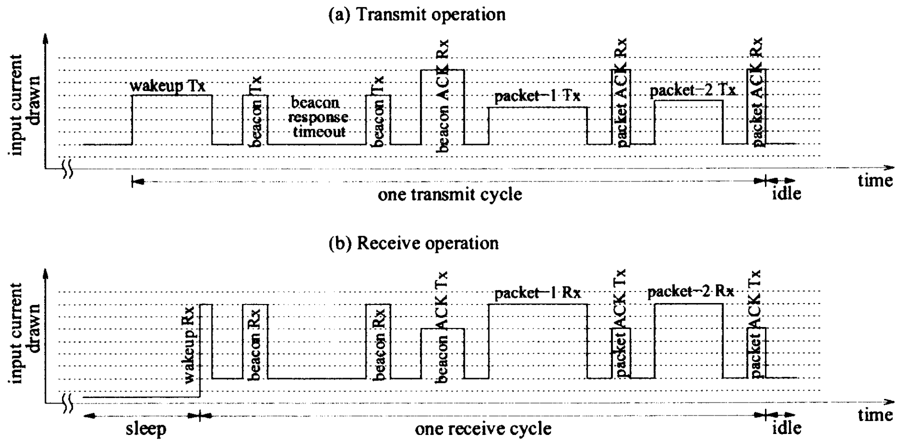

Figure 2.1 Schematic of a sample path of a node as a dynamic stochastic system. Each state determines certain energy consumption. The first beacon from the transmitter went unanswered. After successful beaconing and slot alignment, two packets of different size are transferred with varying transmit power.

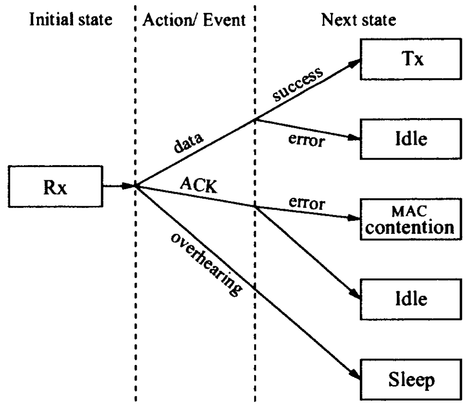

Figure 2.2 Decision rule in *receive* (*Rx*) state.

2.1.1 Action/Event Selection Probabilities

In this section conditional probabilities of "receiving data", "receiving ACK", and "overhearing" are computed, given that the node is in Rx state. These probabilities depend on the node's neighborhood density (average number of local neighbors), traffic load, and the network's multi-hop communication strategy.

Referring to Figure 1.2, assume that all nodes have the same average number of forward direction neighbors $|\mathcal{N}_{\text{FDN}}(y)| = |\mathcal{N}_{\text{FDN}}(x)| = n_1$ and reverse direction neighbors

$|\mathcal{N}_{\text{RDN}}(y)| = |\mathcal{N}_{\text{RDN}}(x)| = n_2$. Thus, the average number of local neighbors of a node, $|N(y)| = |N(x)| = n = n_1 + n_2$. By (1.1), the node y is a potential relay of x if $y \in \mathcal{N}_{\text{FDN}}(x)$ (or equivalently if $x \in \mathcal{N}_{\text{RDN}}(y)$). Consequently, $\mathcal{N}_{\text{RDN}}(x)$ represents the set of n_2 nodes that may chose node y as a relaying node. Also, since in random forwarding each node requesting a relay service evenly balances its traffic among its n_1 FDNs, only $1/n_1$ fraction of the forward direction traffic of node x is destined to y. Thus, a node picks up on average per unit time:

- Transmissions from its n_2 RDN
  - $n_2(\lambda_n + \lambda_r)$ data, of which $\frac{n_2}{n_1}(\lambda_n + \lambda_r)$ are destined to x,
  - $n_2\lambda_r$ ACK, of which none is destined to x.
- Transmissions from its n_1 FDN
  - $n_1(\lambda_n + \lambda_r)$ data, of which none is destined to x,
  - $n_1\lambda_r$ ACK, of which $\frac{n_1}{n_2}\lambda_r$ is destined to x.

From the above observations, the probabilities of receiving a data packet, an ACK, or an overheard packet at a node, while it is in Rx state, can be obtained respectively as:

$$p_{rx}(\text{data}) = \frac{n_2\left(\lambda_n + \lambda_r\right)}{n_1\left(n_1 + n_2\right)\left(\lambda_n + 2\lambda_r\right)} \tag{2.1}$$

$$p_{rx}(\text{ack}) = \frac{n_1\lambda_n}{n_2\left(n_1 + n_2\right)\left(\lambda_n + 2\lambda_r\right)} \tag{2.2}$$

$$p_{rx}(\text{oh}) = 1 - \frac{{n_2}^2\lambda_n + \left({n_1}^2 + {n_2}^2\right)\lambda_r}{{n_1}^2{n_2}^2\left(n_1 + n_2\right)\left(\lambda_n + 2\lambda_r\right)} \tag{2.3}$$

For further simplification let $n_1 = n_2$, i.e., assume symmetricity of forwarding directions, which is approximately the case when the distance to the sink is large and the nodal transmission range $r \ll R$ (see Figure 1.2). This gives the simplified probability expressions:

$$p_{rx}(\text{data}) = \frac{\lambda_n + \lambda_r}{n\,(\lambda_n + 2\lambda_r)} \tag{2.4}$$

$$p_{rx}(\text{ack}) = \frac{\lambda_r}{n\,(\lambda_n + 2\lambda_r)} \tag{2.5}$$

$$p_{rx}(\text{oh}) = 1 - \frac{1}{n} \tag{2.6}$$

where $n = n_1 + n_2$ is the average number of local neighbors. Note that (2.6) verifies the intuition that with symmetricity assumption and random forwarding strategy, the intended transmission from a node (say node y) to a neighbor (say node x) in its Rx state is with probability $1/n$.

2.1.2 Sojourn Times

To determine the sojourn time distribution in the *Rx* state, recall that the SMDP model allows the time spent in a particular state to have a general distribution, and can depend on the action chosen by the decision maker (virtual controller) as well as the next state of the system. In capturing the average cost (energy spent) per unit time, the analysis is particularly interested in the averages of sojourn times.

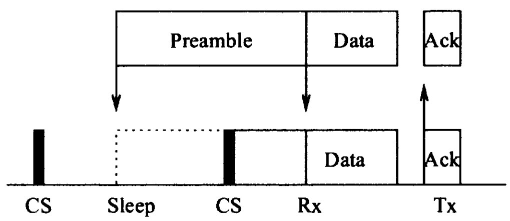

Figure 2.3 Data transmit and receive process

Receiving data: For successfully receiving a data packet, a node has to remain in *Rx* state during the time necessary to receive the entire packet. Fixed-sized packet are considered with length L_{data} from all nodes. As depicted in Figure 2.3, it is also assumed that every data packet reception is preceded by a part or whole of a preamble (for waking up the receiving node). It follows that the sojourn time in *Rx* state while successfully receiving data is uniformly distributed between $\left(\frac{L_{data}}{C}\right)$ and $\left(\frac{L_{data}+L_{preamble}}{C}\right)$, where $l_{preamble}$ is the fix length of the preamble and C the data rate of the wireless channel. Recalling that every successful data packet reception is followed by an ACK transmission (i.e., the next state is *Tx*), and denoting by $E_i(a_\kappa, j)$ the average sojourn time in state i if the action a_κ is taken in the present and the next state of the system is j, it follows that

$$E_{rx}(\text{data}, \mathit{Tx}) = \frac{L_{data} + L_{preamble}/2}{C}. \tag{2.7}$$

Also, since the data packet error is assumed due forwarding inability or channel access conflict, and noting that the error checksum is appended at the trailer of the data packet, the sojourn time in *Rx* state while a data packet reception fails is uniformly distributed between 0 and $\left(\frac{L_{data}+L_{preamble}}{C}\right)$. Thus,

$$E_{rx}(\text{data}, \mathit{idle}) = \frac{L_{data} + L_{preamble}}{2C}. \tag{2.8}$$

Receiving ACK: ACK packets are in general of smaller size and are sent without preamble. It is also assumed that every data packet is acknowledged separately. Piggybacking is not used because, in the type of network under consideration, data and ACK packets follow opposite directions. The distribution of sojourn time in *Rx* state when ACK is being received is therefore the time needed to successfully or unsuccessfully receive the ACK packet, which are respectively given by

$$E_{rx}(\text{ack}, \mathit{idle}) = \frac{L_{ack}}{C}, \tag{2.9}$$

for successfully receiving an ACK, and

$$E_{rx}(\text{ack}, \mathit{MAC}) = \frac{L_{ack}}{2C}, \tag{2.10}$$

in case of failure to successfully receive the ACK, where L_{ack} is the length of an ACK packet.

Overhearing: It occurs when a node is engaged in receiving transmission that is not intended for itself. A node overhears for as long as it needs, to determine that it is not the intended recipient. It is assumed that the time necessary for a node to detect overhearing is at least the time to receive the header of the data or ACK packet (which could also be RTS or CTS packets, if they are used). Let L_h the length of the header of both data and ACK packets. In case the overheard transmission is a data packet, preamble time also needs to be taken into account. Particularly, the sojourn time is exactly $\frac{L_h}{C}$ when the overheard transmission is an ACK, and it is uniformly distributed between $\left(\frac{L_h}{C}\right)$ and $\left(\frac{L_h+L_{preamble}}{C}\right)$ if the overheard transmission is a data packet. Following the description in Section 2.1.1, it can be shown that in unit time a node overhears

- $\left(n_1 + n_2 - \frac{n_2}{n_1}\right)(\lambda_n + \lambda_r)$ data packet transmissions, and
- $\left(n_1 + n_2 - \frac{n_1}{n_2}\right)\lambda_r$ ACK packet transmissions.

Therefore, the conditional probabilities of overhearing data and ACK packets are respectively given by

$$\Pr\{\text{data}|\text{oh}\} = \frac{\left(n_1 + n_2 - \frac{n_2}{n_1}\right)(\lambda_n + \lambda_r)}{\left(n_1 + n_2 - \frac{n_2}{n_1}\right)(\lambda_n + \lambda_r) + \left(n_1 + n_2 - \frac{n_1}{n_2}\right)\lambda_r} \tag{2.11a}$$

$$\Pr\{\text{ack}|\text{oh}\} = \frac{\left(n_1 + n_2 - \frac{n_1}{n_2}\right)\lambda_r}{\left(n_1 + n_2 - \frac{n_2}{n_1}\right)(\lambda_n + \lambda_r) + \left(n_1 + n_2 - \frac{n_1}{n_2}\right)\lambda_r} \tag{2.11b}$$

The average sojourn time would then be obtained as:

$$E_{rx}(\text{oh}, \textit{sleep}) =$$

$$\frac{L_h}{C} \cdot \Pr\{\text{ack}|\text{oh}\} + \frac{L_h+L_{preamble}/2}{C} \cdot \Pr\{\text{data}|\text{oh}\}. \tag{2.12}$$

Making the same simplifying assumption as in (2.4)-(2.6) (i.e., by letting $n_1 = n_2$) It follows from (2.11):

$$\Pr\{\text{data}|\text{oh}\} = \frac{\lambda_n + \lambda_r}{\lambda_n + 2\lambda_r} \tag{2.13a}$$

$$\Pr\{\text{ack}|\text{oh}\} = \frac{\lambda_r}{\lambda_n + 2\lambda_r} \tag{2.13b}$$

which gives the average sojourn time

$$E_{rx}(\text{oh}, \mathit{sleep}) = \frac{\lambda_n + \lambda_r}{\lambda_n + 2\lambda_r}\frac{L_h}{C} + \frac{\lambda_r}{\lambda_n + 2\lambda_r}\frac{L_h + L_{preamble}/2}{C} \tag{2.14}$$

2.1.3 Transition Probabilities

Referring to Figure 2.2 the transition probabilities from *Rx* state to other state of the node model are computed.

Data packets: After successful reception of data packet, nodes switch to Tx state to transmit ACK packet. Consequently, the probability of going to *Tx* after data packet reception correspond to the probability the the packet is received successfully. To capture only the MAC related power consumption, it is assumed packet reception fails only in case of collision.

Consider node x is receiving data from node y. For simplification, further consider that only nodes in the neighborhood of node x but not in that of node y can cause a collision with the packet from node y. In the worst case the number of interfering node is

$$\max\{|N(x)| - |N(x) \cap N(y)|\} = \frac{n}{\pi}\left(\frac{2\pi + 3\sqrt{3}}{6}\right)$$

There transition probability into Tx follows

$$p_{rx}(\text{data}, \mathit{Tx}) = 1 - p_{collision}^{data}$$

where

$$p_{collision}^{data} = e^{\left\{-\frac{n}{\pi}\left(\frac{2\pi+3\sqrt{3}}{6}\right)(\lambda_n+\lambda_r)E_{rx}(\text{data}, \mathit{Tx})\right\}} \tag{2.15}$$

and $E_{rx}(\text{data}, \mathit{Tx})$ is the time spent by node x to receive the entire data packet including preamble. Correspondingly, the probability of going into *idle* state is

$$p_{rx}(\text{data}, \mathit{idle}) = p_{collision}^{data} \tag{2.16}$$

ACK packets: First note that a node expecting/receiving ACK packet has been a data packet transmitter at the previous state. If the ACK packet is received successfully the node simply idles. However, unsuccessful/missing ACK represent the failure of the previously sent data packet. The node switches to MAC contention mode for retransmission.

The same analysis made above to compute the probability of collision for data packets can be made to compute the probability of collision of ACK packets.

$$p_{collision}^{ack} = e^{\left\{-\frac{n}{\pi}\left(\frac{2\pi+3\sqrt{3}}{6}\right)(\lambda_n+\lambda_r)E_{rx}(\text{ack}, \mathit{MAC})\right\}} \tag{2.17}$$

where $E_{rx}(\text{ack}, \mathit{MAC})$ is the time duration of an ACK packet. The probability of switching to *MAC contention* and the probability of switching to *idle* are given by

$$p_{rx}(\text{ack}, \mathit{MAC}) = p_{collision}^{ack} \tag{2.18}$$

$$p_{rx}(\text{ack}, \mathit{idle}) = 1 - p_{collision}^{ack} \tag{2.19}$$

Overhearing: The outcome of transmission overhearing is invariably switching to *sleep* state, i.e.,

$$p_{rx}(\text{oh}, \mathit{sleep}) = 1. \tag{2.20}$$

2.2 Idle State Analysis

Idle listening corresponds to the state in which a node listens actively for potential transmit or receive packets, and is a major source unnecessary energy consumption in sensor networks. To reduce energy consumption due to idle listening, timeout policy is often used [6] to automatically switch the controlled system into lower power consuming state.

In *idle* state, the events and the corresponding allowable actions of the decision maker are (see also Figure 2.4):

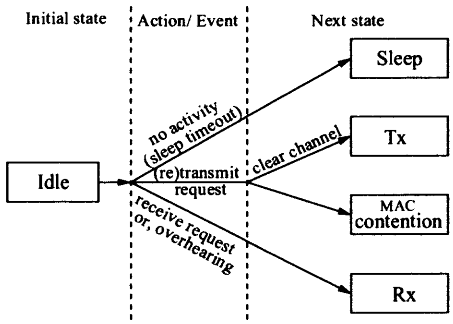

Figure 2.4 Decision rule in *idle* state.

- no activity until a timeout threshold t_{th} – goes to *sleep*,
- interruption of idling period because of new or backlogged data packet to be sent – goes to *Tx* or MAC contention, depending on the channel condition, and
- interruption of idling period because of data packet to be received – goes to *Rx*.

Below, the probability of each of these events, corresponding sojourn times, and associated costs are computed.

2.2.1 Action/Event Selection Probabilities

First note that when a node x is in idle state, all transmissions and receptions from or to node x concern data packets, and no ACK packets are involved. The reason is that whenever an ACK packet has to be transmitted or received, the node will either switch directly from

Tx to *Rx* mode (to receive the ACK) or from *Rx* to *Tx* mode (to transmit the ACK). In both cases no transition through the *idle* state is needed. In other words, when idling, a node is not required to transmit or receive an ACK packet. However, an idling node can overhear ACK packets from their neighbors. Thus, the idling period of node x can only be interrupted due to (i) data packets to be transmitted by the node itself, (ii) data packets to be received or overheard from neighboring nodes, and (iii) overhearing ACKs. It follows that in the idling phase, the total rate of transmit attempt would be $\lambda_n + \lambda_r$, and the total rate of receive/ovehear attempt would be $n(\lambda_n + \lambda_r) + n\lambda_r - (\lambda_n + \lambda_r) = (n-1)\lambda_n + (2n-1)\lambda_r$, where $n(\lambda_n + \lambda_r)$ represents the total number of data packets sent by n neighbors of node x and $n\lambda_r$ is the total number of ACK sent of which $(\lambda_n + \lambda_r)$ represents the portion destined to node x.

Therefore, sleep timeout probability, i.e., the probability that there is no transmit, receive, or overhearing activity until timeout t_{th} occurs is given by

$$p_{idle}(\text{sleep timeout}) = e^{-n(\lambda_n+2\lambda_r)t_{th}}. \tag{2.21}$$

Correspondingly, transmit request probability, i.e., the probability of the idling period being interrupted by a transmit or retransmit request is given by

$$p_{idle}(\text{tx request}) = \left(1 - e^{-n(\lambda_n+2\lambda_r)t_{th}}\right) \frac{\lambda_n + \lambda_r}{n\left(\lambda_n + 2\lambda_r\right)} \tag{2.22}$$

where $\left(1 - e^{-n(\lambda_n+2\lambda_r)t_{th}}\right)$ is the probability that at least one activity (transmit or receive request or overhearing) occurs before timeout and $\frac{\lambda_n+\lambda_r}{n(\lambda_n+2\lambda_r)}$ represents the probability that a data transmit request occurs before receive request, or overhearing.

A receive request or overhearing probability during the idle period is

$$p_{idle}(\text{rx request or oh}) = \left(1 - e^{-(\lambda_n+2\lambda_r)t_{th}}\right) \frac{(n-1)\,\lambda_n + (2n-1)\,\lambda_r}{n\left(\lambda_n + 2\lambda_r\right)} \tag{2.23}$$

2.2.2 Sojourn Times

As in the other states, the sojourn time in the idle state depends on the actions chosen by the decision maker.

Sleep timeout: In the event of sleep timeout, the node stays in idle mode for the time t_{th}. Therefore,

$$E_{idle}(\text{no activity}, \textit{sleep}) = t_{th} \tag{2.24}$$

Transmit/retransmit request: If the idle period is interrupted by a (re)transmit request for a new or backlogged packet, there could be two possible outcomes: going to *Tx* state or MAC contention (or carrier sense (CS)) state. Since the (re)transmit requests arrival is a Poisson process with parameter $\lambda_n + \lambda_r$, the interarrival time is exponentially distributed, however truncated at the upper limit of t_{th}. The average sojourn time is therefore

$$\begin{aligned} E_{idle}(\text{tx request}, \textit{Tx}) &= E_{idle}(\text{tx request}, \textit{cs}) \\ &= \frac{1}{\lambda_n + \lambda_r} - e^{-(\lambda_n+\lambda_r)t_{th}}\left(t_{th} + \frac{1}{\lambda_n + \lambda_r}\right) \end{aligned} \tag{2.25}$$

Receive request or overhearing: Since λ_n and λ_r are Poisson distributed, the interarrival times of receive requests and overhear signals are also exponentially distributed with parameter $(n-1)\lambda_n + (2n-1)\lambda_r \triangleq \lambda_{rx}$, however truncated at the upper limit of t_{th}. The corresponding average sojourn time is given by

$$E_{idle}(\text{rx request}, \textit{Rx}) = \frac{1}{\lambda_{rx}} - e^{-\lambda_{rx}t_{th}}\left(t_{th} + \frac{1}{\lambda_{rx}}\right) \tag{2.26}$$

2.2.3 Transition Probabilities

This section computes the transition probabilities out of *idle* state. Referring to Figure 2.4, the transition probabilities in the event of sleep timeout or receive request are straight forward:

$$p_{idle}(\text{sleep timeout}, \textit{sleep}) = 1 \tag{2.27}$$

$$p_{idle}\left(\text{rx request}, Rx\right) = 1 \tag{2.28}$$

When a node in *idle* state needs to transmit its own data, the wireless channel is first sampled according to the MAC protocol and the node only proceeds to *Tx* state if the channel is found idle. Bellow, the probability of finding the channel idle is evaluated.

Since all data packets are of the same size, it takes a given amount of time to complete the full cycle to sending data packet including preamble, actual data packet and subsequent (eventual) ACK (see Figure 2.3). In the worst case, every transmission attempt from a node causes the wireless channel to be unavailable for other neighboring nodes for a time $t_{TxCycle} = \frac{L_{preamble}+L_{data}+L_{ACK}}{C}$. The probability for a node (say x) to find the the channel idle is the probability that none the n neighboring nodes of node x initiated transmission in preceding $t_{TxCycle}$ interval of time. It follows that

$$p_{idle}\left(\text{tx request}, Tx\right) = e^{-n(\lambda_n+\lambda_r)t_{TxCycle}} \tag{2.29}$$

$$p_{idle}\left(\text{tx request}, Tx\right) = 1 - e^{-n(\lambda_n+\lambda_r)t_{TxCycle}} \tag{2.30}$$

2.3 Transmit State Analysis

When a node is in *transmit* (*Tx*) mode, two alternative events are possible: the node is transmitting either data, or ACK. After successfully sending a data packet, the transmitter immediately switches to *Rx* mode to receive an ACK packet. If the data packet transmission is in error (i.e., if no ACK is received), the transmitter goes to *idle* state from *Tx* state. At the receiver end, after receiving the packet successfully, it goes to *Tx* state to send an ACK packet and then becomes *idle*. The state transition rules are shown in Figure 2.5.

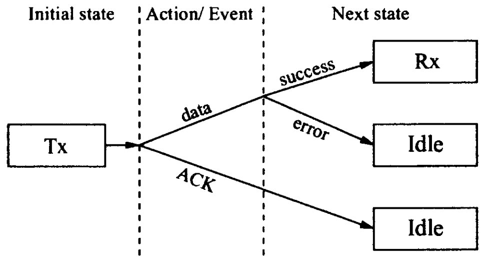

Figure 2.5 Decision rule in *transmit* (*Tx*) state.

Note that, even if the data packet is received successfully at the receiver, it could be retransmitted due to ACK failure (handled by the *Rx* state, shown in Figure 2.2). On the other hand, if the data is not received successfully at the receiver, the data packet is backlogged at the transmitter and from the idle state it goes through the MAC contention phase for its retransmission attempt (as depicted in Figure 2.4).

2.3.1 Action/Event Selection Probabilities

In the transmit state, the net data packet arrival rate is $\lambda_n + \lambda_r$ and the ACK arrival rate is λ_r. Therefore, the event selection probabilities are, respectively,

$$p_{tx}(\text{data}) = \frac{\lambda_n + \lambda_r}{\lambda_n + 2\lambda_r} \tag{2.31}$$

$$p_{tx}(\text{ack}) = \frac{\lambda_r}{\lambda_n + 2\lambda_r}. \tag{2.32}$$

2.3.2 Transition Probabilities

Here the same analysis as in the case of Rx is made. When Data packets are involved, all packet collision assumed to happen only on receiver end because nodes in the neighborhood of the transmitter refrain from transmitting if the sense any activity on the channel. The probability of packet failure due to collision is the same as in equation (2.15).

$$p_{tx}(\text{data}, Rx) = 1 - p^{data}_{collision} \tag{2.33}$$

$$p_{tx}(\text{data}, idle) = 1 - p^{data}_{collision} \tag{2.34}$$

$$p_{tx}(\text{ack}, idle) = 1 \tag{2.35}$$

2.3.3 Sojourn Times

As in the *Rx* state, following the similar arguments for sojourn time evaluation,

$$E_{tx}(\text{data}, Rx) = \frac{L_{data} + L_{preamble}}{C} \tag{2.36}$$

$$E_{tx}(\text{data}, idle) = \frac{L_{data} + L_{preamble} + L_{ack}}{C} \tag{2.37}$$

$$E_{tx}(\text{ack}, idle) = \frac{L_{ack}}{C}. \tag{2.38}$$

2.4 Sleep State Analysis

In the *sleep* state the radio transceiver of the sensor node is switched off to avoid unnecessary idle listening and minimize energy consumption. During the sleep state, if a transmit request arrives at its own buffer, the node immediately wakes up to transmit the data packet. Also, when without any transmit request, a sleeping node periodically wakes up and samples the wireless channel for activity to maintain the network connectivity. If the node detects any activity it switches to *Rx* mode. Otherwise, it goes to the next sleep cycle. A fixed sleeping length t_{sleep} is considered for all nodes. However, the occurrence of sleep state of a node is independent of the others', so the probability of all nodes in the same neighborhood being in sleep mode is negligible. Thus, in sleep state of a node three alternative events can occur (see Figure 2.6).

- No activity: the sleeping period finishes without any transmit or receive request,
- Transmit request: immediate wakeup if a new packet arrives in node's own buffer during the sleeping period, and

- Receive request: Wake up after the current sleep period if a receive request is made from a from neighboring node.

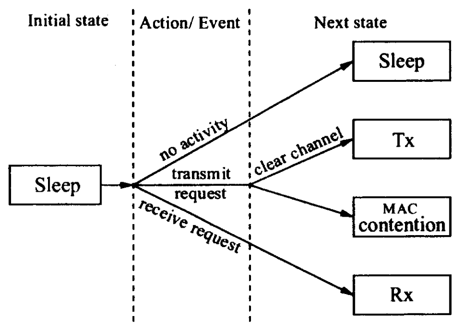

Figure 2.6 Decision rule in *sleep* state.

Here, as in the previous cases, although the above events are only known to the real node at end of the current sleep cycle when the node samples the channel or checks its own buffer, it is assumed that the virtual node controller possesses a-priori knowledge of the events and select the appropriate actions just as soon as the node enters the current sleeping cycle. Since the probability with which the virtual controller selects the different actions equals the probability of occurrence of the corresponding events, the performance of the SMDP is the same as that of the real node.

2.4.1 Action/Event Selection Probabilities

The probability of no activity in all n neighbors of the sleeping node and no new packet arrival at the node's own buffer during a sleep period t_{sleep} is given by

$$p_{sleep}(\text{no activity}) = e^{-(n+1)(\lambda_n+\lambda_r)t_{sleep}}. \tag{2.39}$$

A transmit request arrives at its own buffer with probability

$$p_{sleep}(\text{tx request}) = 1 - e^{-(\lambda_n+\lambda_r)t_{sleep}}. \tag{2.40}$$

Likewise, the probability of a sleeping node receiving a relay request from one of its neighbors, but no new packet arrived at its own buffer, is given by

$$p_{sleep}(\text{rx request}) = e^{-(\lambda_n+\lambda_r)t_{sleep}}\left(1 - e^{-n(\lambda_n+\lambda_r)t_{sleep}}\right). \tag{2.41}$$

2.4.2 Sojourn Times

Since a receive request is attended only after the current sleep period, the sojourn time due to receive request is the same as that due to no activity, which is

$$E_{sleep}(\text{rx request}, Rx) = E_{sleep}(\text{no activity}, sleep) = t_{sleep}. \tag{2.42}$$

A transmit request originated at the sleeping node leads to two possibilities: going to the *Tx* state (with clear channel) or the MAC contention (or CS) state (with channel busy). Since in either case sojourn time is the same, as in case of idle state:

$$\begin{aligned} E_{sleep}(\text{tx request}, Tx) &= E_{sleep}(\text{tx request}, CS) \\ &= \tfrac{1}{\lambda_n+\lambda_r} - e^{-(\lambda_n+\lambda_r)t_{sleep}}\left(t_{sleep} + \tfrac{1}{\lambda_n+\lambda_r}\right). \end{aligned} \tag{2.43}$$

The transitions probabilities in *sleep* state are similar to that of *Idle* state (compare Figure 2.6 and Figure 2.4).

2.5 MAC Contention

MAC contention resolution is used to make efficient use of the shared wireless medium among nodes in the same vicinity. In the considered MAC scheme, a node samples the wireless channel whenever it desires to transmit data, and backs-off for a random time if the channel is not idle, otherwise the node transmits its data (see Figure 2.7). For simplicity it is assumed that the back-off time is continuously uniformly distributed between 0 and $t_{back-off}$. In the decision model, MAC contention (or physical carrier sensing, CS) corresponds

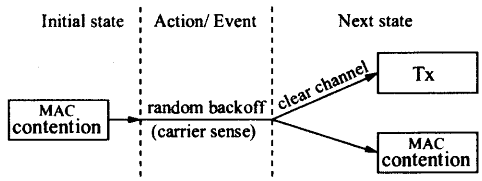

Figure 2.7 Decision rule in MAC contention state.

to a single action *back-off* selected with probability 1 whenever the node is this state.

$$p_{CS}(\text{back-off}) = 1 \tag{2.44}$$

A the end of each *back-off* period, the node samples the wireless channel again and transmits its data if the channel is found idle, otherwise it backs-off for a random time.

The transition probabilities for *MAC contention* can be derived from that of *idle* state (compare Figure 2.7 and Figure 2.4).

2.6 Cost Structure

To study the power consumption performance of a sensor node as a controlled stochastic system, it is associated with the SMDP model a cost structure in which every pair of allowable state action/event pair (i, a_κ) $i \in \mathcal{I}$, $a_\kappa \in A(i)$ is associated with a cost rate $\mathcal{C}_i\,(a_\kappa)$ at which costs are incurred of the sensor node is i and action/event a_κ is chosen. Typically the cost structure associated with a SMDP defines the nature and the complexity of the underlying optimization or evaluation problem. In general the cost incurred in a state is made of a fixed lump sum state dependent cost, e.g. system initialization or switching cost, and a state and action (possibly next state) dependent cost rate.

In our sensor node model, as energy consumption is the focus, costs are represented in each state by communication related power consumption level of the sensor node. The power consumption performance of the protocol governing the behavior of the node is then computed as long run expected accumulated cost per unit time. Formally the long run average cost under a policy f is defined as

$$G^f = \lim_{T \to \infty} E\left(\frac{C(T)}{T}\right),$$

where $E(\cdot)$ is the expectation operation and $C(T)$ is the accumulated cost over a period of time T. It has been shown in [12] that the long-run average cost can be compute as

$$G^f = \frac{\sum_{i \in \mathcal{I}} \sum_{a_\kappa \in A(i)} \hat{\pi}_i p_i(a_\kappa) c_i(a_\kappa)}{\sum_{i \in \mathcal{I}} \sum_{a_\kappa \in A(i)} \hat{\pi}_i p_i(a_\kappa) E_i(a_\kappa)}.$$

Referring to Section 1.3.2, state transition probability of the embedded Markov process $\{X_\ell\}$ is $P_{ij} = \sum_{a_\kappa \in A(i)} p_i(a_\kappa) p_i(a_\kappa, j)$. The state probability $\hat{\pi}_i$ is computed using $\hat{\pi}\,[P_{ij}] = \hat{\pi}$, where $\sum_i \hat{\pi}_i = 1$.

CHAPTER 3

NUMERICAL RESULTS

This chapter presents an application of the proposed power consumption model and gives results obtained through numerical computation. The proposed model of sensor nodes as controlled stochastic dynamic systems is applied to study the effect on communication related power consumption of node sleep cycle. As protocol designers, communication scientists and engineers often face the optimal design and control problems. The goal of the protocol designers is to find a fixed optimal value of system parameter, while communication scientists and engineers search for strategies to dynamically control the system parameter in order to achieve optimal system operation. In both cases, an in-depth understanding of the system behavior under complex protocols is needed.

The Effect on Sleep Time on Power Consumption

To illustrate how the power consumption model can be used to gain in-depth understanding of the complex behavior of sensor nodes under the combined actions of MAC protocols and power management policies, a study of the effect of sleeping period on the node level long run power consumption was made under low and high traffic load. In addition, based on the traffic model and assumptions from Shapter 1.3, an energy consumption map of the sensor field at was obtained for low and high traffic loads. In all numerical studies, network radius considered is $R = 100$, and the sink is located at the center. Nodal radio range is $r = 10$, and the near field distance $d_0 = 1$. Node density (i.e., average number of neighbors n of a node) and new traffic generation rate λ_n is varied. λ_n in turn modifies the relay traffic $\lambda_r(d)$.

Figure 3.1(a) shows both the power consumption and throughput performance of a node x at fixed distance ($d = 95$) from the sink. As all nodes in the same neighborhood sleep longer, the packet collision probability drops and the throughput of node increases which in turn increases the overall energy consumption due to transmission. Figure 3.1(b)

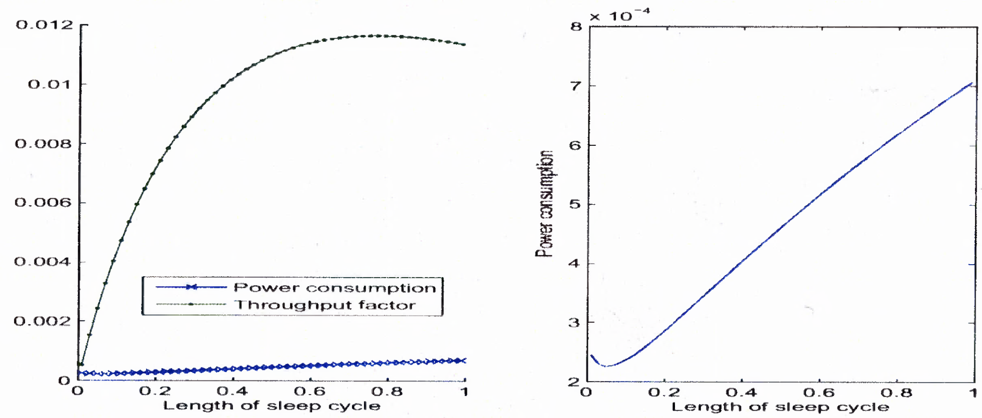

Figure 3.1 Long run average power consumption and throughput performance of a node at $d = 95$ from the sink. $\lambda_n = 0.01$.

shows the energy consumption per node as the length of sleep cycle is increased. The results are counter intuitive: the long run average power consumption increases as a node sleeps longer in each sleep cycle. To have an insight, a look needs to be taken on the node's overall throughput performance. Because nodes have no queue, it is important in this case to compare the node's power consumption performance to it's utilization. A throughput performance factor is therefor introduced as the product of the probability that a node is in transmit state, the probability that the node is transmitting data, and the probability that the data is successfully received at the receiver end. As defined, the throughput performance factor represents the effective utilization of the sensor node, i.e., the fraction of time the node spends successfully sending data packet.

The previous observation is however only holds true for low traffic load. Refer to Figure 3.2 where energy and throughput performances are shown for a node a distance $d = 40$ to the sink. Since the traffic load is high in the neighborhood of node y, collision is more likely and nodes (mostly backlogged) will spend more time in lower consuming state (sleep and MAC contention). As a consequence, an increase in sleep time results in a decrease in energy consumption, at the cost of lowered throughput performance.

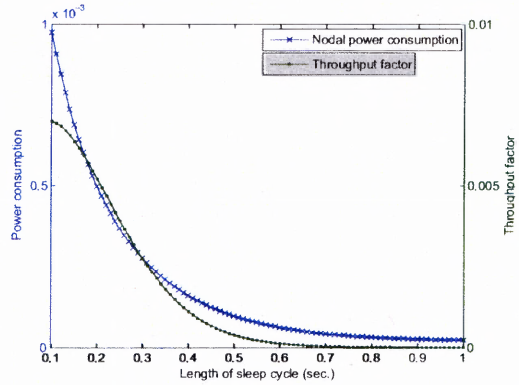

Figure 3.2 Effect of sleep cycle on nodal energy consumption and throughput at ($d = 40$. $\lambda_n = 0.01$).

Existence of an Undesirable Operational Region

The above observation suggests the existence of two regions of operating conditions for the sensor nodes:

- one desirable region in which an increase in traffic load results in an increase of energy consumption, and
- one undesirable region in which sensor nodes in which an increase in traffic load may result in a decrease in overall energy consumption because nodes, being overwhelmed by traffic, block most of the incoming packets.

Although the analysis consider nodes with no buffer, the behavior of nodes in a practical implementation will be similar to nodes behavior in the undesirable region if nodes are overwhelmed beyond there storage capacity. In this case, most part of incoming traffic is blocked and nodes spend most of the time in contention mode. Because energy consumption in contention state is less than that required for packet transmission and reception, there is an apparent net decrease in long run energy consumption. However, because throughput also decreases at the same time, most of the energy is wasted resolving contention.

Avoiding the Undesirable Region

The numerical results show that undesirable operational region can be avoided under given traffic pattern by selecting node density and system parameters as to keep collision probability low ($\ll 1.0$) for all nodes at all time. This condition is met in Figure 3.3 where the

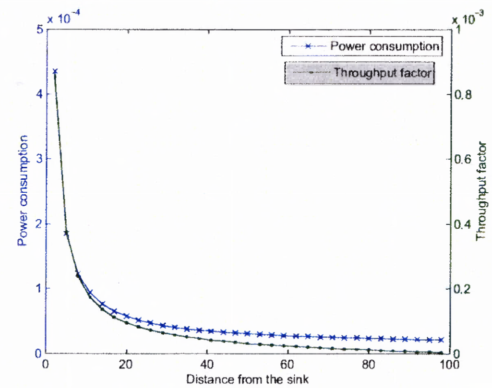

Figure 3.3 Energy consumption map of the sensor field with low traffic load. $\lambda_n = 0.0001$, $n = 15$.

nodes, when subject to very low traffic load, consume energy that mostly follow the traffic pattern. However, Figure 3.4 shows a network scenario where nodes around the sink are

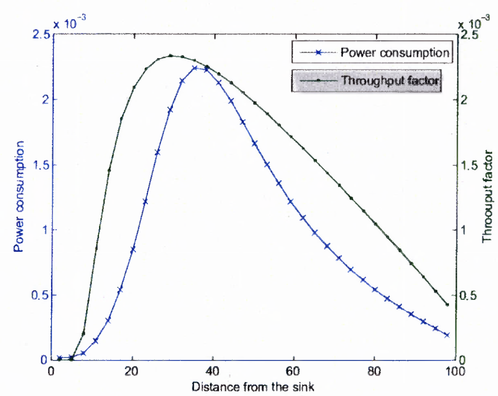

Figure 3.4 Power consumption map of the sensor field with high traffic load. $\lambda_n = 0.01$, $n = 15$.

overwhelmed with traffic and operate in the undesirable region. Figure 3.3 also indicates that for a given single sink location, a suitable distance-dependent node density could be derived that would enable all nodes in the network spend the same amount of energy.

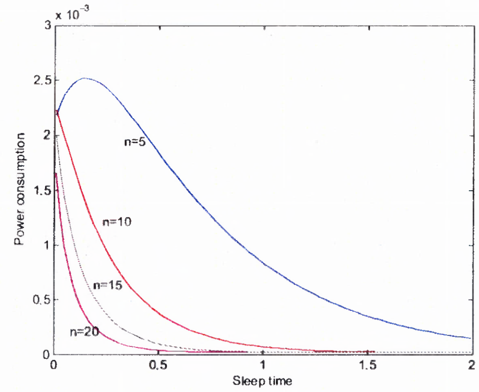

Figure 3.5 Power consumption at various sleep time. $\lambda_n = 0.1$, $d = 40$.

The plots in Figure 3.5 show power consumption behavior with sleep time at different node densities. Since the new traffic arrival rate is kept at a constant high ($\lambda_n = 0.1$), at very low sleep time, low node density cannot handle the traffic properly and shows the undesired convex region as in Figure 3.4. This indicates again that, for a desirable network behavior, there is an allowable traffic load with a given uniformly random node density.

CHAPTER 4

CONCLUSION

Power efficiency being a crucial requirement for adequate operation of battery-powered sensor networks, accurate models are needed to evaluate power consumption performance of communication protocols for wireless sensor networks. An analytic model for capturing the power consumption of a peer-to-peer communicating sensor node under a given communication protocol was developed. A sensor node was modeled as a controlled stochastic dynamic system, and the model was formulated as a semi-Markov decision process, wherein a node's states, sojourn times, and the transition probabilities are controlled by a virtual node controller, that is specific to a given communication protocol set. The energy efficiency of a protocol was measured as a long-run average cost per unit time.

The developed model assumed a preamble sampling based low power communication protocol as an underlying MAC protocol and a random forwarding strategy that randomly selects a forward direction node for multi-hop forwarding a packet. The numerical results demonstrated that the model can be effectively used as a tool for optimizing different protocol parameters to achieve energy efficient communication goals.

Although constraining assumptions on the traffic buffering model have been used for the sake of analytical tractability, the proposed methodology can be extended to more practical traffic models and will be useful in power-aware sensor networks design without requiring to conduct extensive field studies and simulations. With minor modifications, the model can also be used for comparing performances of different communication protocols in sensor networks and can provide guidelines to determining energy consumption pattern for a given node and sink deployment strategy and vice versa.

Future extensions of this work include incorporating other effective collision avoidance strategies (e.g., RTS-CTS-like hand shaking [1]), finite buffering assumption at nodes, other forwarding strategies (e.g., directed diffusion [16]), and carrying out discrete-event simulations with more realistic network and traffic assumptions.

REFERENCES

[1] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proc. IEEE INFOCOM*, New York, NY, June 2002.

[2] J. Hill and D. Culler, "MICA: a wireless platform for deeply embedded networks," *IEEE Micro*, vol. 22, no. 6, pp. 12–24, Nov./Dec. 2002.

[3] P. Basu and J. Redi, "Effect of overhearing transmissions on energy efficiency in dense sensor networks," in *Proc. ACM IPSN*, Berkeley, CA, Apr. 2004.

[4] *http://www.chipcon.com/files/CC2420_Data_Sheet_1_2.pdf*, June 2004.

[5] T.-C. Hou and V. O. K. Li, "Transmission range control in multihop packet radio networks," *IEEE Trans. Commun.*, vol. 34, no. 1, pp. 38–44, Jan. 1986.

[6] T. Simunic, L. Benini, P. Glynn, G. De Micheli, "Dynamic power management for portable systems," in *Proc. ACM MOBICOM*, Boston, MA, Aug. 2000.

[7] A. El-Hoiydi, "Aloha with preamble sampling for sporadic traffic in ad hoc wireless sensor networks, in *Proc. IEEE ICC*, Apr. 2002.

[8] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proc. ACM SenSys*, Baltimore, MD, Nov. 2004.

[9] Q. Qiu, Q. Wu, and M. Pedram, "Stochastic modeling of a power-managed system – Construction and optimization," *IEEE Trans. Computer Aided Design of Integrated Circuits and Systems*, vol. 20, no. 10, pp. 1200–1217, Oct. 2001.

[10] T. Simunic and S. Boyd, "Managing power consumption in networks on Chip," in *Proc. ACM Conf. Design, Automation and Test*, Paris, France, Mar. 2002.

[11] A. Sinha and A. Chandrakasan, "Dynamic power management in wireless sensor networks," *IEEE Design and Test of Computers*, Mar./Apr. 2001.

[12] V. G. Kulkarni, *Modeling, Analysis, Design, and Control of Stochastic Systems.* Springer-Verlag New York, Inc. 1999.

[13] P. Rong and M. Pedram, "Battery-aware power management based on Markovian decision process," in *Proc. IEEE/ACM Intl. Conf. Computer-aided design*, San Jose, CA, Nov. 2002.

[14] C.-F. Chiasserini and M. Garetto, "Modeling the performance of wireless sensor networks," in *Proc. IEEE INFOCOM*, Hong Kong, China, Mar. 2004.

[15] Z. Shelby, C. Pomalaza-Rez, and J. Haapola, "Energy optimization in multihop wireless embedded and sensor networks," in *Proc. IEEE Intl. Symp. PIMRC*, Barcelona, Spain, Sep. 2004.

[16] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in *Proc. ACM MOBICOM*, Boston, MA, Aug. 2000.

[17] *http://www.cisco.com/en/US/products/hw/wireless/ps4555/products_data_sheet09186a0080088828.html*, 2004.

[18] L. M. Feeney and M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment," in *Proc. IEEE INFOCOM*, Anchorage, AK, Apr. 2001, pp. 1548–1557.

[19] B. Burns and J.-P. Ebert, "Power consumption, throughput, and packet error measurements of an IEEE 802.11 WLAN interface," in *Tech. Rep. TKN-01-007*, Technical University of Berlin, Berlin, Aug. 2001.